POL00029046

CHARTERIS

Subpostmasters v Post Office Limited

Expert Report of Dr Robert Worden

Draft 134
05 December 2018

Table of Contents

1. Introduction 4
   - 1.1 Background of the case 4
   - 1.2 Experience 4
   - 1.3 Sources of information 5
   - 1.4 Terminology and Scope of My Report 5
   - 1.5 Document Structure 6
2. Horizon issues and summary of opinions 8
   - 2.1 Robustness of Horizon 8
   - 2.2 Extent of Bugs in Horizon 11
   - 2.3 Reconciliation and Transaction Corrections 13
   - 2.4 Facilities available to Subpostmasters 14
   - 2.5 Facilities available to Post Office 17
   - 2.6 Mr Coyne's opinions 18
3. Business applications in Horizon 21
   - 3.1 Overview of Horizon Requirements 21
   - 3.2 The Point of Sale Application, and Customer Settlement 21
   - 3.3 Agency Activities 22
   - 3.4 Requirements: Branch and Back Office 22
4. Old Horizon (1998 - 2010) 24
   - 4.1 The Four-Level Architecture 24
   - 4.2 Hardware and Software in the Branches 26
   - 4.3 Back-End Architecture 29
   - 4.4 Audit Information 32
   - 4.5 Changes During the Period 2000 - 2010 35
5. Horizon Online (2010 - Present) 36
   - 5.1 Motivation for the Move to Horizon Online 36
   - 5.2 The New Division Between Branches and the Back End 36
   - 5.3 New Architecture in the Branches 37
   - 5.4 Back-End Architecture: Changed and Unchanged Elements 41

POL00029046

CHARTERIS

6. Architectural Topics Across Old Horizon and Horizon Online 46
6.1 User Error Detection and Prevention 46
6.2 Intrinsic Error Prevention 49
6.3 Reconciliation, Transaction Corrections and Acknowledgements 56
6.4 Hardware and Software Resilience (RHW) 57
6.5 Security and User Authentication 58
6.6 Development and Testing of Horizon 60
6.7 Horizon in Service 62

7. Horizon issues – Robustness of Horizon 67
7.1 Issues Addressed in this Section 67
7.2 Robustness of Horizon: My Opinion 67
7.3 Countermeasures to Achieve Robustness 71
7.4 My Experience of Robustness Countermeasures 73
7.5 Effect of Countermeasures on Bugs Which Might Affect Branch Accounts 74
7.6 Assessing How Well Countermeasures Were Applied 79
7.7 Opinions on Robustness Countermeasures 81
7.8 Variations in the Robustness of Horizon Over Time 91
7.9 Horizon Issue 4 94
7.10 Horizon Issue 6 95
7.11 Mr Coyne's Opinions 96

8. The Effect of Horizon Bugs on Branch Accounts 98
8.1 Horizon Issue 1: My Opinions 98
8.2 Unknown Bugs in Horizon 99
8.3 Impact of Bugs on Claimants' Branch Accounts - Qualitative Opinion 101
8.4 Measures of Extent 103
8.5 Scaling of Financial Impacts of Bugs 106
8.6 Analyses of the Three Errors Cited By the Claimants 110
8.7 Financial Impact of All Bugs - Main Analysis 116
8.8 Alternative Approaches to Estimate The Financial Impact of Bugs 125
8.9 Impact of Bugs in Horizon on Individual Claimants 129
8.10 Financial Impact of All Bugs, Using Data Provided by the Claimants 130
8.11 Extent of Bugs - the Number of Different Bugs 136
8.12 Processing and Recording of Transactions 137
8.13 Analyses needed in Support of My Opinions 137
8.14 Mr Coyne's Opinions 138

9. Horizon issues – Reconciliation and Transaction Corrections 142
9.1 The Issues 142

POL00029046

CHARTERIS

9.2 Summary of My Opinions 142
9.3 Reconciliation, Transaction Corrections and Transaction Adjustments 143
9.4 My Opinions on Horizon Issues 5 and 15 144
9.5 Mr Coyne's opinions 145
9.6 Financial Impact of Errors in TCs on Claimants' Branch Accounts 147

10. Horizon issues – Facilities available to subpostmasters 152
10.1 The Issues and My Opinions 152
10.2 Approach to the Issues: Assumptions 154
10.3 Horizon Issue 2 156
10.4 Horizon Issue 9 157
10.5 Horizon Issue 14 161
10.6 Mr Coyne's Opinions on Issues 2, 9, and 14 165

11. Horizon issues – Facilities available to Post Office 167
11.1 The Issues 167
11.2 Summary of My Opinions 167
11.3 Interpretation of the Issues 168
11.4 Horizon Issue 7 169
11.5 Horizon Issue 8 170
11.6 Horizon Issue 10 171
11.7 Horizon Issue 11 177
11.8 Horizon Issue 12 180
11.9 Horizon Issue 13 181
11.10 Mr Coyne's opinions 183

12. Declaration 185

POL00029046

1. INTRODUCTION

1.1 Background of the case

1 In this Group Litigation, over 500 Claimants (most of whom were and/or are Subpostmasters who operated and/or operate Post Office branches) seek damages or other relief arising from alleged shortfalls for which they were allegedly responsible and which some of them repaid.

2 The Claimants dispute that they were responsible for those shortfalls, alleging that the apparent shortfalls were caused or contributed to by errors or data changes in the Horizon point of sale system that they were required to use by Post Office or by failures by Post Office in the training, Helpline support or procedures followed. Some Claimants have been convicted of false accounting, fraud, theft or other offences in relation to shortfalls in Post Office branches and claim against Post Office in malicious prosecution. At present, the Claimants do not plead particulars of those claims, pending the outcome of the Criminal Cases Review Commission review which is currently ongoing in relation to the convictions of a significant number of the Claimants that are claiming malicious prosecution.

1.2 Experience

3 Robert Worden has acted as an expert in a number of disputes in the UK High Court, concerning information technology and intellectual property issues. he has acted in around twelve disputes during the past fifteen years, most of them substantial. Approximately half have settled before trial. In the other half he has given oral expert evidence to the Court. Technical issues addressed in his reports have concerned the construction and performance of large computer programs, the management of large IT systems and of the projects which develop such systems, and intellectual property issues.

4 His background is in physics and applied mathematics and he has a doctorate in theoretical particle physics.

5 He has more than forty years' experience in the software industry, working in a variety of technical and management roles. For the majority of that period, up to and including the present day, he has been actively developing software in a wide range of programming languages. These applications have been developed in several domains, including finance, defence, healthcare, business management, and system software.

6 In 1975 he joined Logica, which at the time was one of the major software houses in the UK. While at Logica he acted in a range of management and technical roles. He designed and developed a relational database management system (RDBMS) and then managed the business unit that commercialised it for several years. He managed a number of large development projects, and then took a role reviewing and trouble-shooting Logica's major high-risk projects in all market sectors.

7 He is currently managing director of Open Mapping Software Ltd., a company specialising in healthcare information technology. When involved in expert witness assignments, he works as an associate of Charteris Consulting Ltd.

POL00029046

8 Chris Emery has assisted Dr Worden in this dispute. He has provided expert advice on a series of IT disputes over a period of six years. These engagements included the following:

- UK National Health Service's National Programme for Information Technology, which was described as "the world's largest civil information technology programme". Cases required, amongst other issues, analysing the impact of delay events in relation to system development, testing, and deployment.
- Investigation into the batch systems failures at the RBS Group and its subsidiaries NatWest Bank and Ulster Bank in June 2012. This was a complex, large-scale IT failure which adversely affected millions of the bank's customers. A number of reports were submitted on different aspects of the incident and the recovery process. The case was settled out of court.
- Advising an engineering company on a dispute about a failed implementation of an ERP system.

9 He also assisted Dr Worden three years ago in two disputes involving ERP implementations.

10 He graduated from Imperial College, London with a first-class honours degree in Computing Science and has spent more than thirty years in IT. Starting out as a software developer, he soon progressed to project and programme management. In addition to providing expert advice, he has also undertaken IT-related consultancy and fulfilled a number of line management roles.

11 Chris Emery has been engaged in the implementation of major IT projects since the 1980s. He has managed many initiatives from inception to a successful conclusion working with teams that have been numbered in hundreds.

12 CVs for Dr Worden and Chris Emery are attached at Appendix I to this report.

1.3 Sources of information

13 I have confirmed from Mr Coyne's report that the sources of information made available to me are the same as the sources made available to him, except as follows:

14 The documents I have reviewed are the same as in points (a) - (n) of his paragraph 1.26, with the following additions:

- An Excel spreadsheet summary of the claimants' claims, taken from the schedules 3.1 and 8.1 of their claims, prepared for me by the defendants' lawyers.
- Evidence disclosed since the date of Mr Coyne's report, which consists of the defendant's further witness statements and attachments to them.

1.4 Terminology and Scope of My Report

15 In this report, I shall use the following terminology when referring to the Horizon system:

- I shall use the term 'Old Horizon' to refer to Horizon pre-2010, which used the Escher Riposte product and stored transaction data in the branches

POL00029046

- I shall use the term 'Horizon Online' to refer to Horizon post-2010, which stored transaction data in the central Branch Database, and not in the branches
- When I use the term 'Horizon' on its own, I am referring to the Horizon system both pre- and post-2010, or to the Horizon Issues.

16 When referring to Subpostmasters of either gender, for brevity I shall always use the pronoun 'he', which should be taken to mean 'he or she'.

17 I shall describe how robustness is achieved (in many IT systems, including Horizon) by applying a number of countermeasures, to ensure that certain risks do not have unacceptable consequences. I describe 18 countermeasures, and I shall refer to them by three-letter acronyms such as RDS (Redundant Data Storage). The 18 countermeasures and their acronyms are introduced in section 2, and the acronyms are in the glossary at Appendix A.

18 Some limitations in the analysis of some detailed points in Mr. Coyne's report which I have been able to carry out in the time available since receiving it, are described in section 2.6 of this report.

19 Before receiving Mr Coyne's report, I had understood that the experts were not being asked to address evidence about individual claimants. I have therefore not had time to address the witness statements of individual claimants, some of which are referred to in Mr Coyne's report.

1.5 Document Structure

20 Section 2 gives a summary of my opinions on the Horizon issues.

21 Section 3 describes the business requirements for the Horizon system, including the point of sale requirement in PO branches, the accounting requirement, and the range of services that PO offers on behalf of its clients

22 The Horizon system has undergone frequent changes, in a complex history since its inception in 1999. I will describe its architecture in two main time periods.

23 Section 4 describes the Old Horizon system as it was in the period 2000 - 2009. After giving a central 'snapshot' of this period, I describe the most important changes during that period

24 Section 5 describes 'Horizon Online' (HNG-X, and later HNG-A), introduced in 2010, in which a major element of the architecture was changed. In Horizon Online, instead of holding persistent transaction data in each branch, all transaction data was held centrally in a single branch database. Horizon Online involved a complete refresh of the software in the branches. Many central elements of Horizon persisted over both periods, as will be described in section 6.

25 Sections 4 and 5 both describe complex Horizon architectures, with many major components, and interactions between the components which can be partially understood from architecture diagrams, showing how data are passed between the systems

26 Section 6 builds on the previous sections to address these topics across the whole sequence of Horizon architectures. Section 6 describes how the robustness countermeasures were built into Horizon.

POL00029046

27 Sections 4, 5, and 6 together provide the basis for understanding the whole Horizon architecture, how it was intended to achieve robustness against a variety of threats, and the extent to which it did achieve robustness. This understanding is, in my opinion, the essential basis for approaching the Horizon issues.

28 The 15 Horizon issues are then addressed in five groups, in the following order:

29 Section 7 addresses the Horizon issues which mainly concern the robustness of Horizon. These are issues 3, 4, and 6.

30 Section 8 addresses Horizon issue 1, the extent to which bugs in Horizon may have affected the claimants' branch accounts.

31 Section 9 addresses Horizon issues 5 and 15, which concern reconciliation and transaction corrections.

32 Section 10 addresses those Horizon issues which concern the access of subpostmasters to information. These are issues 2, 9 and 14.

33 Section 11 addressed those Horizon issues which concern the facilities available to PO centrally. These are issues 7, 8, 10, 11, 12, and 13.

34 Section 12 contains my formal declaration that I understand and have complied with my duty to the Court.

35 Appendix A contains a glossary of terms used in this report.

36 Appendix B gives some background on accounting systems and the principles of double entry book-keeping which they embody.

37 Appendix C follows the same organisation as section 6 of the report, and addresses some of the topics of section 6 in greater detail.

38 Appendix D contains four tables giving my analysis of the KELs which I have examined, both to assess the evidence for application of robustness countermeasures when addressing Horizon issue 3, and to estimate the impact on claimants' branch accounts of all detected bugs and errors in Horizon, when addressing Horizon issue 1.

39 Appendix E contains two quantitative analyses I have made of the impact of bugs in Horizon on claimants' branch accounts, using evidence provided by the claimants as part of their claim. The results of these calculations are summarised in section 8.

40 Appendix F contains two pieces of detailed analysis referred to in section 8 of the report. These concern (a) the possibility that claimants' branches were affected by bugs in Horizon more frequently than other PO branches, and (b) the possible influence of 'micro-bugs' whose financial impact on any one occurrence was so small as not be noticed by SPMs.

41 Appendix G contains a detailed analysis of two sample Peaks.

42 Appendix H contains some responses to some detailed points in Mr Coyne's report.

43 Appendix I contains my CV and Chris Emery's CV.

POL00029046

2. HORIZON ISSUES AND SUMMARY OF OPINIONS

44 In this summary and in the body of my report I have grouped the 15 Horizon issues into five groups of related issues, so that my opinions can be organised as clearly as possible. The groups have overlaps with each other and therefore require some cross-referencing between groups. In what follows, for each group, I will first state the Horizon issues, and then give a summary of my opinions. In section 2.6, I summarise where my opinions differ from Mr Coyne's opinions, as expressed in his report.

2.1 Robustness of Horizon

45 The first group of issues, which addresses the robustness of Horizon, includes the Horizon issues 3, 4, and 6. I address these issues in section 7 of my report.

46 **Issue 3**: To what extent and in what respects is the Horizon System "robust" and extremely unlikely to be the cause of shortfalls in branches?

47 **Issue 4**: To what extent has there been potential for errors in data recorded within Horizon to arise in (a) data entry, (b) transfer or (c) processing of data in Horizon?

48 **Issue 6**: To what extent did measures and/or controls that existed in Horizon prevent, detect, identify, report or reduce to an extremely low level the risk of the following:

a) data entry errors;

b) data packet or system level errors (including data processing, effecting, and recording the same);

c) a failure to detect, correct and remedy software coding errors or bugs;

d) errors in the transmission, replication and storage of transaction record data; and

e) the data stored in the central data centre not being an accurate record of transactions entered on branch terminals?

49 In my opinion the most important of these is issue 3, which encompasses a large and mature area of modern IT practice. Nearly all business IT systems need to be robust - as the business depends on them - and there is a large, mature and tested set of techniques for achieving robustness. Issues 4 and 6 then in effect address some specific subsets of issue 3.

50 I here summarise my opinion on Horizon issue 3.

a) My opinion from the evidence is that at all times for which there are KELs - which is nearly all the lifetime of the system - Horizon has been a very robust system, compared to other major systems I have worked on in sectors such as banking, retail, telecoms, government, and healthcare.

b) I have described 18 types of robustness countermeasure, which I have applied routinely on projects over many years. In my opinion, Fujitsu have applied these countermeasures effectively in building and supporting Horizon.

c) As an accounting system, Horizon particularly needs countermeasures to ensure the accuracy of the accounts, in the face of many types of adverse event. I have focused particularly on these countermeasures and their effectiveness. In my opinion, these countermeasures are well designed,

POL00029046

and have been effective in preventing errors in accounts. Very few adverse events - including user errors and software bugs - have evaded all the countermeasures to the extent of causing significant inaccuracies in branch accounts. Horizon is very unlikely to cause significant shortfalls in branches.

51 This summary of my opinions applies also to the Horizon issues 4 and 6..

52 The experts have agreed that robustness is not a matter of perfection, or the complete absence of bugs. Horizon, in common with all large commercial IT systems, was not completely free of bugs.

53 Robustness involves the use of a set of techniques, which I call countermeasures, to ensure that many kinds of potentially harmful events (including hardware failures, communications failures, user errors and software bugs) do not have harmful consequences - or if they do, the harmful consequences are kept within acceptable limits.

54 Here, the definition of 'acceptable' involves using standard IT techniques of Risk Assessment[1] which I have applied retrospectively to assess the risk of bugs in Horizon introducing errors in branch accounts. The results are reported in section 8.

55 Robustness is a core requirement for any major commercial IT system and has been so for many years. Large parts of IT project budgets are spent ensuring that systems are robust.

56 The techniques for achieving robustness are so important that they have become a well-established and central part of commercial IT practice. I have listed 18 major techniques, or types of countermeasure, which I have routinely applied on major projects over thirty years. These techniques act in concert to minimise harmful effects. They are described in sections 4-6 of my report.

57 To make Horizon robust, Fujitsu had to effectively apply these established techniques. I have examined the evidence of how Fujitsu designed and built Horizon, how they tested it (in section 6.6), and how they supported it (in section 6.7). The 8390 KELs, in particular, are a rich source of evidence about Horizon in service - about events which threatened to have harmful consequences, and how well or badly the robustness countermeasures acted in those cases. My analysis of many KELs implies to me that the countermeasures in Horizon worked well in the live use of Horizon.

58 In some market sectors where I have worked, robustness is often compromised by the presence of very old legacy software (sometimes coming from merged organisations) which is hard to maintain or adapt - and has resulted in an over-complex frozen 'spaghetti' architecture. Horizon does not suffer from these problems. Horizon was a 'green field' development started in 1996 - essentially unencumbered by any IT legacy. Therefore, it was much easier to build a robust architecture from the start.

59 In his report, Mr. Coyne has not described any robustness countermeasures, or assessed how well they were applied in Horizon. In my opinion, robustness of Horizon is the central expert issue of this case, because it

[1] http://prince2.wiki/risk describes the risk management methodologies, which form part of the PRINCE2 project management methodology used by Fujitsu.

POL00029046

is robustness - rather than the number of bugs - which determines the financial impact of bugs on claimants' accounts.

60 In this report, I focus on eighteen categories of countermeasure, which are summarised in the table below.

61 For each robustness countermeasure, I have provided a three-letter acronym to identify it. Most of these acronyms are not commonly used in the industry but are provided to enable the reader to recognise and cross-reference the different countermeasures, in the many places where they will appear in this report.

No.	Countermeasure	Explanation and examples	Described in Section
1	Reliable and redundant hardware (RHW)	Redundancy guards against many types of hardware failure. Examples: RAID discs, disaster recovery sites. Software is designed in many ways to be robust against hardware failures	4.2, 5.4, 6.4
2	Robust data communication and replication (ROC)	Communication systems and protocols are designed to recover from and protect against many kinds of communication failure. Examples: TCP/IP, Riposte	4.2, 5.3, 6.4
3	Transactional Integrity and database recovery (TIN)	Database management systems provide many facilities so that numerous kinds of failure cannot leave the data in an inconsistent, unusable state, or lose any data that have been previously stored	4.2, 4.3, 5.4, 6.2, 6.4
4	Defensive programming (DEP)	Software is divided into small self-contained modules, which do not assume that other modules are correct, but defend themselves by checking their inputs and raising alerts early	5.3, 6.2, 6.4
5	Generic, data driven software (DDS)	Different use cases for software often have much in common. Software is written generically to be able to handle the different cases, using reference data to define which use case is to be handled. Example: variations in PO client products handled by reference data.	4.2, 4.3, 5.3, 6.2
6	Secure kernel hardware and software (SEK)	When a large complex IT system is subject to threats, the design may include a small, well tested and secure kernel which is proof against those threats. Examples: secure kernels of operating systems, Horizon core audit process	4.3, 4.4, 5.4, 6.2
7	Redundant data storage and computing, with cross-checks (RDS)	In large IT systems and sets of systems, data are stored redundantly in several places, and routine operations check automatically that the different copies of the data remain consistent.	4.3, 4.4, 5.4, 6.2, 6.4
8	Double entry accounting (DEA)	Accounting systems operate by the principles of double entry book keeping, so that any change to the accounts must be made in a transaction whose summed effect on all accounts is zero. Transactions which do not obey this constraint are rejected.	4.2, 5.3, 5.4, 6.1, 6.2
9	Early detection of user errors (DUE)	At the point of user input, as many checks as possible are made of the correctness of the input - so that the system will not accept erroneous input and may warn the user of errors.	5.3, 6.1
10	Later correction of user errors (UEC)	In accounting systems, the system's version of reality is periodically checked against external versions of reality and corrected if wrong. Examples: cash balancing and rollover, reconciliation and TCs.	4.2, 4.3, 6.1, 6.2, 6.3

POL00029046

11	Manual workarounds (WOR)	Whenever any part of Horizon does not work as required, there may be potential to define and apply manual workarounds.	6.7
12	Testing good practice (TGP)	The purpose of system testing is not to prove that the system is correct, but to prove that it is incorrect in any way possible. Examples: regression testing, user testing, testing edge cases.	6.6
13	Manual Inspection of data (MID)	Any large business IT system is used by many people, who view its outputs and check them against each other for consistency, and against their own knowledge of the business. SPMs, watching their branch accounts, were a key component of this.	4.3, 6.2
14	Bug Finding and Correction (BFC)	Whenever the system shows any anomalous behaviour, that is investigated, its causes found and corrected. Interim workarounds are deployed. Extra checks may be added to ensure that other similar threats are handled correctly.	6.7
15	Large scale IT architecture (ARC)	In any large IT estate, principles of IT architecture are used to achieve robustness - such as using a distributed network of loosely coupled sub-systems with clearly distinguished functions. The sub-systems are built to well-defined standards with clear interfaces.	4.1, 5.3, 5.4, 6.2, 6.4
16	Quality and change Control (QCC)	Systems are more robust if quality is inherent. This is achieved by organising properly the people who build, maintain and operate the system, by managing them well and by governing what they do through rigorous but effective processes. A system will only continue to be robust if changes are controlled in a way that enhances quality without unnecessary administration.	6.6.2, 6.7.3
17	Managing non-functional requirements (NFR)	Robustness is improved by paying close attention to non-functional requirements and the associated 'ilities' such as manageability, supportability, maintainability and adaptability	-
18	Security (SEC)	Any system that could be easily subverted would not be robust. Horizon is secured mainly through 'separation of duties', user authentication, access control and audit.	6.5

62 Much of sections 4, 5, and 6 is devoted to describing how these countermeasures have been built into the architecture of Horizon.

2.2 Extent of Bugs in Horizon

63 The second group of issues consists of Horizon issue 1, which concerns bugs in Horizon which might have had an impact on branch accounts. It is addressed in section 8 of my report.

64 **Issue 1**: To what extent was it possible or likely for bugs, errors or defects of the nature alleged at §§ 23 and 24 of the GPOC and referred to in §§ 49 to 56 of the Generic Defence to have the potential to (a) cause apparent or alleged discrepancies or shortfalls relating to Subpostmasters' branch accounts or transactions, or (b) undermine the reliability of Horizon accurately to process and to record transactions as alleged at §24.1 GPOC?

POL00029046

65 In my opinion on part (a) of issue 1:

i. **Significant detected defects:** If in some month there was a significant shortfall in any claimant's branch accounts (a shortfall of £300 or more), the chances of that having arisen from a bug or defect in Horizon which has been detected are very small indeed. I have assessed this quantitatively by a retrospective IT Risk Analysis, with the following result: the probability of any of the three known bugs introducing a discrepancy in a claimant's branch accounts in any given month is of the order of two parts in a million. To make that probability as large as one part in 10, there would need to be more than 50,000 distinct bugs in Horizon, each of which created errors in branch accounts comparable to one of the three known bugs (which are discussed in section 8.6). The figure of 50,000 bugs is to be compared to the handful of bugs possibly affecting branch accounts which have been disclosed (i.e. the three known bugs), or found by the experts. This figure is derived in section 8.5, by a simple calculation, using evidence which in my opinion has only a small margin of uncertainty. The result is stable under changes of assumptions; if the assumptions change, the result does not change much.

ii. **Undetected defects**: The claimants have raised the possibility that shortfalls might be caused by defects in Horizon which were never detected and may not be known about to this day. Because of the many countermeasures built into Horizon, the potential for any such 'unknown bugs' is very small indeed. Any bug with significant impact on branch accounts would be known about. The net impact of unknown bugs on branch accounts is therefore very small, compared with the impact of defects which are known about and were recorded in KELs.

iii. **Financial Impact of defects**: Because of (ii), the KELs are a good source of information about bugs and the effect they might have had on branch accounts. One can examine the KELs, determine in which of them there might have been an impact on branch accounts, and place a conservative upper limit on the amount of this impact. Doing this sum, correcting for factors such as any inefficiency of the KEL creation process, lack of detail in KELs, and limitations in the sample of KELs I have been able to examine, I have calculated an upper limit on the financial impact of bugs in Horizon on the claimant's accounts. This upper limit is very small. Even using very conservative assumptions, designed to favour the claimants, the total net impact of all bugs in Horizon on the claimant's branch accounts must be less 0.1% of the shortfalls claimed by the claimants.

66 Bugs in Horizon cannot account for even a small part of the claimants' shortfalls - either for all claimants taken together, or for any individual claimant.

67 In my opinion this result follows inescapably from the evidence.

68 In my opinion of part (b) of issue 1: the Horizon Core Audit Process was designed to create a secure, accurate and immutable record of what was entered into Horizon at the branch, and to record verifiably who had entered it. In my opinion, regardless of any other processing done in other parts of Horizon, the core audit database was an accurate record of transactions entered in the branch. It was carefully designed ,

POL00029046

and I have seen no evidence that it ever failed in service. Therefore, in any case of doubt about processing done in other parts of Horizon, this record was available to establish the true state of any branch's accounts, based on transactions entered in the branch.

69 These opinions apply both to old Horizon (pre-2010) and Horizon Online (HNG).

70 In section 7, I addressed Horizon Issue 3: To what extent and in what respects is the Horizon System "robust" and extremely unlikely to be the cause of shortfalls in branches? I said there that I would postpone addressing the second part of that issue, 'extremely unlikely to be the cause of shortfalls in branches' to this section. As my opinion on part (a) of Issue 1 makes clear, in my opinion on Issue 3, the robustness of Horizon made it extremely unlikely to be the cause of shortfalls in branches.

2.3 Reconciliation and Transaction Corrections

71 The third group of issues includes the Horizon issues 5 and 15, concerning the related topics of reconciliation with external parties, and Transaction Corrections (which often arise from reconciliation). These have been treated as a group because they belong so closely together. However, as they are an important part of the way Horizon is made robust against a variety of user errors, they relate to issue 3 in the first group. The issues are:

72 **Issue 5**: How, if at all, does the Horizon system itself compare transaction data recorded by Horizon against transaction data from sources outside of Horizon?

73 **Issue 15**: How did Horizon process and/or record Transaction Corrections?

74 Issues 5 and 15 are, on the face of it, factual issues, which can be addressed by factual evidence, as is described in sections 6.4 and 9

75 I repeat my comments in section 1 about my role as an expert and my approach to contested factual matters and evidence.

76 These questions do not invite an opinion on the quality, adequacy, sufficiency or other similar judgment on these processes. In light of this, I provide below my opinion on the evidence I have seen to address the factual questions of whether and, if so, how Horizon undertakes certain activities.

77 I also note that these questions are limited to activity regarding Horizon, and do not extend to other manual business processes operated by Post Office. Save for providing useful context on these other areas, my opinion is limited accordingly.

78 Mr Coyne has gone further than the above scope. He has offered opinions on the adequacy of the reconciliation process in a wider sense - in particular, raising the question of errors in transaction corrections. For the sake of balance, in section 9.6 I offer my own commentary on these matters without prejudice to my understanding of the scope of Horizon Issues 5 and 15.

79 My analysis of the evidence is that:

POL00029046

- For most of PO's clients (for whom PO branches carries out agency business) there is a regular automated process of comparing (reconciling) the transactions as recorded by PO, with the transactions as recorded by the client organisation.
- These comparisons may or may not be carried out within Horizon 'itself' ; but in any event, because of the large volume of transactions, the comparison has to be automated
- Whenever the comparison reveals any discrepancy, there appears to be a human process of deciding where to allocate responsibility for the discrepancy. This has to be a human process, and is therefore subject to errors.
- If responsibility is allocated to a branch, it results in a Transaction Correction (TC), which the branch may accept or query before it enters the branch accounts
- There is also reconciliation of cash remmed from branches to PO cash management, or in the reverse direction

80 The thrust of Mr Coyne's opinions on these issues - for instance in his summary paragraphs 3.13 and 3.28 - is to emphasise that reconciliation, and the creation of transaction corrections, are error-prone processes.

81 The significance of this for the claimants' case appears to be that any such errors might have introduced shortfalls in the claimant's branch accounts.

82 Because of this emphasis by Mr Coyne on errors in TCs, I need to address the topic of errors in TCs, and will do so quantitatively in section 9.6. I have calculated the likely extent of shortfalls in claimants' accounts arising from erroneous TCs, using evidence on:

- annual volumes of TCs (numbers and monetary amounts)
- the distribution of types of TC, in a typical year
- proportions of TCs disputed, and the proportion of disputes upheld
- the number and sizes of branches, both for claimants and other PO branches

83 The result is that an upper limit on the magnitude of the mean discrepancy which might have been introduced by erroneous TCs into any claimant's branch accounts in any month, is of the order of £2. This is to be compared to the mean shortfalls of £360 per month claimed by the claimants.

84 Then the probability of some larger discrepancy having been introduced in any given month are very small - for instance, the chances of a discrepancy of £1000 would be one in 500. This is explained in section 8.

2.4 Facilities available to Subpostmasters

85 The third group of issues includes the Horizon issues 2, 9, and 14, because these all relate to the Horizon facilities available to subpostmasters when running their branches. The issues are:

86 **Issue 2**: Did the Horizon IT system itself alert Subpostmasters of such bugs, errors or defects as described in (1) above and if so how?

POL00029046

87 **Issue 9**: At all material times, what transaction data and reporting functions (if any) were available through Horizon to Subpostmasters for:

a) identifying apparent or alleged discrepancies and shortfalls and/or the causes of the same; and

b) accessing and identifying transactions recorded on Horizon?

88 **Issue 14**: How (if at all) does the Horizon system and its functionality:

a) enable Subpostmasters to compare the stock and cash in a branch against the stock and cash indicated on Horizon?

b) enable or require Subpostmasters to decide how to deal with, dispute, accept or make good an alleged discrepancy by (i) providing his or her own personal funds or (ii) settling centrally?

c) record and reflect the consequence of raising a dispute on an alleged discrepancy, on Horizon Branch account data and, in particular:

i. does raising a dispute with the Helpline cause a block to be placed on the value of an alleged shortfall; and

ii. is that recorded on the Horizon system as a debt due to Post Office?

d) enable Subpostmasters to produce (i) Cash Account before 2005 and (ii) Branch Trading Statement after 2005?

e) enable or require Subpostmasters to continue to trade if they did not complete a Branch Trading Statement; and, if so, on what basis and with what consequences on the Horizon system?

89 Issues 2, 9 and 14 are on the face of them factual issues, which can be largely resolved by factual evidence, and might not in themselves lead to much expert disagreement. I address them in section 10 of my report.

90 However, they need to be approached in the light of the claimants' case, and certain assumptions apparently built into it, and in Mr Coyne's report. These assumptions appear to be that:

- It would have been a good thing to provide SPMs more information about the workings of Horizon than was given to them.
- If there was a fault in Horizon, there should have been some useful automatic way for Horizon to tell SPMs what it was.
- In the case of an anomaly, it was incumbent on the SPM to dispute the cause of the anomaly with PO.
- In doing so, SPMs could benefit from information about the back-end systems of Horizon to infer that some anomaly was caused by a bug in Horizon
- Because SPM s did not have all this information, but PO did, there was an asymmetry of information between SPMs and PO - which PO used to unfairly attribute the effects of bugs in Horizon to human error by the SPMs

POL00029046

91 In my opinion, these assumptions all rest on an unrealistic picture of how commercial IT systems are built, used and supported:

- It is not a good thing to give the users information about parts of an IT system which they do not encounter in their daily work, and which they know very little about. They will be perplexed by it.
- To anticipate the small proportion of cases where the IT system is in error, there is no point in trying to educate all the users in details and terminology of the system which will almost never concern them.
- An IT system can give its users useful warnings and error messages in a variety of situations, but generally not in the case of previously undiscovered bugs in the system.
- When the developers of an IT system discover some bug or defect in it, the best thing to do is to fix it, rather than to create some new error message to the users.
- When an IT system gives results, which puzzle its users (for any cause), further automated messages from the system are only of limited help to users. They need support from a human being, who may need to take account of the circumstances and bring to bear a wide variety of knowledge.
- Anomalous results may arise for a wide variety of reasons - from human error, to errors in processing at the back end. Understanding the causes often depends on cooperation between the user (who knows what he did) and support staff (who know much more about back-end systems). To portray this cooperation as a dispute is misleading.
- Staff and organisations who support an IT system have a strong incentive to understand bugs and to get them fixed, to reduce their future workload. They have no interest in leaving bugs unfixed, so the same problems keep recurring.

92 Putting to one side the assumptions in the claimants' case, my opinions are as follows.

93 Issue 2: Horizon did not in general alert SPMs to any significant bugs or other defects in the system itself. Nor should it have done.

94 Issue 9: In my opinion, most discrepancies are caused by human error. The functions available from Horizon, when used in accordance with PO guidance and procedures, enable SPMs to identify the causes of such discrepancies. SPMs and their staff are the best placed to investigate such discrepancies, because they are the only people who have first-hand knowledge of what happens in their branches. The PO and Fujitsu support teams can only use their knowledge of systems and the data stored within them; whereas the SPM can use their knowledge of what happens in the real world.

95 The main concern of an SPM is the successful running of their branch. This means that they may have limited time and patience to investigate discrepancies in their accounts, however they think they may have arisen. The reports available to them focus on activities carried out within the branch, their key area of expertise. If they are, nevertheless, unable to identify the problem, their best course of action is to ask for help.

POL00029046

96 Issue 14 asks a number of specific questions about the facilities of Horizon for SPMs, which I answer in section 10.4

.

2.5 Facilities available to Post Office

97 The fourth group of issues includes the Horizon issues 7, 8, 10, 11, 12, and 13, which all relate to facilities available to the Post Office centrally or to Fujitsu, rather than to subpostmasters. I address these issues in section 11 of my report.

98 The issues are:

99 **Issue 7**: Were Post Office and/or Fujitsu able to access transaction data recorded by Horizon remotely (i.e. not from within a branch)?

100 **Issue 8**: What transaction data and reporting functions were available through Horizon to Post Office for identifying the occurrence of alleged shortfalls and the causes of alleged shortfalls in branches, including whether they were caused by bugs, errors and/or defects in the Horizon system?

101 **Issue 10**: Whether the Defendant and/or Fujitsu have had the ability/facility to: (i) insert, inject, edit or delete transaction data or data in branch accounts; (ii) implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts; or (iii) rebuild branch transaction data:

a) at all;

b) without the knowledge of the Subpostmaster in question; and

c) without the consent of the Subpostmaster in question.

102 **Issue 11**: if they did, did the Horizon system have any permission controls upon the use of the above facility, and did the system maintain a log of such actions and such permission controls?

103 **Issue 12**: If the Defendant and/or Fujitsu did have such ability, how often was that used, if at all?

104 **Issue 13**: To what extent did use of any such facility have the potential to affect the reliability of Branches' accounting positions?

105 For Issue 7, I have interpreted the word 'access' to mean 'read-only access' - because otherwise, issue 7 would be a subset of issue 10With that interpretation, both PO and Fujitsu had access to a wide variety of transaction data. They need this access for a wide range of purposes.

106 On Horizon Issue 8, the information required to investigate alleged shortfalls is available to PO from several sources. Their perspective is to look into branch accounts from the outside, with no first-hand knowledge of what has occurred from day to day. On the other hand, they look out to their external clients on whose behalf they are brokering business based on those clients' services and products. By virtue of their role in the end-to-end business, PO has access to information not available to SPMs and vice versa.

107 Issue 10 relates to both PO and Fujitsu. It comprises three parts, numbered (i) – (iii). The experts are asked to examine each part in three different respects identified as (a) – (c). Therefore, in principle, the

POL00029046

issue calls for 18 opinions although they are not all distinct. Therefore, my opinion on Issue 10 has several parts, which are given in section 11.5, and are summarised in a table there

108 Part (i) of issue 10 is the most complex, and the answer is difficult to summarise in few words. Part (ii) is simpler, in that Fujitsu necessarily had the ability to implement fixes in Horizon, and these fixes necessarily had the potential to affect branch accounts in the future. Similarly, for part (iii) of Issue 10, Fujitsu had the ability to rebuild transaction data, because this was a very necessary part of the robustness countermeasures. It is important to understand that this rebuilding was an automated process, using a redundantly stored copy of the transaction data (RDS), and did not involve discretionary manual rebuilding.

109 Issue 11 asks about permission controls and logs of these processes. As Issue 11 follows from Issue 10, my opinions on it must have several parts, which are described in section 11.6. I only note here that in my opinion any alterations of branch transaction data are necessarily subject to the constraint of double entry accounting, DEA (for instance, when they propagate to POLSAP) - and any central user who made any such change would leave many traces of his activity, through several kinds of redundant data storage, RDS - like footprints in fresh snow.

110 Issue 12 asks about how often the facilities under issue 10 were used. In section 11.7, I summarise the evidence I have seen on this topic.

111 For Issue 13, I interpret 'extent' as I have interpreted extent for Horizon Issue 1; and I address issue 13 with respect to parts (i), (ii), and (iii) of issue 10.

112 I ask the questions with reference to the accounts for a specific claimant in a specific month. If a claimant were to assert that the use of any such facility had introduced a discrepancy into his accounts in any specific month, what is the probability of that assertion being correct?

113 In summary on Horizon Issue 13 applied to changes under issue 10(i) (insert, inject, edit or delete transaction data or data in branch accounts): for these changes to have any significant chance of affecting a claimant's branch accounts in a given month, there would need to be a huge number of them - probably of the order of 1 million. In my opinion, this is not possible.

114 I also addressed Horizon issue 13, as applied to parts (ii) and(iii) of issue 10. In both respects, the chances of introducing an error in a claimant's branch accounts in a given month are very small indeed - unless there are a very large number of such changes made in error. Details are given in section 11.8.

2.6 Mr Coyne's opinions

115 In summarising my response to Mr Coyne, I first point out four limitations which I have found in his report:

a) **Focus on Impact of Errors**: In his report, Mr Coyne draws attention to a number of errors and imperfections in the operation of Horizon over its 18-year history. As Horizon is a very large IT system, inevitably there have been errors. The key issue in Horizon Issue 1, is the extent to which

POL00029046

those errors had impact on branch accounts. Mr Coyne's report indiscriminately cites issues irrespective of their financial impact, and so does not focus on the question of financial impact.

b) **Understanding of Robustness**: Mr Coyne's conclusions on robustness are equivocal - sometimes finding Horizon robust, and at other times not. His conclusions partly reflect the agreement reached in the experts' joint memorandum (that robustness is not the same as perfection), while in other places, he appears to equate robustness with perfection. However, more important than this - Mr Coyne does not make it clear that robustness is not just some general hygiene factor, to be assessed on a single scale as more or less good. It is a specific and established set of IT practices and countermeasures. It is possible to classify the types of these countermeasures, to assess how each type was applied in the building of Horizon, and to assess how well they have worked in practice. Mr Coyne's report does not do this.

c) **Analysis and Context**: In his report, Mr Coyne cites a large number of KELs, Peaks, and other reports. His citations of KELs or reports are brief - typically one or two paragraphs. These brief citations do not explain the context and meaning of each KEL or report. While some KEL or report may appear at first sight to be relevant to an issue, or by selective quotation appear to have a certain implication, this cannot be assessed from the brief treatment, and lack of any deeper assessment, given by Mr Coyne.

d) **Linkage from Evidence to Conclusions**: Because Mr Coyne's treatment of each piece of evidence that he cites is typically brief and superficial, when he states his conclusions, the linkage from evidence to conclusions is often tenuous and difficult to understand. The evidence cited may tend to build up an impression, which loosely points in the direction of the conclusion. However, the impression may be misleading (as the evidence is not analysed) and the linkage is not explained; the conclusion does not follow from the evidence.

116 In responding to Mr Coyne's report, the limitations (c) and (d) have caused me problems. In order to comment on any piece of evidence cited by Mr Coyne (such as a KEL, a Peak or a report), I need in each case to provide a depth of analysis (for instance, in terms of financial impact or robustness countermeasures) not provided by Mr. Coyne. To do this for each point made by Mr. Coyne has not been possible in the time since I received his report. The approach I have taken in this report is therefore:

- First, to ensure that my own opinions on each issue are stated as clearly and concisely as possible, with linkage to the evidence I cite.
- For each Horizon issue, to contrast my own opinions with Mr. Coyne's - pointing out where my opinions agree with or differ from his opinions or go beyond them.
- For the KELs and Peaks cited by Mr Coyne, to provide a preliminary analysis in appendixes in tabular form. This analysis will be converted to a more thorough analysis in my supplemental report. This, I expect, will lead to a fuller account, and possibly to revisions on individual KELs, but not to any substantive new opinions.

POL00029046

- For the reports cited by Mr Coyne, to illustrate by selected examples why a deeper analysis is required to assist the court. In my supplemental report, I will provide that deeper analysis.

POL00029046

3. BUSINESS APPLICATIONS IN HORIZON

3.1 Overview of Horizon Requirements

117 The functionality of Horizon is more than that of an accounting system, because Horizon also supports a large and increasing, number of business applications.

118 For every kind of activity which a customer might enter a post office branch to carry out (such as buying a book of stamps, or paying a bill, or renewing road fund tax, or withdrawing cash from an account) there needs to be functionality in Horizon, both to support the counter activity of carrying out the transaction, and for the back office activity of settling with the PO's 'client' organisation, who has provided some service to the customer - such as the DVLA, or a bank. Accounting is a thread running through all of these business requirements, but it is only a part of them.

119 The number of services provided by PO branches is large and has increased steadily from 1998 to the present day. The functionality of Horizon has expanded in line with the growth in service, both on the counter and in the back office.

3.2 The Point of Sale Application, and Customer Settlement

120 Horizon counter activities are surveyed in the document 'HNG-X Counter Business Applications Architecture'.[2]

121 For part of its activities - such as selling stamps - a Post Office branch acts like a retail outlet, and it needs hardware and software to support this activity. This is the Electronic Point Of Sale Software (EPOSS) component of Horizon. EPOSS must allow the counter staff to record that some goods have been provided to a customer, compute the price of those goods, and allow the customer to pay the money required for all their purchased goods, for instance by cash or a credit card.

122 If a customer wants to carry out two or more different activities in one visit to the counter - for instance, to settle a bill and to buy some stamps - Horizon should not oblige the customer to settle the amount in two separate pieces. So, Horizon has the concept of a customer carrying out a 'basket' of activities and settling the total amount due for the basket in several ways - by one credit card transaction, by a cheque, by cash, or by a mixture of these.

123 However, baskets of PO activities and non-PO activities are not supported. If a customer wishes to buy a newspaper and some stamps, the newspaper is not sold by PO - it is sold by a separate retail outlet which uses the same premises. So, the customer has to settle in two parts. In this respect, the National Lottery is an exception and spans the two businesses.

124 So, Horizon needs to support retail-like activities (such as buying stamps) and agency-like activities (such as paying a bill) within a single customer basket, which may be settled by a compound set of payments.

[2] HNG-X Counter Business Applications Architecture (ARC/APP/ARC/0009)

POL00029046

3.3 Agency Activities

125 The Post Office refers to other organisations, for which it provides customer services in its branches, as its 'clients'. They include high street banks (for offering banking services), gas and electricity companies (for paying bills), DWP (for paying benefits and pensions) and DVLA (for paying road fund tax).

126 The Post Office currently has several hundred client organisations, which shows the diversity of services available in a branch. This also implies that, for most of these clients, the service provided through the Post Office will be different in nature from the service provided for other clients, so some unique software functionality must be provided both in the branch and the back office, to support the activities for that client. This is a part of what makes Horizon such a large and complex system.

127 It is not possible or useful in this report to describe all the types of service provided at PO counters, or the software needed to support them. We will only touch on a few services which either illustrate the diversity of services in a representative way or are important in this dispute.

128 A high-level classification of the services now offered in branches on an agency basis included the following[3]:

- **Paying bills** to BT, utilities, local Government
- **Prepayment services** - DVLA savings stamps, gift vouchers and entertainment tickets
- **Acquiring licences** - local Government permits, television, motor vehicle
- **Money management** - banking deposits and cash withdrawals, savings and investments
- **Insurance services** - general, travel
- **Pensions payments** - e.g. for MoD
- **Lottery** - for Camelot

3.4 Requirements: Branch and Back Office

129 As well as the counter activities described above, Horizon also needs to support the periodic process of balancing and rollover for each branch. Every branch operates in Trading Periods (TP), which are either four or five weeks –according to a timetable published periodically by PO). At the start of each TP, the branch is supposed to be 'in balance'. This means that the physical stock and cash in the branch agrees with the data on stock and cash held in Horizon. Then, during the Trading Period, Horizon records all customer transactions made at the branch, so it records the changes in cash and stock. It also records any replenishments or remittances[4] of cash or stock in the branch. Thus, Horizon records all changes in cash and stock held at the branch during the TP, and can compute, from the starting amounts and the changes, the expected amounts of cash and stock at the end of the period.

---

[3] TD/ARC/001.

4 At PO, 'remittance' is often abbreviated to 'remming'. This means sending surplus cash from a branch to the centre, or replenishing cash or stock in a branch from the centre.

POL00029046

130 At the end of each Trading Period, the subpostmaster (SPM) counts the physical cash and stock in the branch and compares it with Horizon's expectations of the same values. This is called 'balancing'. If the numbers are all equal, the branch is in balance and can 'roll over' to the next period. If the two sets of numbers are not equal, this implies that some of the transactions entered into Horizon during the Trading Period were erroneous or had failed to be entered. For instance, if the counted stock of stamps is less than the expectation from Horizon, this implies that some stamps were given away or lost, without recording a transaction on Horizon. Because it is assumed that this arose through some error by the SPM or his staff, the SPM is required to take responsibility for the discrepancy, in some way he or she chooses - for instance, by paying in cash to cover the discrepancy; or by putting the amount in a local suspense account, to be resolved or paid later[5]. Then the branch is again in balance, and can roll over to start the next TP.

131 To support this process, at the end of each TP, Horizon is required to provide the figures of system generated cash and stock; and if the SPM finds any discrepancy, to enable them to record how the discrepancy will be resolved; and when this has been done, to allow the branch to roll over and start the next TP.

132 Horizon must also support the activities of replenishing stock such as stamps, and of replenishing or remitting cash.

133 It must also support other administrative activities in the branch, such as enabling new staff to use the counter system.

134 The back-office settlement activity of Horizon may be illustrated in the case of a single client organisation, the DVLA. Across the UK in any day, the PO accepts a large amount of money from customers paying their road fund tax. All this money needs to be paid to DVLA. Therefore, PO has a back-office activity - carried out centrally - of summing all these amounts of money and paying DVLA. DVLA knows how much money it expects to receive in this way and checks the amount it expects against the amount calculated by PO. This cross-check is an example of reconciliation and supporting it and reflecting its outcomes are central to Horizon. Some kinds of reconciliation cannot be done as often as daily, because of variable time lags in the information available to clients.

[5] The facility for local suspense accounts has not always been available to SPMRs. It was removed for a period of time. We need PO to tell us when.

POL00029046

4. OLD HORIZON (1998 - 2010)

135 It is important to appreciate of the level of complexity of the Horizon requirements, and of the Horizon IT systems built to meet them. In a document 'Fujitsu's Systems and Operational Services to UK Post Office and the Worldwide Trend of Post Offices' Fujitsu have described Horizon as 'Europe's largest non-military IT contract', so Horizon is at the high end of complexity amongst IT systems. It represents many thousands of man-years effort in development and testing, and its documentation alone more than 100,000 documents. Sections 3- 6 of my report are intended to ensure that the readers understand those aspects of Horizon that most need to be understood to address the Horizon issues.

136 On the other hand, compared with the IT estates of various large organisations we have worked for (such as the NHS, Barclays Bank, UBS or RBS), the Horizon system is probably no more complex, and in some ways less complex. The banks' IT estates, like Horizon, have a complex corporate back end and an extensive branch office network. They were developed over a longer time period (30-40 years) using development team sizes similar to or larger than Horizon, often merging together or integrating the IT systems of previously independent organisations. This gave them a degree of legacy complexity, and design compromise, and corporate amnesia, not found in Horizon. There are parts of these IT estates which 'nobody dares touch'.

137 From time to time, when describing aspects of the Horizon architecture in the next three sections, I shall refer to various robustness countermeasures, which have been introduced in a table in section 2 of this report. This will help to describe the countermeasures by illustration and shows where they are built - into the Horizon architecture. In the table, the countermeasures have each been given a three-letter acronym such as RDS (Redundant Data Storage). It may be worth having a printed copy of the table to hand when reading these sections, to see both the acronym and the summary description of the countermeasure.

4.1 The Four-Level Architecture

138 In what follows, the words 'level', 'layer' and 'tier' all have the same meaning.

139 Nearly all complex IT applications are designed in levels or 'layers', to isolate different kinds of complexity in different layers, and to reduce the possibility of unwanted interactions between functions in different layers. A typical 'client server' layering structure includes at least a user interface layer, a business logic layer, and a data layer. The layering in Old Horizon is more complex than this.

140 The main purpose of defining an architecture in layers is to separate the functionality into parts in the different layers, with well-defined and simple interfaces between the layers. This not only makes each layer easier to design, build and test; but also, if there are errors not found in testing, it makes it easier to understand and isolate the cause of the errors by inspecting the exchanges between layers. Thus, a layered architecture is an important countermeasure for robustness; I have denoted it by the acronym 'ARC'.

141 The architecture of Old Horizon up to 2002 is described in the document TD/ARC/001 'Technical Environment Description' which is 476 pages long. This document states: *'The system architecture adopted to meet these requirements is not based on conventional client-server models. Nor does it conform to traditional central-system*

POL00029046

models. It adopts an entirely original and highly innovative four-tier model that effectively merges the qualities of central systems and client server systems.'

142 This architecture is explained in a diagram, which appears to have five layers rather than four. If the two boxes of 'Agents' and 'Correspondence' are counted as one 'Agents' layer, we get four layers, of:

- Counter
- Agent
- Host
- External Interface

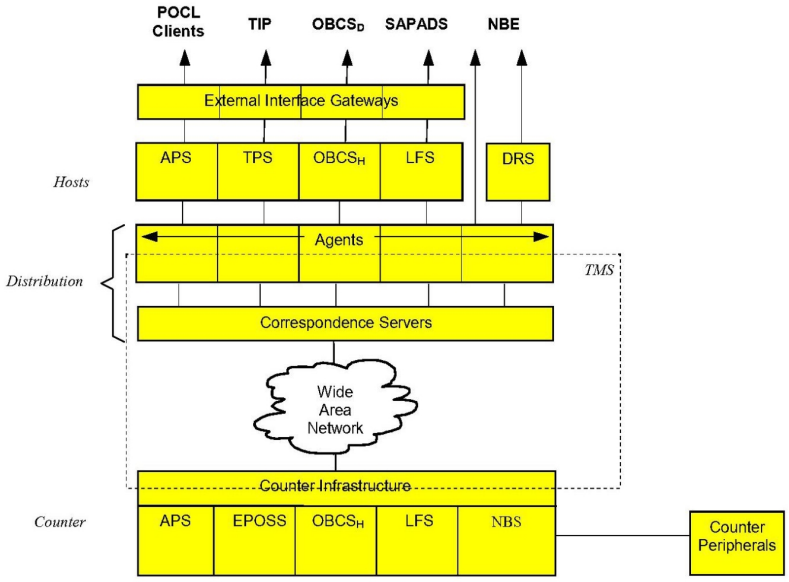


Figure 4.1 - Old Horizon layered architecture[6]

143 This apparently simple diagram hides a huge amount of complexity in Old Horizon, and that complexity is best described in stages.

144 The counter layer consists of all hardware and software in the branch. It includes all hardware and software required to support the counter activities required for all products and customer services offered in the branch. It will be described in the next sub-section 4.2, which will define the acronyms in the Counter layer of the diagram. In this section we will note one aspect of the counter layer: that it was largely built based on a commercial product, Riposte from Escher.

145 Riposte provided much of the Graphical User Interface (the basis of all user input and output at the counter) and provided a mechanism for secure distribution of messages between the branches and the two back-office campuses, which were located at Bootle and Wigan. This message distribution passed through the Wide Area Network in the diagram.

[6] From TD/ARC/001

POL00029046

146 The Correspondence Servers handled communication over the network.

147 The function of the Agent layer was to provide two-way translation of data between the formats used in the counter layer and the network (these formats were described by Attribute Grammars) and the formats used in the Host layer. The agent layer is also responsible for extracting the Audit of all data passing through the Correspondence Servers.

148 An Attribute Grammar is a way of describing a tree-like message structure in terms of its parts and their sub-parts. In more recent IT systems, tree-like messages are usually sent in XML (Extensible Message Language), with their structure defined in a notation called XML Schema. This is used in parts of Horizon Online. Because Old Horizon was developed before the use of XML became widespread, Attribute Grammars fulfilled this function in Old Horizon. I believe this is because the Escher Riposte product worked in this way at the time.

149 As well as reliable communication, Riposte provided a facility for reliable replication of data between the branches and the back-offices campuses. This means that if certain types of data were created at the branches, Riposte guaranteed that the same data would be available on the campuses - although if the underlying network was unreliable, it might take some time for Riposte to deliver this guarantee. Replication guaranteed that despite any network failures, no change to data made at a branch would be omitted at the campus or made more than once at the campus.

150 The bulk of the back-office functionality was provided in the Host layer, which will be described in section 4.3. Host applications were and are typically batch systems, processing data in large batches on a daily basis. A complex daily batch schedule was used to control the sequence and timing of these batch processes, using the Maestro scheduling product. The acronyms in the Host layer of the diagram above will be described in that section. It was the Host layer (and for most purposes, only the Host layer) which communicated with the IT systems of PO client organisations, through the External Interface Gateways.

151 There is an important simplification in the four-tier architecture. Each different business application in Old Horizon (typically tied to a different PO client organisation) can be regarded as a vertical 'slice' though the diagram and is largely independent of the other slices. It is intuitively obvious that different business applications (such as DWP Pensions, and Camelot Lottery) need have very little to do with one another (apart from being able to settle customer payments in the same basket - a facility provided separately from the applications). Therefore, the apparent complexity of some large Old Horizon architecture diagrams can be largely ignored when considering a single business application. This is another example of robustness through architecture (ARC).

4.2 Hardware and Software in the Branches

152 Although the hardware in the branches was not always reliable, and communications infrastructure at the time were not highly reliable, there were strong measures built into Old Horizon to ensure that hardware failures and communication failures could not adversely affect branch accounts. These measures are

POL00029046

described in section 6. They make up the robustness countermeasures of reliable hardware[7] (RHW) and robust data communications[8] (ROC). I shall therefore not spend much time here describing the hardware aspects of Old Horizon, either in the branches or the back-office campuses.

153 In the original Old Horizon architecture[9], sufficient data was held persistently in the branches, that a branch could continue to trade, and could support most business applications, even if the wide-area network was unavailable. Whenever the network became available again, Riposte data replication would ensure that the required data became available to the back-office systems. The only applications which could not run in this way were those that required some immediate validation from a client organisation - for instance, withdrawing cash from a bank account. A branch was able to hold all the data resulting from a Trading Period.

154 As will be described in section 5, with Horizon Online this was no longer the case. Persistent data was all stored remotely in the branch database - so that without a working network, a branch could no longer trade. More reliable network infrastructure by 2010 had made this a viable approach.

155 As well as supporting the business applications described in section 3, the software in the branches needs to support:

- Local user management
- Stock management
- Cash drawer management
- Balancing and reconciliation
- The production of local reports.

156 There had to be sufficient locally-stored data to support all these processes. To keep the counter clerk's view of all these applications consistent and simple, the user interface for all these local applications was provided by the Riposte desktop.

157 To describe how the branch layer of business applications was built on Riposte would involve a lot of technical complexity, most of which would not go to understanding the issues in the trial. I shall instead pick out some aspects which are relevant to the Horizon issues:

- **Zero-sum baskets for customers**: Whatever applications were invoked to serve a customer, the net impact of all the services provided for one customer was a sum of money which the customer was required to settle. It was required that the cash or other money produced by the customer should exactly match the cost of services provided; therefore, the whole basket of services and customer settlement had to be zero-sum, before the basket could be recorded in the branch (and then, through

[7] High Availability and Disaster Recovery: Concepts, Design, Implementation Hardcover by Klaus Schmidt, Springer Verlag 2006

[8] Data and Computer Communications by William Stallings, Pearson, 10th edition, 2013

[9] TD/ARC/001

POL00029046

Riposte replication, later recorded in the back-office systems). This was a necessary requirement on all business applications, because the impact of every business application would need at some stage (typically overnight) to be fed into PO's accounting systems, which operated by double entry bookkeeping. The only way to put postings into the accounting system was by double entry, which in turn could only be done for zero-sum baskets. This is the robustness countermeasure of double entry accounting (DEA)

- **Other branch actions had to be zero-sum**: Any other actions performed in the branches which had an impact on PO accounts (including stock management, cash drawer management, balancing and reconciliation) could only be carried out in the branch in packages of updates which were zero-sum, when summed across different PO account codes. This had to be the case, because the only way that the results could be posted to the accounts was to respect double entry bookkeeping - which is zero-sum across the accounts. This is another instance of the countermeasure DEA.
- **Transactional integrity**: All branch applications (including all customer business applications, balancing and reconciliation, cash management and stock management) were built so that any zero-sum package of updates from those applications would either succeed completely, or would fail completely and have no impact. This transactional integrity was enforced by the Riposte infrastructure, and I denote it by the robustness countermeasure with acronym TIN. Therefore, it was impossible in any event (such as hardware failure) for a part-completed set of updates to be recorded in the branch and then replicated to the back-office systems. This was necessary to prevent the accounting system from being subjected to non-zero sum updates, which would violate its double entry basis and cause later failures of its trial balances.

  The only exception to this principle was the so-called 'recoverable transactions' - where some irreversible interaction with a PO client system took place part way through a transaction - so it could not be undone in the case of a later failure. In these cases, the user on the counter would be guided through a short set of recovery steps, to produce a consistent zero-sum result which reflected what had happened. It was, of course, possible for the user to make some mistake in these steps, which may have been unfamiliar. In these cases, the mistake would be detected later by a reconciliation process, which would typically lead to a transaction correction. This robustness measure was a correction of user errors (UEC).
- **Applications driven by reference data**: Many of the business applications were not coded individually but were coded as generic applications which could be configured to run different specific applications by altering reference data. These were referred to in Old Horizon as 'soft-centred' applications. They had considerable benefits of adaptability and reliability over hard-coded applications (of which there were still a few). New applications can be built and deployed simply by providing reference data, rather than code. Errors could often be corrected rapidly, by simply correcting a piece of reference data. Reference data is much more concise and understandable than code, so it is much easier to create it or detect errors in it. Finally, any errors in the underlying generic

POL00029046

code would affect a set of specific applications, and so be easy to detect. This was the robustness measure of data driven software (DDS)

158 Reference data could be as simple as lists of available products and their prices (which clearly might change frequently), or might be more complex - for instance, to describe the sequence of steps needed to handle some business transaction, so that different types of business with similar but varying sequences of operations could all be handled by the same software, using different reference data.

4.3 Back-End Architecture

159 The three layers of the architecture which resided in the campuses at Wigan and Bootle were the Agent layer (which included the Correspondence layer), the Host layer, and the External Interface layer.

160 As has been described above, the role of the Agent layer was to manage communications and translate data between the representation used in the branches and the network on Riposte, and the representations used in the Host layer.

161 The main design document on Old Horizon[10] says:

'The systems at the Host Layer can provide permanent storage for information if required by the application's business rules. The Host systems can accept data from external Clients, and translate a file-based view of this information into discrete transactions or "messages". These are then passed to the Counters via the Agent and Correspondence Layers. Similarly, messages received from the Counters are translated back into a file-based view for transmission to the external Clients.'

162 Another description in the same document says:

'[Host systems] Servers run mainly large background batch processes and represent the part of the architecture that is responsible for the following functions.

- *Manipulating the information received from the External Client Systems into a form that is appropriate for the presentation mechanism and vice versa*
- *Applying business rules that are relevant to that information*
- *Storing non-transient information within the "Data Storage" component. This includes metrics needed for the computation of SLAs that may modify the payments due from PO Ltd for the achievement of key deliverables.*
- *Manipulating any such stored information'*

163 These descriptions only begin to describe the range of functions in the Host layer; to do more, I need to look at specific IT systems in that layer, aligned with different business streams. As will be described in section 5, many of these IT systems did not change with the introduction of Horizon Online in 2010.

164 One fairly simple diagram, which shows a number of important components of the back-office systems, is the following:

[10] TD/ARC/001

POL00029046

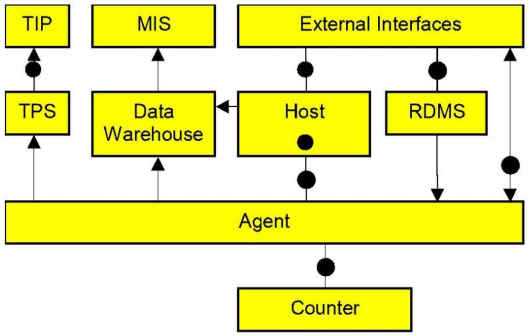


Figure 4.2 - Application components[11]

165 It is worth briefly describing those elements of this diagram which have not already been described as part of the four layers introduced in this section:

- **RDMS** stands for Reference Data Management System. As was described above, many business applications in the branches are driven by reference data, and this approach has many advantages over hard-coding of all the different business applications. It is much more flexible, to manage changes over time and across branches; and it is more reliable. However, this approach implies that the reliability of Horizon depends on the reliability of the reference data (much of which, for instance, is maintained by PO staff rather than by Fujitsu IT staff). Therefore, a dedicated IT application is needed to manage the reference data, and to distribute it appropriately to branches. This is needed for the robustness measure of data-driven software (DDS)
- The **Data Warehouse** consists of one or more databases, whose structure is designed to support flexible and open-ended querying and reporting by PO business staff, to help them understand the whole state of PO business from day to day. Many different kinds of information which pass through the host systems are siphoned off into the data warehouse and stored there in data structures designed for querying and reporting. In practice there appear to have been more than one data warehouse, used by both PO and Fujitsu staff. Functionality which depends on a data warehouse includes MIS (described next) and other applications such as 'data mining' to look for unanticipated trends and correlations in data. The data warehouse contributes to two robustness measures: redundant storage and computing (RDS), and Manual Inspection of Data (MID)
- **MIS** stands for Management Information System, the component built on the data warehouse to provide PO staff with the flexible access to information about all aspects of PO business. The data warehouse and the MIS are an important part of the checks built into Horizon. In cases of human error in business processes, operational errors in managing PO business on Horizon, or software errors in Horizon, some resulting discrepancy or aberration will be rapidly visible through the MIS. MIS facilities were also used by Fujitsu staff. Many pairs of eyes are inspecting the outputs of the MIS, in hundreds of different reports or spreadsheets. One purpose of this is to ensure the rapid

[11] From TD/ARC/001

POL00029046

detection and correction of many types of errors. These include software errors. So, the MIS also contributed to the robustness measures of RDS and MID.

- **TPS** stands for Transaction Processing System. The purpose of TPS is to 'harvest' all types of transaction taking place in the branches, and to pass them on to other IT systems in the Post Office - initially to TIP, and later to POL FS.
- **TIP** stands for Transaction Information Processing, which in 2003 (the date of the diagram above) was the gateway to all other PO data processing, including accounting. After 2004, PO accounts were held on a SAP system, POL FS; so TPS passed data to POL FS, rather than to TIP.
- The black circles denote points *'at which ownership of data conceptually changes and hence at which audit information is generated'*. Audit is addressed in section 4.4 below, and is part of the robustness measure SEK (secure kernel)

166 To quote the 2003 design document[12]: *'One essential task that can only be carried out at the Host layer is reconciliation. The Host is the only system component that can detect discrepancies between the transactions carried out at the Counter (and hence reported back to PO Ltd via TPS), and those that were authorised or expected. It should be in a position to send reconciliation reports back to its Client. These enable the discrepancy with the TPS records to be identified and resolved.'*

167 This reconciliation, carried out in the Host layer, is an essential element within Old Horizon for detecting and correcting errors made at the counter (robustness measure UEC). Reconciliation and Transaction Corrections (TCs) are described for both Old Horizon and Horizon Online in section 6.

168 Reconciliation and TCs, which often have the effect of correcting human errors, also have the effect of detecting and correcting the effects of many possible software errors. If there were any such software error, it would probably occur with such high frequency, and occur uniformly across all branches, giving rise to so many TCs, that the PO would soon suspect a software error (for instance, seeing the effect repeatedly in some MIS report) and require Fujitsu to correct it.

169 The likelihood of any software error in Horizon staying disguised as a human error, and thus of not being detected, is extremely small; and even if it were not detected, by the analysis above, it would have no distorting effects on branch accounts.

170 The PO has several hundred different client organisations[13], and so there are different types of reconciliation which may be carried out. In my opinion, it would be extremely unlikely for any large client organisation to appoint the PO as agents for any other kind of financial transaction such as bill paying, without requiring some check that PO was paying them the correct amounts of money. So, this kind of error detection and correction is used for the vast majority of money that passes through PO branches -

[12] TD/ARC/001

[13] TD/ARC/039

POL00029046

for all of its agency business. It combines the robustness measures of redundant data storage (RDS) and user error correction (UEC).

171 Host applications fall into one of three classes:

- Complex applications that require a large amount of persistent storage, with high volumes and/or high transaction rates. These generally have their own Oracle database and are located on one of the Host Central Servers. They incorporate the robustness measure of transactional integrity and database recovery (TIN).
- Less complex applications, with little persistent storage requirement. These may run on the Host Central Server, or on a Host Ancillary Server, an Intel Platform running under Windows NT Server. Oracle or Microsoft SQL Server can be used to provide the database functionality and storage mechanisms. They still benefit from the TIN robustness measure.
- Simple applications that have no requirement for a persistent database may be implemented on a dedicated Intel-based Host Ancillary Server running under Windows NT server. Typically, these generate or process tabular files of text and numbers.

172 I shall mainly consider the first type of host application, which are responsible for the great majority of the money passing through PO branches.

173 Other system diagrams of the Host layer are complex, showing many distinct systems, and there is no 'universal' diagram which is suitable for explaining every issue.

4.4 Audit Information

174 The Horizon system includes an audit database[14], which is an accurate and immutable record of any activity which can affect the branch accounts. In the event of any discrepancy arising anywhere in Horizon (for instance, due to a bug in some other Horizon application, or some operational error in running a batch process, or a dispute about what data was entered at the counter) it is possible to compare other records - for instance, records extracted from other applications, or the data warehouse - with the audit records, which are guaranteed to be an accurate record of what was entered into Horizon at the counter. In this way many kinds of error can be traced and corrected. Audit records are normally retained for seven years, as required by the PO contract with SPMs. The audit database is a robustness measure of a secure kernel (SEK) which also involves redundant data storage (RDS).

175 It is important to understand that many measures were used to ensure the integrity of the audit data. A slide set produced by Fujitsu[15] describes this well, and I will summarise the main points here. This can be done by following the sequence of operations by which data travels from the counter to the audit database.

---

[14] TD/ARC/001

[15] 'Horizon Core Audit Process' 30/01/2014 (has 2 enclosed word documents for Horizon and HNG)

POL00029046

176 In Old Horizon, all data travels from the counter through the software application at the branch, through Riposte data replication to the two campuses, then through the Audit Agent to the Audit Store. This is shown in the diagram:

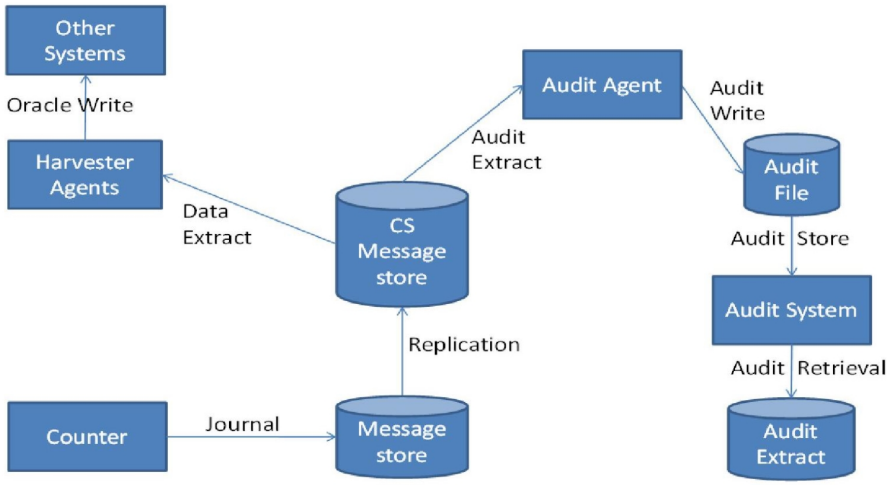


Figure 4.3 - Old Horizon audit data flow[16]

177 While this section 4 is devoted to Old Horizon, I shall digress for a moment to the later Horizon Online, to compare the facilities for audit in the two systems.

178 In Horizon Online, all data travels from the counter through the software application at the branch, through communications hardware and software to the Branch Access Layer (BAL), into the Branch Database (BRDB) and then nightly to the audit store[17].

[16] From 'Horizon Core Audit Process' 30/01/2014

[17] ARC/SOL/ARC/0001 Horizon Solution Architecture Outline

POL00029046

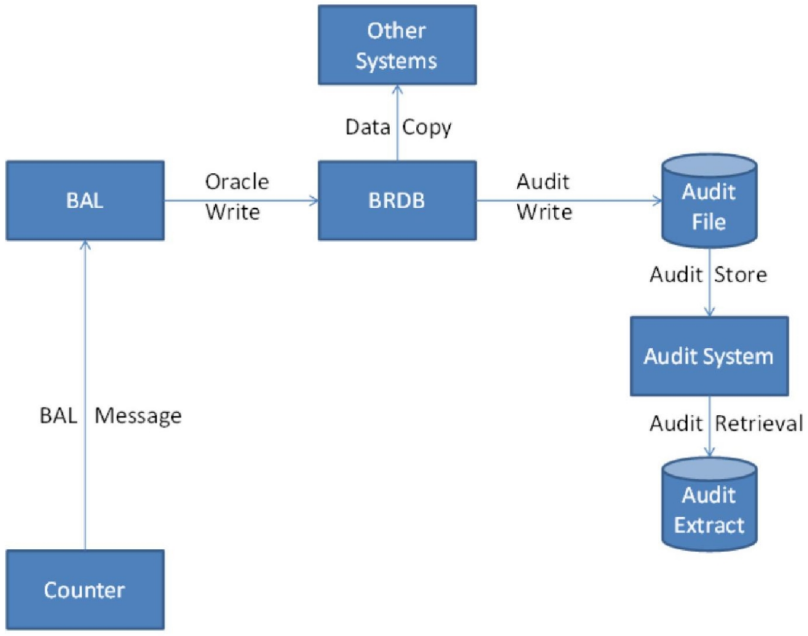


Figure 4.4 - Horizon Online audit data flow[18]

179 The principles ensuring the integrity of the audit data are the same in both cases of Old Horizon and HNG:

- When a user signs on at a counter, the password he or she provides is used to create cryptographic keys, which are used to encrypt all messages sent over the network to the BRDB, and subsequently used to create digital signatures on the audit records. Thus, any audit record is digitally signed in a way that proves it could only have originated from a certain counter, and that it has not been modified since it left that counter.
- As the counter clerk provides one or more services to a customer, these services and the money paid for them in settlement by the customer are collected in a basket whose monetary sum must be zero. At all stages on the journey of this basket to the audit record, there are checks that it has a zero sum. This is typically not just a check that two numbers are equal and of opposite sign; it is a check that several numbers add up to zero. Thus, any failure in hardware or software, which affects one or more of the numbers, is most likely to destroy the zero sum; if the zero sum survives and the record is stored in the audit database, all the numbers in it are an accurate record of what happened at the counter.
- All baskets are given a journal sequence number (JSN) which must ascend in increments of 1, with no gaps or duplicates. This ensures that no gaps or duplications are introduced in the baskets from

[18] From slide set 'Horizon Core Audit Process' 30/01/2014

POL00029046

any counter, for instance by communications failures or recovery processes. No extra baskets can be introduced without destroying the sequence. All audit entries are time-stamped.

- In communication, data replication, and in storage in any database, principles of transactional integrity are applied. This means that a basket is either stored in its entirety, or no part of it is stored. If it is not stored, appropriate information is sent to the branch, and recovery processes initiated.
- Digital signatures and seals such as MD5 hashes[19] on the audit records ensure they cannot be altered once they have been created.
- Recovery procedures are designed so that should any of these checks fail (e.g. in the event of a hardware failure at the counter), appropriate remedial steps are taken, and the integrity of the audit is preserved.

180 In my opinion, these integrity measures are well designed.

4.5 Changes During the Period 2000 - 2010

181 A series of significant changes were made in Old Horizon during the period 2000 – 2010. Each new application typically required changes at the branch and at the campuses. Some changes were superseded by later ones. Some important changes were:

- In 2003 the Data Reconciliatio Services (DRS) and debit card processing were introduced.
- POL FS was introduced around 2004.
- In 2005, Pension & Allowance Order Books were replaced by the Post Office Card Account which necessitated building banking services into Old Horizon. PO had always had business relationships with banks including Girobank.
- AP/ADC was introduced around 2007/2008.
- Around 2010, POL FS and SAP ADS were merged to make POL SAP.

[19] An MD5 hash is a short string, computed from the entire contents of a file, which will change if the contents of the file are changed in any way.

POL00029046

5. HORIZON ONLINE (2010 - PRESENT)

5.1 Motivation for the Move to Horizon Online

182 Horizon moved from the previous Riposte-based architecture to Horizon New Generation[20] (HNG - which I shall refer to as 'Horizon Online') in 2010. At this time, there was no sudden change in the range of business applications supported by Old Horizon in the branches. This range of applications has increased continually over the lifetime of Horizon and Horizon Online.

183 There were several motivations for the change from Old Horizon to Horizon Online. The main driver was to exploit advances in the underlying communication technology, and improvements in its reliability - which meant that it had become possible to store all persistent data at the centre rather than in a branch, with the consequence that a branch could only operate when communications were available - but the risk of failed communications was by then so low as to be acceptable. This change mirrored the wider changes across the IT industry, where increased reliability of communications infrastructure means that applications can now be 'cloud-based' (entirely dependent on remote data, stored by some cloud provider such as Amazon; and dependent on remote functionality in the cloud) and therefore simpler to deploy and manage.

184 The centralised storage of transaction data allowed several changes and improvements:

- A simplification and rationalisation of the architecture in many respects (just as most cloud-based applications are now simpler than their antecedents);
- Simpler management of the branches in the event of hardware failures or replacements and other events, because in those cases branch data would not be lost and did not need to be recovered;
- No dependence on Riposte data replication, which meant that Riposte could be removed entirely, and all applications could be supported by more modern software technology.

185 These, rather than any change in the business applications to be supported, were the motivations for the move from Old Horizon to Horizon Online.

186 The document 'Counter Business Architecture' (ARC/APP/ARC/0009) states that:

> *'The objective of the HNG-X programme is to develop a system with structural and operational characteristics that substantially reduce ongoing support and maintenance costs with respect to the current Horizon system.*
>
> *The overall requirement is that the business capabilities offered by the current system (Horizon) are preserved in the new system (HNG-X). However, a limited number of business capabilities will be revised based on a joint optimisation of business requirements and system properties.'*

5.2 The New Division Between Branches and the Back End

187 The fundamental change was that in Horizon Online, no transaction data was held in any persistent form in the branches. The Counter Business Architecture document explains the rationale for this:

---

[20] ARC/SOL/ARC/0001 Horizon Solution Architecture Outline

POL00029046

'The analysis of the serviceability profile for Horizon has highlighted data management as one of the most significant drivers for cost. The storage of transactional data within counters causes the need for security mechanisms that impact both the structural complexity and the operational performance of the Counter Business Application. In addition, the presence of sensitive data on the counter increases the time, complexity, and ultimately the cost of maintenance procedures.'

188 In Horizon, on completion of a basket of customer services, that basket was held locally in the branch in the Riposte message store - until Riposte could replicate it to the campuses at Bootle and Wigan, which might have been hours or days later, depending on the state of communications. Meanwhile, the branch could continue to function for many types of transaction. However, the number of applications such as bank withdrawals which required immediate confirmation from a third party, and therefore could not function in the absence of communications, had steadily increased.

189 In Horizon Online, before completion of a basket of customer services, that basket was transmitted and had to be acknowledged by the Branch Database[21] (BRDB)[22]. The basket could not complete successfully at the counter until that had happened - so Horizon Online could not operate in the branches without working communications.

190 Because the branch was no longer responsible for persistent storage or replication of transaction data, the architecture within the branches was simplified.

191 The main difference at the back end was the existence of the Branch Database, which was the main persistent store of all transactions for all branches. Many business applications in the back end were unchanged (and were referred to as 'legacy'[23]), except for the need for them to interface with the BRDB rather than with the previous Agent layer. Other copies of transaction data continued to be stored in those applications.

5.3 New Architecture in the Branches

192 The previous branch architecture had been based on Riposte, which provided functionality on many levels (including for instance user interfaces, some business applications, and message storage and replication). In Horizon Online, Riposte was completely removed; therefore, all elements of the branch software were replaced[24].

193 Whereas much of the Old Horizon branch code had been written in Visual Basic, for Horizon Online nearly all the branch software was written in Java - a newer language with good support for modern programming paradigms such as object orientation and service-oriented architecture. This allowed a more modern and elegant software architecture in the branch, which did not have to be fitted around the architecture of Riposte.

---

[21] ARC/SOL/ARC/0001 Horizon Solution Architecture Outline

[22] DES/APP/HLD/0020 Branch Database high level design

[23] In IT, the term 'legacy' is used to refer to older technology, which may have been superseded. For Horizon, this means the original generation before Horizon Online was implemented in 2010.

[24] ARC/APP/ARC/0009 HNG-X Counter Business Applications Architecture

POL00029046

194 The many modern object-oriented features of the Java language provided better support for several robustness features, such as defensive programming (DEP).

195 A view of this architecture is shown in the following diagram:

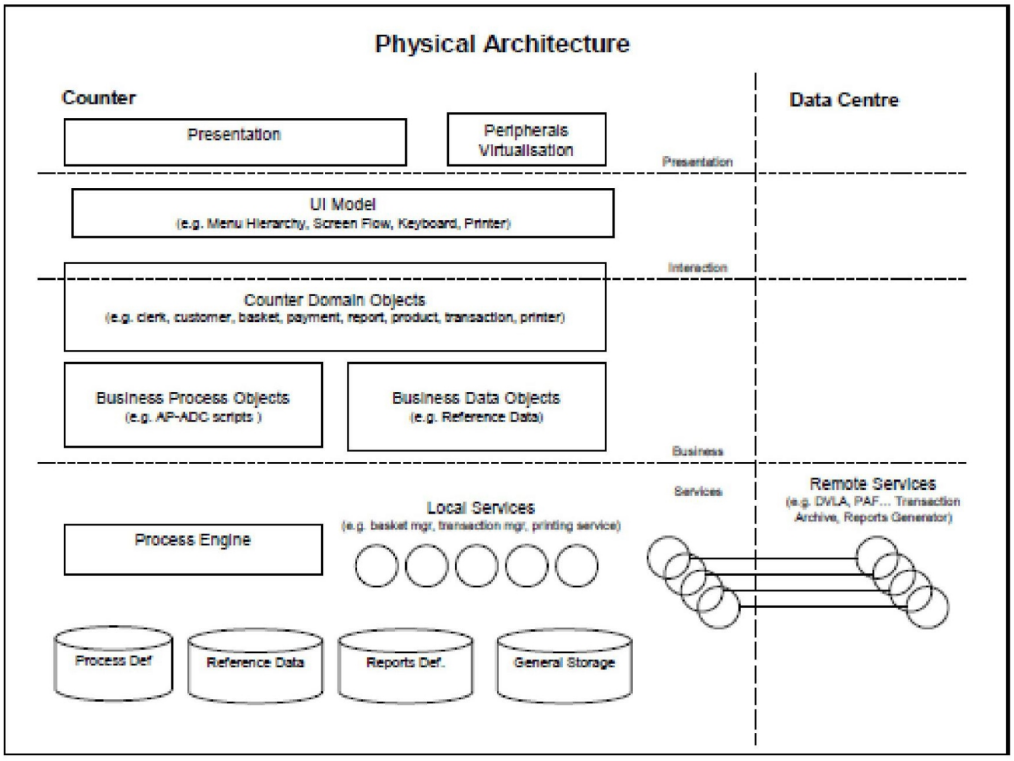


Figure 5.1 - Counter application architecture[25]

196 The top 'Presentation' layer is responsible for displaying information to the users and for accepting their inputs. The next 'Interaction' layer provides the building blocks for this interaction, such as menus. The effect of these two layers is to provide a user interface similar in style to that which had been provided by Riposte in Old Horizon - to make the user experience similar to what it was in Old Horizon, but using Java technology, rather than Riposte, to build it. These two layers were largely responsible for the early detection of user errors (DUE).

197 As had been the case for Old Horizon, the layered architecture of the counter software in Horizon Online, with a clean separation between layers, was an important means of providing architectural robustness (ARC).

198 The 'Business' layer provides the functionality of the many business applications, in an object-oriented fashion. This means that there are several general-purpose software objects (i.e. modular blocks of software) with names such as customer, basket and payment, which represent the required behaviour of those entities in the real world, in a way that can be easily reused in many different business applications.

[25] From HNG-X Counter Business Applications Architecture (ARC/APP/ARC/0009)

POL00029046

The reuse of core design elements for many applications was another example of architectural robustness (ARC)

199 The Business Process objects and Business Data objects are more specialised to support the many business applications. As their names imply, the Business Process objects support the sequence of steps which make up a business process, and the Business Data objects hold the necessary data, which is presented at the counter or stored. However, this is generally not done by writing completely different software for each business application (i.e. for each type of service that can be offered to a customer). Many applications are driven by reference data[26], such as data which defines the sequence of steps in completing each type of service for a customer. This reference data-driven style of software is common modern practice and is effective in making software easier to write and test. New applications can frequently be supported just by adding new reference data, rather than by writing new software. This was intended to achieve robustness through the use of generic, data driven software (DDS)

200 For instance, all automated payment (AP) applications are provided in this reference data-driven manner. This makes it very easy to build and test a new AP application, for a new client organisation.

201 The Counter Business Architecture[27] summarises the capabilities in the business layer.

'In summary the set that are provided by the Counter Business Application are:

- *Point of Sale Capability;*
- *In / Out Payment Capability;*
- *APOP Facility;*
- *Banking Capability;*
- *DVLA Licensing Capability;*
- *Electronic Top-Up Capability;*
- *Bureau de Change Capability;*
- *Postal Services Capability;*
- *Generic Online Capability;*
- *Payment Management Capability (Cash, Cheque, Vouchers, Debit or Credit Cards);*
- *Cash and Stock Management Capability;*
- *Branch Management Capability (Stock Unit Balancing, Branch accounting, Branch Reports, Reversals and Refunds, Transaction Corrections);*
- *Branch Administration Facility (User Log On / Off, User / Password Management, Stock Unit Creation / Allocation, Provision of Secure Inactivity Time-Out Facilities, Generic User Help System),*

[26] ARC/SOL/ARC/0001

[27] ARC/APP/ARC/0009; DES/APP/HLD/0047 (HNG Counter High Level Design)

POL00029046

- *Branch Support Facility (Sales Prompts, Bulk Input of transactions, Reference Data, PAF, Message Handling, Audit and Training).'*

202 Just as the Presentation layer does, the Services layer provides a set of software objects which provide services in support of many business applications. Most of these services are not to do with the user interface but help in organising information and sending it for storage in the BRDB.

203 For instance, the facilities for double entry bookkeeping (ensuring that each basket is zero-sum before it is sent) and transactional integrity (ensuring that a basket is either sent and stored in its entirety, or none of it is stored at all) are provided generically in the services layer, and so do not need to be coded individually in the business objects. This design practice helps to ensure that the powerful checks of transactional integrity and double entry bookkeeping (robustness measures TIN and DEA) are applied universally, and do not have to be built individually into any new business application.

204 One key component of the services layer is the Process Engine. This provides a simplified way for the counter to provide services which involve a sequence of steps. The sequences of steps need not be defined in Java code but are defined in a specialised Process Definition Language (PDL), which is executed by the Process Engine. PDL was developed for Horizon Online by Fujitsu. The use of PDL means that complex sequences of steps are much simpler to define and test. This is another example of generic data-driven software (DDS)

205 The disc-shaped boxes in the services layer in the diagram above show that some data are stored persistently on the branch hardware; however, these data do not include customer transaction information. They include business process definitions (definitions of sequences of steps in a process), other reference data, data defining reports that can be output in a branch, and other information required to support operations. The reference data is refreshed daily from the data centre. There are services which provide these data to the other layers in forms that are convenient for them to use.

206 As can be seen from the diagram, the Services layer of the branch architecture is the only layer which communicates with the data centre, through the communications subsystem. Individual services provide reliable and robust communication for various types of information(this is the robustness measure ROC). The purpose, as always, of this layered approach is to provide each kind of functionality (such as reliable and robust communication with the data centre) in one layer only, and not have to reinvent it for many different business applications. In effect, the services layer in Horizon Online now provides many of the services which were formerly provided by Riposte.

207 The services layer also provides interfaces for online services, where to provide some service at the counter, it is necessary to contact some non-PO IT system. These online services include:

- Banking
- Credit / debit cards
- Mobile phone E-Top Ups
- DVLA online

POL00029046

- APOP services, such as postal orders
- PAF lookup
- Generic Online Services
- Some types of PIN Pad accesses (WSPOS[28]).

5.4 Back-End Architecture: Changed and Unchanged Elements

208 The two completely new elements of the Horizon Online back end are the Branch Access Layer and the Branch Database[29].

209 The principal function of the Branch Access Layer (BAL) is to exchange messages with the counter software in the branches. However, the BAL goes well beyond the mere exchange of information, into checking that the information is conformant (for instance, that each basket is zero-sum, applying the DEA robustness measure), logging of all exchanges, and recovery from many kinds of error conditions. Because it has to handle more than 25 million transactions per day, the BAL has many design features to ensure high performance (principally by distributing the load in parallel across many machines), as well as robustness - for instance, through reliable and redundant hardware (RHW).

210 The Branch Database[30] (BRDB) is a large, high-performance Oracle database whose main function is to store all customer transactions which originate in any branch. It, too, has many features to ensure high performance and robustness, for instance through transactional integrity and recovery (TIN).

211 The types of data held in the branch database include:

- Customer transaction data, including both internal counter transactions and external client transactions
- Reference data to be distributed to branches
- Data that applies only to individual branches, such as users, stock units and messages
- Branch report data
- Recovery data
- Journal data
- Postal address data.

212 The logical sub-divisions of the branch database are shown below:

---

[28] Web Services POS (Point of Sale)

[29] ARC/SOL/ARC/0001

[30] DES/APP/HLD/0020 Branch Database High Level Design

POL00029046

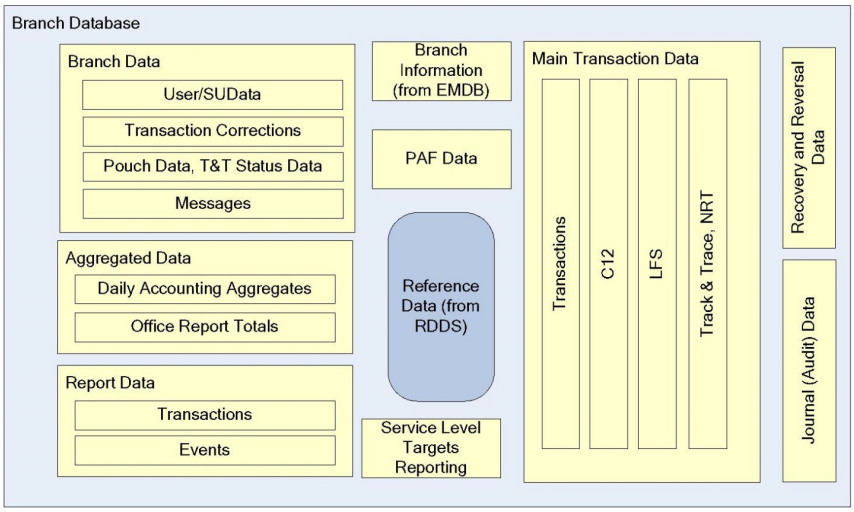


Figure 5.2 - Logical subdivisions of the Branch Database[31]

213 The architecture of the Horizon Online data centre is shown in the next diagram:

[31] From ARC/SOL/ARC/0001

POL00029046

CHARTERIS

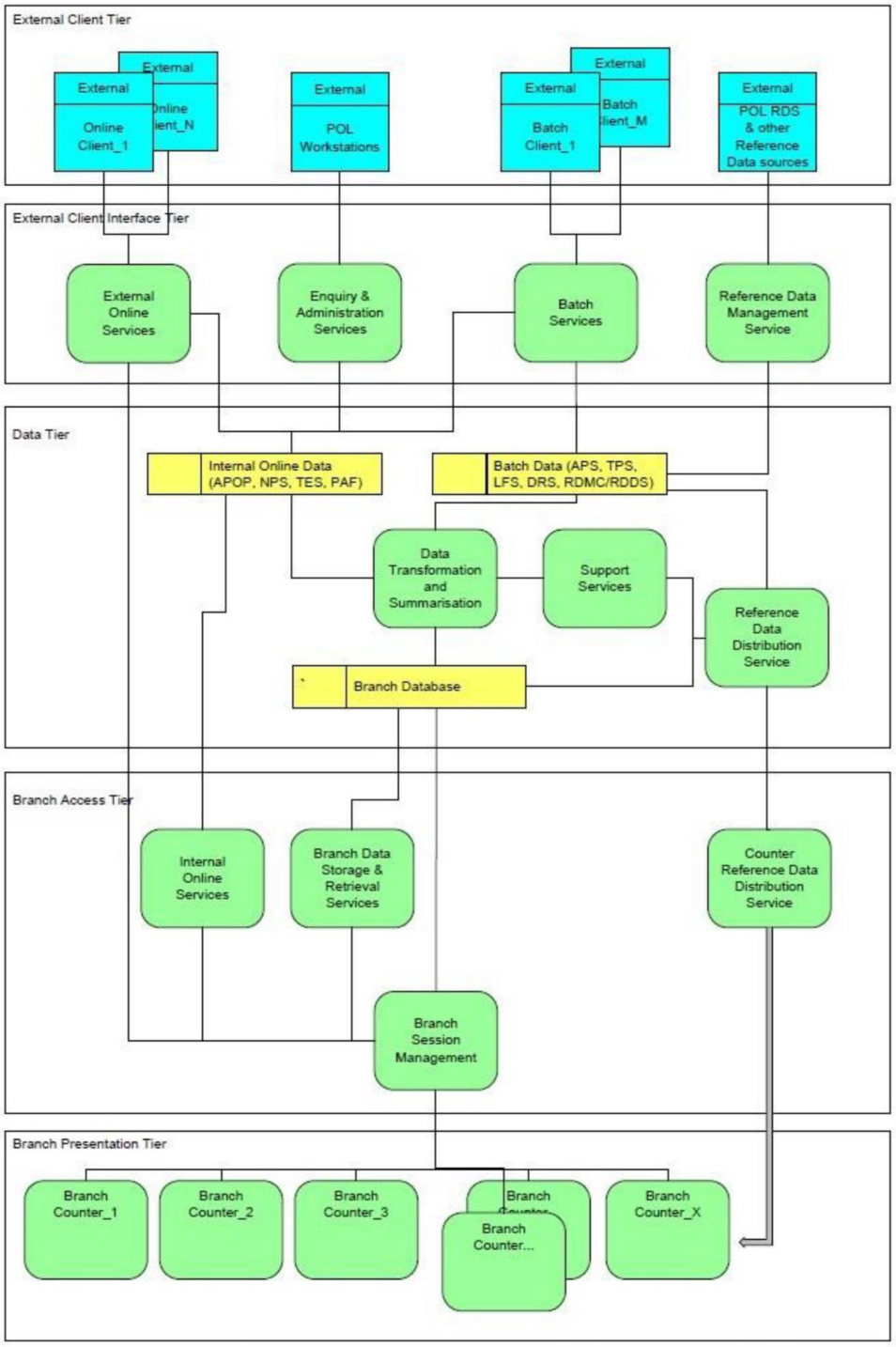


Figure 5.3 - Horizon Online data centre application architecture[32]

[32] From ARC/SOL/ARC/0001

POL00029046

214 From the bottom of this diagram upwards, the Branch Presentation Tier is the branch software, discussed in the previous section. The Branch Access tier (or layer) has been described above, as has the Branch Database.

215 The many layers and the defined interfaces between the applications - rather than a few monolithic applications) are all applications of architectural robustness (ARC)

216 The data tier, which includes the branch database, is a data-oriented view of the business applications, and other functionality. The External Client Interface Tier provides interfaces in both directions to external client IT systems, pictured in blue in the top layer.

217 Most of the complexity of Horizon Online occurs in the Data Tier and the External Client Interface Tier, which together do all the back-office processing for all the different applications (several hundred of them) supported in the branches. Because the PO has several hundred client organisations, and each one of them may have differing requirements for back-office processing such as settlement and reconciliation (depending on their own differing IT systems), there are at least several hundred kinds of back office processing to be supported. While many of these have strong similarities between them, the differences between client organisations cannot be entirely removed by the External Client Interface Tier; but many of these differences can be handled by reference data (DDS) Many of these applications are batch applications, harvesting transaction data from the BRDB and running once per day in a complex batch schedule.

218 The documentation provided by Fujitsu includes many different 'wiring diagrams' of these back-office applications - each from a slightly different perspective, emphasising some aspects and abstracting out , or omitting, others. Because the full picture is so complex (with probably several hundred boxes, and many more lines between them), it is very hard to provide the reader with simplified and useful views which may not need to be revised and amplified later for specific purposes. However, the following data-oriented diagram gives one useful view:

POL00029046

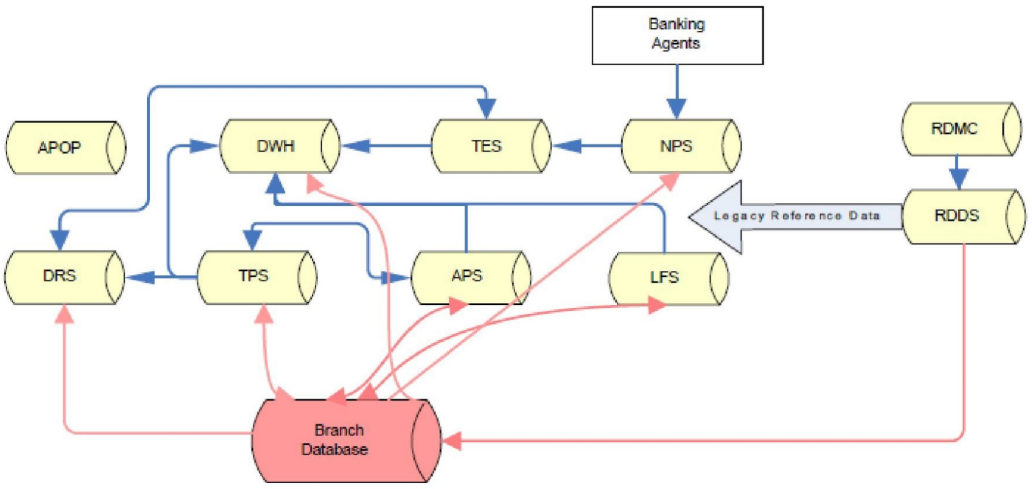


Figure 5.4 - Application Database Architecture[33]

219 This diagram shows the central role of the Branch Database (shown in pink) and a number of so-called 'legacy databases' shown in pale pink which survived unchanged from Old Horizon. The acronyms for the legacy databases are as follows:

- APOP: Automated Payment Out-pay Database
- APS: Automated Payment Service
- DRS: Data Reconciliation Service
- DWH: Data Warehouse
- LFS: Logistics Feeder Service
- TES: Transaction Enquiry Service
- TPS: Transaction Processing Service
- RDMC: Reference Data Management
- RDDS: Reference Data Delivery Service

220 There are of course omissions from this diagram - including, for instance, the Audit database, which continued in its previous role as described in the section 4.4 above, storing the same data in the same form as before, but now taking its information from the branch database - and providing robustness through a being a secure Kernel (SEK) with redundant storage of data (RDS).

[33] ARC/SOL/ARC/0001

POL00029046

6. ARCHITECTURAL TOPICS ACROSS OLD HORIZON AND HORIZON ONLINE

221 In this section of the report, I survey some architectural topics, bringing out particularly how those features of the architecture support the various robustness countermeasures[34], first introduced in section 2 above - as an introduction to the main discussion of robustness, in section 7. Here I introduce these topics to highlight and explain the robustness countermeasures as briefly as possible. Supporting detail is provided at Appendix C, the structure of which follows the sub-sections of this section 6.

222 My intention in section 6 is to introduce and describe the countermeasures built into Horizon - to describe how they were implemented, rather than give my opinions on how well they were implemented. My opinions on 'how well' are given in 7.6, when addressing Horizon Issues 3,4, and 6. Inevitably, however, the 'how' of section 6 has strayed into the 'how well', so section 6 gives some previews of my opinions in section 7.

6.1 User Error Detection and Prevention

6.1.1 Detection of User Errors (DUE)

223 In the design of the Horizon counter user interface, there are large numbers of measures to prevent user errors. Many of these measures have by now become common practice in the design of user interfaces - such as the use of menus and buttons, rather than free text input, to allow the user at any time only to choose one of the actions or inputs which are allowed at that time, or the use of facilities for inputting numerical values, which only accept numbers (not characters) in an allowed range, and confirmation buttons to ensure that the user really intended to take the action he chose. These are all cases of the robustness measure Detection of User Errors[35] (DUE).

224 Among these measures is the check that when a customer makes one or more payments for a basket of items or services, the sum of the payments entered must be the same as the summed cost of the items purchased; the basket cannot be concluded unless it is zero-sum[36]. Also, in all possible cases (such as credit card payments) capture of the amount paid is automatic rather than manual, preventing any user error. This form of robustness is a combination of DUE and Double Entry Accounting (DEA) - in that each basket is a zero-sum set of items - a set of entries which cancel one another.

225 The claimants have drawn attention to the user error of 'mis-keying' , to the question of how well Horizon prevented mis-keying, and whether Horizon might have prevented it more effectively.

226 In building and deploying Horizon, PO and Fujitsu face particular challenges which are not faced by most retail point-of-sale systems.

---

[34] Anti-fragile ICT Systems, Kjell Jørgen Jørgen Hole, Springer 2006

[35] The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques by Wilbert O. Galitz, Wiley 2007

[36] Slide set 'Horizon Core Audit Process' 30/01/2014

POL00029046

227 The first challenge is the very large volume of transactions, which (as I describe in section 8.5) is of the order of 6 million transactions per day across the PO branch network. If even a tiny fraction of these were in error through mis-keying, and those errors involved PO central support costs in correcting the errors, then the central support costs would be significant. Some of these costs are quantified in a PO Internal feasibility study[37]. I make some further comments on that study and the costs in Appendix C.2.

228 The second challenge was that large sums of money were often involved - but because of the PO's very small profit margins on these sums, a high degree of precision was required. A 'Horizon Architecture Overview' in 2006[38] describes this:

> *Within the branch estate, the majority of the products that are sold by Post Office are on behalf of a third party (a "Client" in Post Office language) – for example payment of a British Gas or BT Bill. The fees paid by the Client for this service are typically related to the amount of manual work that needs to be undertaken by branch staff rather than the value of the transaction – resulting in very low margins (Post Office's turnover is approximately 1% of the £110 billion worth of transactions it handles each year and its margin is a low percentage of this).*
>
> *One consequence of the low margins is that Post Office has to be extremely careful to minimise the impact of any errors or faults in the solution. One example of this is that for online authorisations every individual transaction is reconciled with the third parties view and all errors are investigated (typical retail organisations would just check that the total for the day is accurate to within an agreed error margin with the third party).*

229 For the PO, a very small profit margin requires a very small margin of error in data entry. For a typical retail organisation, if it did transactions worth £110 billion per annum, the gross margin on those transactions would be many billions of pounds - giving plenty of budget to sort out data entry problems. For PO, this was not the case.

230 For many retail organisations, prices of goods are captured automatically, through barcodes and reference data that links barcodes to prices. PO is also peculiar in the high proportion of transactions for which the monetary amounts have to be keyed in.

231 For all these reasons, the requirements for detection of user errors in Horizon were very stringent. Horizon needed to have the countermeasure DUE built into its user interface, much more so than most retail point-of-sale systems. If it did not, in my opinion the costs would fall more heavily on PO centrally than on the individual branches.

6.1.2 User Error Correction (UEC)

232 Despite these measures, there remain cases where the amount of cash entered into Horizon (which gives a balancing basket) is not the amount actually put in the till; or where the amount of stock given out, such as stamps, is not the same as that entered in Horizon. There is in principle no way in which Horizon could detect or prevent many of these user errors. They are errors made outside Horizon, in the handling of

[37] POL-0215788

[38] TD/ARC/039

POL00029046

cash or stock. So, they can only be trapped by later error correction measures (User Error Correction - UEC). These measures are powerful and, in the absence of later errors made in the correction process itself, will eventually correct these sorts of user error. The delay involved in 'eventually' will be discussed below.

233 Through these measures, many kinds of errors in entering amounts of cash or stock in daily trading are prevented or are corrected after some delay. It is hard to think of any such error in counter transactions which is not caught in one of these ways. This is necessary, because, as we have seen, there are probably several thousand such errors made at the counter every day.

234 User errors made in stock taking or monthly balancing have a similar effect; they get corrected eventually. For illustration, suppose, during monthly balancing, the subpostmaster mis-counts some item of stock; so, for instance, he thinks the stock is in balance with Horizon, when it is not. Horizon thinks that the stock is in balance, with a physical stock of X units. But in fact, the physical stock is Y units. The subpostmaster could have counted it as Y units, and then made up the discrepancy (X-Y) in cash; but he did not. Then, over the next month, the changes in stock (as recorded in Horizon, and in fact) are Z units. At the end of the month, Horizon thinks that the stock should be (X+Z) units. But the physical stock is actually (Y + Z) units. So, the discrepancy, which is (X-Y) units, still has to be made up at the end of the next month. The effect of a user error in balancing in one month is just to postpone the need to balance, for another month. After balancing correctly in that month, the accounts will be accurate again.

235 All of these robustness measures are part of the essential countermeasure of User Error Correction (UEC)

236 There are possible user errors in recovery situations. In practice, because these situations occur more rarely than typical counter transactions, and are often more complex and unfamiliar to them, they are more prone to user errors.

237 A transaction that could not be processed correctly owing to a system failure may be a 'recoverable transaction', where some action is needed to bring the transaction to a correct final state, which records what actually took place. A recoverable transaction[39] occurs when some irreversible interaction with an external agency, such as an authorisation of a payment by a bank, occurs at some stage during a customer basket, and the basket later fails for some reason (such as a hardware or communication failure). Then typically some action is required from the counter staff to 'recover' the transaction to a consistent state. User errors may occur during this recovery process - which is less familiar than the normal operation of Horizon. In these cases, typically the error is trapped later in a reconciliation with the external party and is corrected by a TC.

238 It is necessary to understand the basis of these measures for user error correction (UEC) and to understand that they are successfully used many thousands of times in a year, across the PO branch network. They have not only been designed into Horizon; they have been tested in live use many times over.

---

[39] ARC/APP/ARC/0008, BRDB Architecture

POL00029046

6.2 Intrinsic Error Prevention

239 Intrinsic error prevention includes the following techniques:

a) Double entry accounting (DEA), which ensures that any numerical error affecting only one part of an accounting transaction will destroy a trial balance and be rapidly detected.

b) Transactional integrity (TIN), which ensures that in many cases, partial updates to databases, which would destroy their integrity and consistency, cannot happen.

c) Measures designed to detect or correct user errors (DUE). These are so important that they have been described separately in section 6.2.3 below.

d) Defensive programming (DEP), where small parts of a program are written to assume that other parts of the program may be in error and are written to always check their inputs for the presence of errors.

e) Redundant storage of data (RDS), where the same information is stored repeatedly and in different forms in distinct parts of the IT estate, with consistency checks on versions of the same data. These checks include arithmetic checks of monetary sums, and many manual inspections of data (MID).

f) The audit system provides a highly secure and tamper-proof record of what is entered into Horizon at the counter, which can be used, in cases of any anomaly, to provide a 'gold standard' for comparison with data held in other parts of the Horizon estate, supporting the diagnosis of software errors. This acts as a secure kernel and redundant store of data (SEK and RDS)

g) Data-driven programming (DDS), where specific functionality is achieved by generic software modules, driven by reference data: such generic modules are simpler to code and easier to test, and the reference data is easier to manage and is less error-prone, than software code. Errors in the generic code would have such widespread effects as to be rapidly detected and corrected.

h) Software coding standards, to ensure consistency of work by different developers and to discourage coding techniques which are more error-prone. These are in effect a form of architectural robustness (ARC).

240 From reading the extensive documentation of Old Horizon and Horizon Online, it is my opinion that these techniques have been widely and consistently applied across the whole Horizon IT estate. To catalogue exhaustively the many ways in which all the techniques have been used across Horizon would be a lengthy exercise.

241 I shall discuss each of the topics (a) - (h) in the following sub-sections, illustrating where it is applied in Horizon rather than listing all its applications; then I shall later apply the topics to the analysis of specific KELs or bugs.

6.2.1 Double Entry Accounting (DEA)

242 Wherever double entry accounting is used, it implies that every financial transaction is split, at an early stage of its journey through the IT systems, into separate monetary amounts (accounting postings) whose sum should be zero. In the simplest case, there are two postings of amounts +X and -X; but there may

POL00029046

equally be three postings X, Y and Z which sum to zero[40]. The different postings then take different routes through the system - whether they are similar parallel routes into the same tables of a database, or widely divergent routes to different databases.

243 This means that any software error affecting the finances (i.e. an error which would have an effect on branch accounts) is likely to have different effects on the different postings. Typically, a software error will affect one type of posting, but not another. This means the sum of the postings in a basket will no longer be zero, and the sum of all postings will be changed. Errors which do not affect the zero sum (e.g. doubling all postings in a zero-sum set) may occur but are in practice rare. Some errors can cause the same zero-sum set of postings to be made twice or not at all - which will not destroy the trial balance - but this type of error is usually detected by the measures for detection of user errors.

244 Accounting systems[41] such as POL FS sometimes require that account postings are put to them in zero-sum sets (this is an example of defensive programming, described below); but more important, in all cases they from time to time perform 'trial balances' to ensure that all postings that have been put to them, in whatever time order, have had a zero sum since the last trial balance. A wide class of software errors, which might affect branch accounts, would destroy the trial balance. In my experience of accounting systems, a failure to balance the accounts is always treated as a serious condition, so any software bug which led to it would have to be diagnosed and corrected very rapidly. Most such bugs are found in testing and never make their way into live use.

245 The double entry or zero-sum constraint was applied widely in Old Horizon and Horizon Online. In the Horizon Online counter software and the Old Horizon counter software, any customer basket, made up of any mix of products, was required to be zero sum, and this check was made at several places in the counter software. It was also made in the Horizon Online Branch Access Layer before entry into the BRDB[42]; and finally, postings from the BRDB into POL FS had to be zero sum (were not allowed to destroy the trial balance). Similar constraints applied to many types of non-customer operation, such as replenishment of stock or monthly balancing; although, as I shall describe in section 8.6, they did not apply to all such operations.

246 In summary, the DEA countermeasure was a core element of Horizon, and I have seen evidence that it was applied in many places.

6.2.2 Transactional Integrity and Recovery (TIN)

247 Because transactional integrity is a fundamental facility built into all database management software[43], and it is necessary, for any relational database, to describe in its schema the integrity constraints which it must

---

[40] Slide set 'Horizon Core Audit Process' 30/01/2014

[41] Core Concepts of Accounting Information Systems, Mark G. Simkin , Carolyn S. Norman, Jacob M. Rose, Wiley 2015

[42] ARC/APP/ARC/0008

[43] An Introduction to Database Systems 8th Edition by Date, C.J. , Pearson (2003)

POL00029046

obey at all times, Transactional integrity was applied to all of the many databases of financial information in the Horizon system[44] - including the BRDB, the POL FS database, and many others.

248 This means that any compound package of updates, applied to any of these databases, would have been applied as a single transaction or 'success unit' which would either completely succeed, or completely failed leaving no trace. It would be impossible to leave any of these databases in an inconsistent state, not satisfying its integrity constraints.

249 Transactional integrity gives protection against a wide variety of conditions:

- Hardware errors or communication failures;
- Software errors which led to failures and cancellations;
- Users deciding to cancel some operation when it is half-way through.

250 The use of transactional integrity at the database level makes it much easier to write software which will recover correctly from a wide range of conditions such as these.

251 TIN is core element of Horizon, and I have seen evidence of its pervasive presence in all Horizon subsystems that I have examined.

6.2.3 Measures to Correct User Errors, which also Cancel the Effects of Software Errors (DUE, UEC)

252 In section 6.1, I described how the requirements on Horizon for detection of user errors (DUE). such as mis-keying, are particularly stringent because of the huge volume of transactions per day, and the small margin of profit on those transactions. Therefore Horizon incorporated accepted industry practices for detection of user errors, and in my opinion did so effectively. My opinion on this is given in section 7.6.

253 In section 6.1 I have also described how many classes of user error are detected and then corrected by stock counting or reconciliation processes, so that there is no permanent error introduced in branch accounts by those user errors.

254 The same design features - introduced in Horizon for correcting user errors - also cancel the effects of a large class of possible software errors.

255 Consider for instance a software error whose effect was to lose a whole basket of customer transactions, while making it appear to the counter user that the basket had been fully processed. The effect of this software error would be identical to the user error of carrying out a physical transaction with a customer - such as selling stamps - and neglecting to enter it into Horizon at all. The previous two checks - double entry accounting and transactional integrity - would not catch this software error. However, since its effects are identical with those of a user error, the measures which ultimately cancel out the effects of the user error, would also cancel out the effects of the software error.

256 In this case, the regular check of stock against physical stock would reveal two discrepancies - one of stamps, and one of cash. In order to achieve balance and roll over, the subpostmaster would have to make

[44] TD/ARC/001, ARC/SOL/ARC/0001

POL00029046

good both discrepancies, which he can do at no cost to himself. The final effect is that the accounts on Horizon are accurate - and the software error has not adversely affected the subpostmaster.

257 Similarly, a software error which resulted in a basket of postings being stored twice would resemble the user error of entering the same basket twice - and its effects would later be cancelled by the same mechanism.

258 In my opinion, the countermeasure of UEC was so essential in Horizon, and it was effectively implemented. Because of this, many software errors resembling user errors were also corrected.

6.2.4 Defensive Programming (DEP)

259 It is a universal modern software engineering practice[45] to write programs defensively. This means to design a program as a set of small software modules, making the interfaces between the modules as simple as possible, with each module expecting the inputs it receives from other 'sending' modules to obey certain constraints - such as the double entry bookkeeping constraint of zero-sum baskets. The sending module is built so that its outputs should always obey those constraints, and it is tested to ensure that its outputs obey those constraints.

260 However, in a 'belt and braces' approach, the receiving module should not trust the sending module - but should where possible check that its inputs obey the constraints it is expecting to be obeyed and should automatically raise an alarm if they do not. The receiving module is tested with sets of invalid inputs, to ensure that it really does raise the alarm. Then, raising the alarm is treated as a serious condition, which requires immediate diagnosis of the error and re-testing of the source module - which is usually done before any live use. If this is done, then any failures in the design, coding or testing of the sending module are detected by the receiving module.

261 This technique has become especially powerful with the use of object-oriented programming, which encourages and supports design in terms of small software modules (objects) with well-defined interfaces between them.

262 The result is that as the scale and complexity of IT applications has grown, the number of serious bugs which survive testing or persist in live use does not grow linearly with the number of lines of code - because although the number of bugs initially written in to the code may grow with the number of lines of code, the number of checks which detect and expose those bugs also grows, and may even grow faster; so the number of serious bugs which survive beyond integration testing does not grow. This good practice has been essential as the complexity of IT systems has grown in recent years.

263 I have seen evidence that defensive programming was built into Old Horizon from the first[46], and was continued in Horizon Online[47] and specifically in the Horizon Online counter architecture[48]

---

[45] Design patterns : elements of reusable object-oriented software, by Erich Gamma, Richard Helm, Ralph Johnson, Addison Wesley, 1994

[46] TD/ARC/001

[47] ARC/SOL/ARC/0001

POL00029046

6.2.5 Redundant Storage of Data (RDS, MID)

264 The data generated at the Horizon counter must flow not only to the BRDB and to the central accounting system, POL FS. It must also flow to other databases and data warehouses[49] used to generate other reports. Furthermore, as IT systems are usually, for technical reasons, more complex than a layman would expect from understanding their requirements, there is a larger number of different stores for the same data - in many different slices and representations - than one would at first expect from PO's business needs.

265 Because there are many redundant copies of the same data, it becomes possible to carry out automatic checks that these redundant copies of data are consistent with one another. In a simple example, one database may hold daily summaries of some financial or stock information, while another database holds weekly summaries. These two summaries may have reached those databases by different routes through the organisation and its IT estate. However, there can be a simple and powerful arithmetic check that the weekly summaries are consistent with the daily summaries. It is common good practice to build the IT systems to make these checks wherever possible, and to raise an alarm if any check fails. Failure of a check may arise from some software error, or from a user error, or from an IT operational error such as failing to run some daily batch process. Whatever the cause, it needs to be diagnosed quickly - because until it is fixed, some of the reports from those data will not be useful to managers - being known to be unreliable.

266 There is a further check on the correctness of the data, in that different managers all look at the reports they receive and have many discussions about the content of those reports. If any error in the data leads to regular inconsistencies between the different managers' views of the business, those inconsistencies are usually soon revealed and must be corrected.

267 The architecture of Horizon contained many redundant copies of the same data[50], and implemented the RDS countermeasure. It further implemented RDS by the many ways of recording exceptions and logging of events[51]. I have also seen evidence of widespread and comprehensive reporting features in Horizon, in support of the countermeasure MID.

6.2.6 The Audit System (SEK, RDS)

268 As has been described in section 4.4 above, the audit sub-system of Horizon holds a reliable and tamper-proof record of all accounting transactions initiated at the counter[52]. So, in the case of any discrepancy between two or more of the many other databases and systems which comprise Horizon, the conflicting versions can each be compared with the audit system record, which can serve as a reliable record of what was entered at the counter.

---

[48] ARC/APP/ARC/0009

[49] The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition by Ralph Kimball,, Wiley 2013

[50] TD/ARC/001

[51] DES/APP/HLD/0035 Exceptions And Logging Frameworks High Level Design

[52] Slideset 'Horizon Core Audit Process' 30/01/2014

POL00029046

269 This means that any software error occurring in any of the back end systems in Horizon, which impacts branch accounts, will lead to a discrepancy between that system and the audit system. For a software error, the same kind of discrepancy will probably occur on many occasions, and on each occasion can be investigated by a comparison with audit data. The presence of reliable audit data makes it easier to isolate and correct software errors in any other Horizon system, by comparing data from those systems with the audit data - particularly those errors which might affect branch accounts.

270 Comparison with the audit system may also help in detecting data errors which may have arisen in other ways, such as:

- an error in running one of the many daily batch processes;
- a user error by some member of PO's back office staff; or
- any tampering with branch accounting data.

271 However, the evidence that I have seen in KELs indicates that use of the audit database was a backstop, and rarely used - because other comparisons of data were usually sufficient to diagnose the problem.

272 The audit system gives protection against tampering with branch account data, because audit records are signed with a private key generated when the counter clerk signed on using his password.

6.2.7 Data-Driven Software (DDS)

273 Another common modern software engineering practice[53] which has been applied in Horizon is data-driven programming. The data in question is often referred to in Horizon as reference data (although there are also other kinds of reference data).

274 For instance, as described in section 4.2, data-driven programming has been referred to in the old Horizon desktop software as 'soft-centred' applications. As a second example, section 5.3 describes the Services layer of Horizon Online which contains a general 'process engine', driven by reference data in the form of PDL (Process Definition Language), which provides a straightforward way of supporting different business processes at the counter.

275 The effect of this data-driven approach that, instead of having to write specific software to handle different use cases, the developer writes generic software which is driven by different reference data for each use case. This improves the reliability of software in three ways:

- The generic software is often simpler that the specific software which would have to be written to support different use cases; therefore, it is less prone to errors.
- The generic software is tested by applying it to all the different use cases; therefore, the generic software is more thoroughly tested, and less likely to contain undetected errors.
- The reference data, which must be supplied for each use case, is simpler and easier to read and understand than the code which would otherwise be written, and is often the subject of static

[53] Design patterns : elements of reusable object-oriented software, by Erich Gamma, Richard Helm, Ralph Johnson, Addison Wesley, 1994

POL00029046

validation checks, which are easily made. Therefore, any errors in the reference data are easily detected and corrected.

276 The data driven approach to software design was widely built into the Horizon architecture, for instance in the BRDB high level design[54], and the counter software high level design[55].

277 This has had an important consequence in the stability of Horizon over the years of its service. In my experience, many IT systems are afflicted by major changes in requirements, happening every few years, and requiring major rewrites. For Horizon, the core business of the PO has been stable over the period 1999-2018, so there has only been one major reimplementation (of Horizon Online in 2009) - and even that did not affect many back end systems. So compared to many large IT estates, Horizon has been very stable.

278 That stability has been underpinned by the use of data-driven software. Over those years there have been many changes of detail in the requirements for Horizon - particularly in the products offered. It has only been possible to keep the underlying software of Horizon stable by absorbing those changes in detailed requirements in the reference data used by the data-driven software - and reflecting changes in requirements by changes in the reference data, without having to alter the underlying data-driven software.

6.2.8 Software Coding Standards (ARC)

279 The use of software coding standards is mentioned in section 6.6 below on development and testing of Horizon, as a part of the software Architecture[56] countermeasure (ARC). It is addressed here because of its impact on the likely level of serious errors in Horizon.

280 If programmers are allowed to develop software, each in their own preferred style, then each one may use different programming techniques, with the result that they may have difficulty understanding the code that other people have written. It is a practical necessity for programmers to understand, extend and modify each other's work - so lack of mutual understanding of code would be harmful. Therefore, most organisations apply software coding standards to ensure consistency and mutual understanding of code.

281 These coding standards develop over time in such a way as to discourage programming styles which are more error-prone. They may include recommendations such as defensive programming, and other forms of checking - which will reduce the level of undetected errors.

282 Fujitsu' certification to CMMI level 3 demonstrates that they applied software coding standards. Much of the documentation I have seen attests to their application of standards in other respects, such as architecture, design, documentation, and testing. Informally, compared with many other large IT estates I have seen, Horizon appears to be a tightly-run ship.

---

[54] DES/APP/HLD/0020

[55] ARC/SOL/ARC/0001

[56] IT architectures and Middleware, Chris Britton, Peter Bye, Addison Wesley

POL00029046

6.3 Reconciliation, Transaction Corrections and Acknowledgements

283 Whenever the PO acts as an agent for some external client organisation, financial transactions take place for which both the PO and the client organisation have a record. The PO's record of the transaction starts at some branch counter on Horizon. The client's record of the transaction may be available to the client immediately - as in the case of a cash withdrawal from a bank - or it may only be possible to link the client's record with the PO record after some delay. This happens, for instance, where customers pay bills at a PO branch. Initially, the client organisation issues a bill to a customer - so knows the amount of the bill. The customer then pays the bill at a PO branch - after which the client organisation may check the amount paid.

284 There is subsequently a process of comparing a client organisation's record of events taking place at branches, with the PO's own records. This process is called reconciliation[57], and it is done on a transaction by transaction basis[58]. There are many differences of detail in how reconciliation is carried out for different client organisations, or where it is carried out; sometimes the client organisation does it from a file sent it by the PO, and sometimes the PO does it.

285 However it is done, and wherever it is done, the result of reconciliation is always in principle the same. For the vast majority of transactions carried out (approximately 6 million per day), the client's record of a transaction and the PO's record match, and there is nothing more to be done. However, for a small minority of transactions (which is typically a few thousand per day) there is some mismatch, which needs to be investigated and corrected.

286 When reconciliation finds a transaction for which the PO record and the client record do not match, it is passed to a department in PO accounts which handles reconciliation discrepancies. Each such discrepancy is, until it has been dealt with, an error in the accounts - and so it must be dealt with. The task of this department is to determine how each discrepancy arose, which therefore determines how it needs to be handled.

287 When the appropriate department in PO decides that responsibility for a discrepancy lies with the branch, a request for a transaction correction is issued and is passed from POL FS to the BRDB in Horizon Online. At this point, there is no impact on branch accounts. The request is passed on from the BRDB to the branch Horizon system, so that it will show on the subpostmaster's screen when he starts the Horizon system the next morning. At this point, the subpostmaster may either accept the correction or may question it and ask for further investigation.

288 It is only when the subpostmaster has accepted a transaction correction that it enters his branch accounts, and therefore enters the audit sub-system in a record sealed with his own password.

---

[57] TD/ARC/014

[58] *Reconciliation and Incident Management Joint Working Document*, 18 March 2013 [POL-0032909]

POL00029046

289 Transaction corrections may arise for reasons other than reconciliation discrepancies. For instance, when a subpostmaster recognises that his accounts need correction (for instance, after mistakenly remming in the same amount of cash twice), he may request a transaction correction.

290 Transaction Acknowledgements[59] (TAs) occur more frequently than Transaction Corrections, because they occur through normal branch business, in the absence of any errors.

291 For instance, when a customer pays a bill at a Paystation, the Paystation terminal transmits amounts to Ingenico[60], who inform the PO each day what the transaction totals were for each branch. This results in a TA being sent to the branch. When the TA is accepted by the subpostmaster, it enters the branch accounts to balance the cash from the Paystation.

292 Similar processes apply to Camelot, because lottery terminals are not directly connected to Horizon, and operate outside Post Office business hours.

293 Thus, Transaction Acknowledgements are used to balance branch accounts for cash received at a branch which is not automatically entered into Horizon at the time it is received - because the cash amount is recorded on a separate device not connected to Horizon.

294 There are cases where the TA route through Ingenico does not work, because of connectivity issues. In those cases, the subpostmaster can find out the cash amount on a printed summary from the Paystation and request a manual TC to balance the cash with the branch accounts. Without either a TA or a TC, the branch would have a cash surplus.

295 The processes of reconciliation, TAs and TCs are a very important part of the robustness countermeasures built into Horizon - particularly for UEC.

6.4 Hardware and Software Resilience (RHW)

296 The ability of an IT system to protect users from any type of disruption and to maintain acceptable service levels[61] is known as 'resilience'.

297 Resilience is required in all the major components of Horizon in branches, data centres and networks:

- Hardware
- System software, such as the DBMS and communications products
- Horizon infrastructure and applications software
- Networks

298 'A single point of failure (SPOF) is a risk posed by a flaw in the design, implementation or configuration of a system in which one fault or malfunction causes an entire system to stop operating.' [62] One strategy

---

[59] ARC/APP/ARC/0008

[60] Ingenico Group describe themselves as 'the global leader in seamless payment, providing smart, trusted and secure solutions to empower commerce across all channels, in-store, online and mobile' (https://ingenico.co.uk).

[61] High Availability and Disaster Recovery: Concepts, Design, Implementation Hardcover by Klaus Schmidt, Springer Verlag 2006

POL00029046

for minimising the impact of a failure is to replicate major components of the system. Error processing and recovery procedures also improve resilience.

299 Horizon had many features for robustness against hardware failures built into it - or instance, in the counter architecture[63], the branch database[64] and the back-end systems[65]

300 Several specific qualities of the system, which can be viewed as aspects of resilience, are addressed in other sections of this report:

- Recovery from failure is discussed under Intrinsic Error Prevention - section 6.2 above;
- Transactional Integrity – section 6.2.2;
- Security – section 6.5.

6.5 Security and User Authentication

301 In my opinion, the countermeasure of data security (SEC) which addresses a range of threats to the integrity of the system[66], has been implemented thoroughly in Horizon - as is described, for instance, in the Horizon Online architecture[67].

302 Horizon data is secured using the standard security principle of 'separation of duties'[68]. Separation of duties ensures that an individual cannot complete a critical task by themselves. For example: someone who submits a request for reimbursement should not also be able to authorise payment. An applications programmer should not also be the server or database administrator - these roles and responsibilities must be separated from one another.

303 This principle has been implemented by ensuring:

- Development units cannot have update access to any of the system data.
- Database administration functions are carried out by IS (Infrastructure Services) staff.
- Data repair is carried out by Software Support Centre (SSC) staff.

304 Data protection legislation requires that access to personal data remains within the European Union and PCI (Payment Card Industry) data security standards mandate physical security restrictions must be applied where update access is allowed to user data. Only SSC and the IS Unix team fulfil these requirements for data access. The responsibility for data correction is rests with the SSC although IS

---

[62] https://searchdatacenter.techtarget.com/definition/Single-point-of-failure-SPOF

[63] ARC/APP/ARC/0009

[64] DES/APP/HLD/0020

[65] TD/ARC/001, ARC/SOL/ARC/0001

[66] Security and Usability: Designing Secure Systems that People Can Use, Lorrie Faith Cranor, Simson Garfinkel , O'Reilly 2005

[67] ARC/SOL/ARC/0001

[68] See SVM/SDM/PRO/0875, section 1.5.3

POL00029046

sometimes act under SSC authorisation. In my opinion, this SEC countermeasure helps to reduce errors and prevent fraud and enhances the overall robustness of Horizon.

305 Access to Horizon services and system components is restricted to those who are properly authorised to use those specific services and components. Authentication seeks to verify the identity of a person (or system component) seeking to gain access to a system resource.

306 Three important pillars of security are as follows:

- Confidentiality - unauthorised observation or inference of information (e.g. about benefits paid to claimants);
- Integrity - unauthorised manipulation of information (e.g. of the data passed to and from PO and its clients);
- Availability - unauthorised denial of service.

307 A fourth pillar is audit and accountability: ensuring that users are held accountable for their actions by recording information about these actions. Threats are reduced if users understand that they will be held accountable for their actions. Audit is discussed in Appendix B.12 and in sections 4.4 and 6.2.6.

308 As described above, the Horizon architecture includes measures to address confidentiality, integrity, availability and audit. Some of the data is supplied by the government and is classified as Restricted. Because of this, security measures must follow the guidance of CESG[69]. Other data is associated with financial transactions and so the regulations of the financial services industry are applied.

309 A detailed risk assessment was undertaken for Old Horizon when the system was being designed in the late 1990s, which resulted in the following security policies being adopted:

- Physical and logical access to the system is controlled, with access granted selectively and permitted only where there is a specific need. Access is restricted to people with appropriate authorisation.
- The identity claimed by a user is verified before any access is granted to the system. Authentication mechanisms also ensure that trust relationships are established between components within, and external to, Horizon.
- All users are individually accountable for their actions. Owners are assigned for all information assets. The owners are responsible for defining who is authorised to access the information. Responsibilities may be delegated, but accountability remains with the designated owner of the asset.
- Audit mechanisms monitor and detect events that might threaten the security of Horizon itself or any service to which it is connected. These mechanisms also ensure that transactions and other events are reliably and securely recorded as described earlier in this report.
- Security personnel are alerted to violations that could seriously threaten the services.

310 The main security risks that could impact branch accounts may be summarised as follows:

[69] The UK government's National Technical Authority for Information Assurance, now part of the National Cyber Security Centre

POL00029046

- Unauthorised access – processing transactions in branch via a user account or remotely without appropriate permission; this risk includes direct access to or manipulation of branch accounts without permission. The risk is minimised by rigorous control of user identities and their access to resources.
- Theft or damage to Horizon equipment, after which the correct position is not reinstated. Resilience to this risk depends on robust procedures for recovering from failure or loss of system components and any other dislocations of the service.

6.6 Development and Testing of Horizon

6.6.1 Organisation and Governance

311 The development, testing, and support of Horizon have required large team efforts, at times involving more than three hundred full-time equivalent staff. In my experience, even teams much smaller than this need to be well structured and managed if they are to work effectively and produce good quality results. I have therefore examined the organisation and governance of the Fujitsu effort, and the results are contained in the Appendix C.

312 In my opinion, the Fujitsu Horizon organisation is well designed and is capable of working effectively. The quality and content of the documentation I have seen produced by this organisation is consistent with it having worked effectively and as intended.

6.6.2 Quality

313 In my opinion, a large system such as Horizon can only be robust if sufficient attention has been paid to issues of quality in its development, testing and support.

314 Fujitsu established a set of quality policies and processes for the Post Office Account, which were known as the Business Management System (BMS)[70]. This was intended to ensure that the company met all the requirements agreed with Post Office. The BMS and its supporting documentation were consistent not only with Fujitsu's Corporate Policies and Processes, but also with national and international standards (BS EN ISO 9001:2000, the TickIT Guide v5.0 and the CMMI for Software and Systems Engineering v1.1)[71].

315 In the year 2000, a plan[72] was published to manage quality assurance for the Horizon programme. The plan was updated annually.

316 Bill Membery is currently Head of Quality and Compliance for Fujitsu's POA (Post Office Account). His witness statement describes several different audit processes, designed to ensure by independent external review that quality is managed effectively in Horizon. If these audits have taken place as described in Mr

---

[70] PA/POL/002

[71] ISO is the International Standards Organisation, and ISO 9001 is a Quality Management System. The TickIT guide provides guidance on applying ISO 9001 in the IT industry. CMMI stands for Capability Maturity Model Integration, which is the leading measure of software engineering process maturity.

[72] QU/PLA/005

POL00029046

Membery's witness statement, and if the results are broadly as he describes them, then that would increase my confidence that the quality of Horizon has been effectively managed.

317 One of these audits involved the CMMI (Capability Maturity Model Integration), which is the leading objective measure of software engineering process maturity internationally. It can be used to assess an organization against a scale of five process maturity levels:

- Level 1 – Initial
- Level 2 – Repeatable
- Level 3 – Defined
- Level 4 – Managed
- Level 5 – Optimising

318 Fujitsu undertook a corporate initiative to obtain CMMI Maturity Level 3. As part of this initiative, POA achieved that status in December 2005. CMMI measurement was continued as HNG-X moved forward with the aim that the programme would be formally appraised during 2007 to Maturity Level 3.

319 Very few organisations are certified to CMMI levels 4 and 5. In my opinion, Fujitsu's certification at level 3 indicates that their standards were at least better than average among those organisations which seek certification - which in turn are better than the average of all organisations.

6.6.3 Lifecycle

320 To develop a robust large piece of software, it is necessary to apply an effective and recognised software development lifecycle. I have examined the lifecycle applied by Fujitsu, and in my opinion it was effective. Details are given in the appendix. Documentation I have seen indicates that Fujitsu applied this lifecycle to Horizon.

6.6.4 Testing

321 To assess the robustness of Horizon, it is in my opinion necessary to assess how effectively Fujitsu tested the software. There are many facets to software testing[73], as described in Appendix C. The evidence I have seen on Fujitsu's testing processes indicates that it was well managed and effective, and so that the robustness countermeasure of Testing Good Practice (TGP) has been applied. Details are contained in Appendix C, but I mention some important points here.

322 Fujitsu carries out a risk assessment of planned changes to Horizon to determine the extent and depth of the testing required.

323 The term 'regression testing' means re-running tests automatically to ensure that previously developed and tested software still performs correctly after a change. If not, the software would have regressed. I have seen evidence that Fujitsu used these automated testing techniques.

---

[73] Black Box Testing: Techniques for Functional Testing of Software and Systems, Boris Beizer , Wiley 1995

POL00029046

324 Because the tests can be re-run automatically, this makes it easy to ensure that new software has not introduced new bugs - that it still successfully does what the previous software did. It is therefore an important aspect of TGP, to ensure that software does not regress, and that aspects of robustness are not lost with new releases.

325 Changes that may require regression testing include bug fixes, enhancements and configuration changes (reference data). The risk assessment is used to determine which tests to repeat.

326 Non-functional aspects (NFR), such as resilience and recovery, and security, are also tested.

327 Stephen Parker (Fujitsu's SSC Manager) addresses the quality of their regression testing at paragraph 45 of his witness statement: '*I am aware of only one or two cases where a fix regressed in my time at Fujitsu.*'. If this is correct, it would reinforce my opinion that Fujitsu's testing was effective.

328 In this section, I have introduced a series of robustness measures that have been built into Horizon. Most of those rely upon functional and non-functional requirements, which are tested whenever changes are made. Therefore, the testing process also assures the quality of the countermeasure implementation.

329 Fujitsu's testing team is independent of development, support and maintenance and services – with a separate reporting line into senior management. Their job is to find bugs in Horizon and they are managed accordingly.

330 However effective is a testing organisation and its staff and tools, and however thoroughly testing is carried out, it is never possible to test every conceivable scenario which may arise, or to test all possible paths and sets of data that the software may encounter. There is a combinatorial explosion of possible paths and data, which mean that testing can never be 100% effective. That is why there are always some bugs in commercial software - because it has not been possible to test all possible scenarios which will arise in live use of the system. Some KELs describe very rare combinations of circumstances, or rare sequences of user actions, which had not been tested. In my experience this does not reflect badly on Fujitsu but is the reality of developing commercial software.

6.7 Horizon in Service

331 In my experience, there is no such thing as a commercial IT system which goes into service and works automatically from that day on. A considerable amount of work is required to keep it operating and to support it. This includes all the manual (i.e. not fully automated) countermeasures to ensure robustness. In my experience, no commercial IT system can be without these. So I have examined how Fujitsu have supported Horizon in live service. Details are in Appendix C. The points most directly relevant to the Horizon issues are summarised here.

332 ITIL is the basis for the Horizon service model. Originating from the UK Government in the 1980s, ITIL (formerly an acronym for Information Technology Infrastructure Library) is a set of detailed practices for IT service management that focuses on aligning IT services with the needs of business. ITIL underpins ISO 20000, the International Service Management Standard for IT, although there are some differences between that standard and the ITIL framework. In Appendix C, I describe selected aspects of the ITIL framework.

POL00029046

6.7.1 Systems Management

333 The scale and complexity of the PO branch estate requires proactive and comprehensive systems management. Every branch and individual counter position is under management and is being supported in successfully performing business transactions.

334 The same applies to applications running in the data centres. Any disruption can impact large parts of the branch estate.

335 Systems management facilities are needed to maximise Horizon availability and to ensure that the system delivers the service levels agreed with PO. These facilities also reduce service delivery costs as follows:

- Reducing the need for human intervention, e.g. visits to branches to install software
- Automating routine management activities, thus reducing the need for human operators
- Enabling problems to be anticipated and avoided, or their impact reduced
- Managing hardware upgrades as cost effectively as possible
- Ensuring the auditing of security events, such as authentication failures and unauthorised attempts to access resources.

336 Some of the key management products used by Horizon are as follows:

- Tivoli, supplied by IBM, was used for all services on NT platforms. It provided a central event management service, software distribution and resource monitoring facilities.
- HP OpenView together with Cisco Works are used to manage network products such as routers.
- BMC Patrol is used to handle the host central servers and the Oracle applications running on them. Patrol is specifically tailored to the management of Unix systems and applications. A Patrol Tivoli Event Adapter is provided to map Patrol events onto Tivoli events. BMC can generate pager alerts if problems arise on the platforms it manages
- Maestro, also from IBM, is used to provide scheduling facilities for batch operations.

337 Many of these products are mentioned in KELs, and many KELs refer to system management issues rather than user or software issues. The evidence I have seen indicates that Fujitsu used mature and appropriate systems management techniques - as they would need to do, to manage the large and complex Horizon estate.

6.7.2 Horizon support services

338 From time to time most subpostmasters and their staff have questions, problems or requests related to their usage of Horizon. PO and Fujitsu provide a set of services and tools to support users.

339 Horizon support was designed and is still maintained through a structured set of documentation, which defines policies, organisation, process and detailed procedures.

POL00029046

340 Most organisations providing technical support adopt a tiered model. PO and Fujitsu do this.[74]

> *'The support strategy expects that incidents will be raised by users and then passed through the chain of support units until a resolution can be supplied to the user. It is important that an incident starts at 1st line and then follows each stage of the chain as appropriate. This ensures:*
>
> 1. *The incident is quickly defined and logged*
> 2. *An initial response is given*
> 3. *Priority is correctly evaluated*
> 4. *The correct skills are applied such that a resolution is supplied quickly*
> 5. *The call is correctly recorded, auditable and relevant metrics can be produced.*
>
> *As incidents move from left to right across the support chain they become:*
>
> - *More difficult to resolve*
> - *More time consuming to resolve*
> - *The training level and cost of the staff resolving the incident rises*
> - *Tooling and supporting infrastructure costs rise*
>
> *Support costs and timescales for resolution increase as the incident moves to the right. Hence the effort spent "moving support to the left". Ensuring that the incident is resolved as early in the chain as possible reduces the cost and increases customer satisfaction (assuming a first time fix is achieved).'*

341 Mr Parker says in his witness statement (paragraph 25):

> *'Having said that, there is often overlap of skills between adjacent lines of support and while a team may be responsible for a particular level of support, staff within that team can have skills which allow them to perform a role that is more usually performed by the next level of support.'*

342 This is what I would expect from my experience of IT support organisations.

343 Horizon is supported by a four-level model, which is essentially a triage process:

1st line	• **The Horizon Service Desk (HSD)** is the branches' first point of contact for technical issues **relating to the Horizon** software **or the hardware provided in branch**. It has been operated by Atos from Manila in the Philippines since June 2014 but was previously run by Fujitsu. - deals with straightforward queries such as password issues and scheduling hardware engineers; - monitors the live estate: a System Management Centre (SMC) is run by a part of

[74] See End to End Application Support Strategy (SVM/SDM/PRO/0875)

POL00029046

	Fujitsu based in India. This team monitors Horizon system operations, taking corrective actions defined in the Horizon knowledge base whenever possible; - refers other issues to 2nd line support. • PO operates a helpdesk for operational business issues called the National Business Support Centre (NBSC). SPMs requiring assistance to determine the cause of a discrepancy contact NBSC in the first instance.
2nd line	Provided by senior members of the HSD and SMC and junior members of the SSC (Software Support Centre) - who also provide 3rd line. Note that the SSC is shared across Fujitsu customers. 2nd line support mainly involves searching knowledge articles based on the descriptions of issues reported by branches, gathering evidence and applying simple, well-defined work-arounds (often on the phone).
3rd line	Provided by SSC staff with a detailed knowledge of the Horizon application based on documentation and some inspection of source code. SSC use a defect management system called Peak (PinICL until 2003) to log and manage incidents passed to them which were suspected to be faults. Peak is also used to manage faults identified by testing. The SSC maintains the SSCWeb application, which includes the Known Error Log (KEL). This enables searching of system operations events and provides access to help text as well as other support and technical information. The KEL describes the symptoms of problems with some analysis of causes, potential solutions to the problems and workarounds (WOR) that might be needed before a permanent solution can be implemented.
4th line	Members of this team have specialised knowledge of specific areas of the system and are responsible for the producing permanent fixes to repair the root causes of incidents or other problems in the live application. They amend source code to fix problems. There is clearly an overlap between 4th line and the development team, which adds new features into the application.

344 Therefore, 1st and 2nd line support do not identify software bugs.

345 Over the lifetime of Horizon, the organisation of support services has evolved.

346 Because it is relevant to the creation of KELs and Peaks, I have examined the Fujitsu incident management process, and the results of that examination are documented in Appendix C. In Appendix G, I provide evidence of how Horizon incident management works in practice by examining sample Peaks with their corresponding KELs.

POL00029046

347 While I cannot comment on how effective the first line of support was in directly helping SPMs, in my opinion the second, third and fourth lines were effective in recognising common causes of different incidents, in diagnosing the causes, in discovering which branches were affected, and in finding fixes and temporary workarounds. In my opinion the evidence I have seen, particularly from KELs and their associated Peaks, is indicative of a high-quality team who understood Horizon in great depth and applied that knowledge effectively. This evidence indicates to me that their incentive was to fix problems, so that the problems would not recur, adding to their costs and workload.

348 In my opinion, these teams carried out the manual countermeasures of MID, WOR and BFC very effectively. They were able to do this because of the many automatic countermeasures such as TIN, RDS, DEA, ARC, DEP, and DDS (which on many other occasions, meant that their services were not necessary in any case); and by collaborating with SPMs who were applying MID.

Known Error Log (KEL)

349 The KEL is primarily a knowledge base, rather than a 'bug list'. The log currently comprises more than 8,000 entries.

350 KELs may be raised by testers to flag minor issues that are not resolved immediately. In exceptional circumstances, the development group has also raised KELs to inform the support teams of potential problems coming their way. The KEL is used mainly for supporting operational users, rather than by Fujitsu's internal teams.

351 There is no defined vocabulary, which means that searching for particular terms may be unreliable. The lack of defined terms may reduce quality when new incidents and problems are recorded. This is counter-balanced using both process and procedure documentation (including standards) and categorisation wherever practical. KELs are searched and read by a wide community of support users, so their quality is actively maintained via mandatory reviews both by a manager and by a forum (KEL Approval).

6.7.3 Reference Data Management

352 Processes and systems for reference data management are described in Appendix C. My opinions of the significance of reference data management for the Horizon issues are contained elsewhere in this report - particularly in section 7 below, when assessing the countermeasure of Data-Driven software (DDS), and in section 8 (extent of bugs caused by faulty reference data)

6.7.4 Change Management

353 Effective management of changes in Horizon is part of the robustness countermeasure Quality and Change Control[75] (QCC). I describe Fujitsu's processes for change management in Appendix C and give my options on the QCC countermeasure in section 7.

[75] http://prince2.wiki/Change_Control_Approach_Template

POL00029046

7. HORIZON ISSUES – ROBUSTNESS OF HORIZON

7.1 Issues Addressed in this Section

354 This section of my report addresses the Horizon issues in our Group 1. These are Horizon Issues 3, 4, and 6, which concern robustness of Horizon. These issues are:

355 **Issue 3**: To what extent and in what respects is the Horizon System "robust" and extremely unlikely to be the cause of shortfalls in branches?

356 **Issue 4**: To what extent has there been potential for errors in data recorded within Horizon to arise in (a) data entry, (b) transfer or (c) processing of data in Horizon?

357 **Issue 6**: To what extent did measures and/or controls that existed in Horizon prevent, detect, identify, report or reduce to an extremely low level the risk of the following:

a) data entry errors;

a) data packet or system level errors (including data processing, effecting, and recording the same);

b) a failure to detect, correct and remedy software coding errors or bugs;

c) errors in the transmission, replication and storage of transaction record data; and

d) the data stored in the central data centre not being an accurate record of transactions entered on branch terminals?

7.2 Robustness of Horizon: My Opinion

358 I here summarise my opinion on Horizon issue 3.

b) My opinion from the evidence is that at all times for which there are KELs - which is nearly all the lifetime of the system - Horizon has been a very robust system, compared to other major systems I have worked on in sectors such as banking, retail, telecomms, government, and healthcare.

c) I have described 18 types of robustness countermeasure, which I have applied routinely on projects over many years. In my opinion, Fujitsu have applied these countermeasures effectively in building and supporting Horizon.

d) As an accounting system, Horizon particularly needs countermeasures to ensure the accuracy of the accounts, in the face of many types of adverse event. I have focused particularly on these countermeasures and their effectiveness. In my opinion, these countermeasures are well designed, and have been effective in preventing errors in accounts. Very few adverse events - including user errors and software bugs - have evaded all the countermeasures to the extent of causing significant inaccuracies in branch accounts. Horizon is very unlikely to cause significant shortfalls in branches.

359 This summary of my opinions applies also to the Horizon issues 4 and 6.

360 The experts have agreed that robustness is not a matter of perfection, or the complete absence of bugs. Horizon, in common with all large commercial IT systems, was not completely free of bugs.

361 Robustness involves the use of a set of techniques, which I call countermeasures, to ensure that many kinds of potentially harmful events (including hardware failures, communications failures, user errors and

POL00029046

software bugs) do not have harmful consequences - or if they do, the harmful consequences are kept within acceptable limits.

362 Here, the definition of 'acceptable' may involve using standard IT techniques of Risk Assessment[76] which I have applied retrospectively to assess the risk of bugs in Horizon introducing errors in branch accounts. The results are reported in section 8.

363 Robustness is a core requirement for any major commercial IT system and has been so for many years. Large parts of IT project budgets are spent ensuring that systems are robust. Robustness countermeasures are described in sections 4-6 of my report.

364 The techniques for achieving robustness are so important that they have become a well-established and central part of commercial IT practice. I have listed 18 major techniques, or types of countermeasure, which I have routinely applied on major projects over thirty years. These techniques act in concert to minimise harmful effects.

365 To make Horizon robust, Fujitsu had to effectively apply these established techniques. I have examined the evidence of how Fujitsu designed and built Horizon, how they tested it (in section 6.6), and how they supported it (in section 6.7). The 8390 KELs, in particular, are a rich source of evidence about Horizon in service - about events which threatened to have harmful consequences, and how well or badly the robustness countermeasures acted in those cases. My analysis of many KELs implies to me that the countermeasures in Horizon worked well in the live use of Horizon.

366 In some market sectors where I have worked - such as banking, telecoms, and healthcare - robustness is often compromised by the presence of very old legacy software (sometimes coming from merged organisations) which is hard to maintain or adapt - and has resulted in an over-complex frozen 'spaghetti' architectures. Horizon does not suffer from these problems. Horizon was a 'green fields' development started in 1996 - essentially unencumbered by any IT legacy. Therefore, it was much easier to build a robust architecture from the start.

367 In his report, Mr. Coyne has not described any robustness countermeasures, or assessed how well they were applied in Horizon. In my opinion, robustness of Horizon is the central expert issue of this case, because it is robustness - rather than the number of bugs - which determines the financial impact of bugs on claimants' accounts.

368 I first address Horizon issue 3: To what extent and in what respects is the Horizon System "robust" and extremely unlikely to be the cause of shortfalls in branches?

369 Issue 3 overlaps with issue 1, because 'unlikely to be the cause of shortfalls in branches' overlaps with the issue of bugs, errors or defects which 'cause apparent or alleged discrepancies or shortfalls' (Issue 1).

[76] see http://prince2.wiki/Risk

POL00029046

370 For clarity of presentation and description, I shall postpone the discussion of the 'unlikely to be the cause of shortfalls' aspect of issue 3 until the next section of the report, which is about Horizon issue 1. This is for two reasons:

- As I shall describe in section 8, the extent of shortfalls caused by Horizon depends on the robustness of Horizon, and on a range of well-known robustness countermeasures as implemented in Horizon. So, describing those countermeasures, and how successfully they have been implemented, as will be done in this section, is an essential pre-requisite for addressing Horizon Issue 1 in section 8. Mixing the two issues together would lead to a confusing presentation.
- Robustness involves preventing or minimising many other harmful effects, as well as shortfalls.

371 In their outline of 17 August 2018, the claimants say that they do not understand any 'objective meaning' of the term 'robust', and imply that it may be an IT marketing term, used just for public relations.

372 In my opinion, this is not so.

373 In large IT projects, robustness (and the almost synonymous term, resilience) are very important design objectives. Large parts of project budgets are devoted to achieving them.

374 The term 'robust' receives its meaning from the phrase 'robust against ... [some risk or threat]' and there are many threats which business IT systems need to be robust against. Horizon needs to be robust against hardware failures, communications failures, power cuts, user errors, disasters, fraud, and hacker attacks - to name just some of the high-level threats, which can then be further subdivided. For instance, communication failures can be subdivided into a simple failure to communicate, and the communication of erroneous information. Robustness of IT systems is a large and mature topic.

375 In software engineering there is a well-established practice and terminology for discussing robustness under the heading of 'risk management', within which the extent of robustness, as in Horizon issue 3, can be addressed. The extent of the robustness of Horizon consists of:

- The list of risks and threats that Horizon needs to be robust against
- For each risk, some measure of the probability of it occurring (which may for instance be measured on a scale of red/amber/green, in typical risk management methodologies)
- For each risk, the seriousness of the consequences both before it is not handled, and after risk management measures have been put in place (also possibly on a scale or red/amber/green)

376 These risk management methodologies are a part of the PRINCE2 project management methodology[77] applied by Fujitsu to Horizon - as was described in section 1 above.

377 My emphasis in this section will not be to discuss the dimensions of threat or risk that robustness is designed to counter, but to discuss the many different countermeasures which have been developed in the IT industry to counter these risks, and how well or otherwise Fujitsu have applied these countermeasures.

[77] See http://prince2.wiki/risk

POL00029046

378 In practice there is no simple relationship between risks and countermeasures. Typically, the effects of any type of risk are to be mitigated by several different types of countermeasure, acting together in different ways.

379 I introduced the various types of robustness countermeasure in a table in the summary of my opinions at section 2 of this report and defined a three-letter acronym for each class of countermeasure. Typical of these acronyms are RDS (Redundant Data Storage) and UEC (User Error Correction).

380 In sections 4-6 of this report, I gave further details about how the different countermeasures have been built into the architecture of Horizon, and how they have been tested.

381 It is important to understand that robustness does not mean to be 'perfect' - as that is impossible. It means: 'manage the risks of imperfection so they are acceptable'.

382 The question then arises: "What counts as 'acceptable'?". In many IT projects, the answer is supplied by standard IT techniques of risk analysis - which consist of applying simple probability theory to estimate the probabilities of certain unwanted events taking place. Typically, some target probability is adopted for commercial, safety or other reasons, and the project then needs to apply robustness countermeasures and testing and mathematical analysis, to demonstrate to stakeholders that the target probability will not be exceeded.

383 For instance, when building software to control the dosage delivered by an X-Ray machine, it may be required to ensure that the probability of an incorrect dose does not exceed one part in ten million. (This example and these figures are purely illustrative).

384 For a commercial system like Horizon, one such target probability might, for instance, be 'the probability of a software bug causing a significant shortfall in any branch accounts in any month'. I have not seen evidence of any numerical risk analysis of probabilities like this being carried out by PO or Fujitsu. This does not surprise me, because in my experience the numerical analysis of such probabilities is more commonly done in the context of safety-critical systems, for instance in transport, manufacturing and medicine. However, I have been able to analyse one such probability (the probability of bugs creating errors in a set of monthly branch accounts) retrospectively, from Horizon's track record in service, and the results are reported in section 8.

385 Countermeasures typically work by there being many different countermeasures acting together - so that even if one countermeasure is not fully implemented, and does not catch and counter some risk, other countermeasures will act as extra lines of defence - so that very few threats get past all the lines of defence.

386 Therefore, as was agreed in the joint expert statement of 04 September 2018, robustness does not mean perfection. It means recognising that imperfections exist, and adopting countermeasures (typically, many of them) to minimise the likely harmful impact of those imperfections. Here, imperfections can include external factors such as user errors or power failures, or internal factors such as errors in software. They may include errors in the countermeasures themselves.

POL00029046

387 A familiar example may illustrate this. Many people create and edit documents using Microsoft Word. This product is not perfect. It may freeze up for no good reason, or it may allow its users to work for hours without saving their work - and so be vulnerable to hardware failures or power cuts. But over the years, it has become more robust, so that the costs of these failures are now usually acceptable. Whatever goes wrong, there is usually a recent 'auto-saved' version of your document, so you do not lose too much of your work. That is practical robustness, not perfection.

388 The robustness of an IT system can be understood through a biological analogy, to the immune system. Just as there are many risks and threats to an IT system, so there are many risks and threats to the human body. The body cannot be designed as if there were no such things as bacteria, viruses, injuries, poisons, genetic defects and so on; so the immune system is part of a multi-layered defence system against these threats. The robustness of the immune system depends on its being multi-layered; if a first layer of defence, such as white blood cells, does not deal with the threat, then an increasingly complex set of defences, including T cells, B cells and so on, is mobilised until the threat is dealt with. Running a fever is not a sign of perfection; but it is a sign that the body is defending itself robustly.

389 Similarly, it is easy to point to evidence such as KELs which imply that Horizon was not perfect; but perfection in an IT system is no more possible than a lifetime of perfect health. Often the same KELs describe how the threat was addressed in Horizon, through a multi-layered defence system which includes robustness of the platform software, robustness of the IT architecture, error correction measures such as reconciliation, and investigation of remaining anomalies through the four levels of IT support. Robustness depends on these many layers. A failure of Horizon's robustness would mean the failure of many layers of defence.

7.3 Countermeasures to Achieve Robustness

390 This section contains a brief survey of all the different types of countermeasure which are commonly used to achieve robustness. Because these countermeasures have been described earlier in the report, the description takes the form of a table, with references to the sections of the report which describe the countermeasure. This table is duplicated in the summary of opinions, as section 2..

391 For each countermeasure, I have defined a three-letter acronym for easy reference through the report.

No.	Countermeasure	Explanation and examples	Described in Section
1	Reliable and redundant hardware (RHW)	Redundancy guards against many types of hardware failure. Examples: RAID discs, disaster recovery sites. Software is designed in many ways to be robust against hardware failures	4.2, 5.4, 6.4
2	Robust data communication and replication (ROC)	Communication systems and protocols are designed to recover from and protect against many kinds of communication failure. Examples: TCP/IP, Riposte	4.2, 5.3, 6.4
3	Transactional Integrity and database recovery (TIN)	Database management systems provide many facilities so that numerous kinds of failure cannot leave the data in an inconsistent, unusable state, or lose any data that have been previously stored	4.2, 4.3, 5.4, 6.2, 6.4

POL00029046

4	Defensive programming (DEP)	Software is divided into small self-contained modules, which do not assume that other modules are correct, but defend themselves by checking their inputs and raising alerts early	5.3, 6.2, 6.4
5	Generic, data driven software (DDS)	Different use cases for software often have much in common. Software is written generically to be able to handle the different cases, using reference data to define which use case is to be handled. Example: variations in PO client products handled by reference data.	4.2, 4.3, 5.3, 6.2
6	Secure kernel hardware and software (SEK)	When a large complex IT system is subject to threats, the design may include a small, well tested and secure kernel which is proof against those threats. Examples: secure kernels of operating systems, Horizon core audit process	4.3, 4.4, 5.4, 6.2
7	Redundant data storage and computing, with cross-checks (RDS)	In large IT systems and sets of systems, data are stored redundantly in several places, and routine operations check automatically that the different copies of the data remain consistent.	4.3, 4.4, 5.4, 6.2, 6.4
8	Double entry accounting (DEA)	Accounting systems operate by the principles of double entry book keeping, so that any change to the accounts must be made in a transaction whose summed effect on all accounts is zero. Transactions which do not obey this constraint are rejected.	4.2, 5.3, 5.4, 6.1, 6.2
9	Early detection of user errors (DUE)	At the point of user input, as many checks as possible are made of the correctness of the input - so that the system will not accept erroneous input and may warn the user of errors.	5.3, 6.1
10	Later correction of user errors (UEC)	In accounting systems, the system's version of reality is periodically checked against external versions of reality and corrected if wrong. Examples: cash balancing and rollover, reconciliation and TCs.	4.2, 4.3, 6.1, 6.2, 6.3
11	Manual workarounds (WOR)	Whenever any part of Horizon does not work as required, there may be potential to define and apply manual workarounds.	6.7
12	Testing good practice (TGP)	The purpose of system testing is not to prove that the system is correct, but to prove that it is incorrect in any way possible. Examples: regression testing, user testing, testing edge cases.	6.6
13	Manual Inspection of data (MID)	Any large business IT system is used by many people, who view its outputs and check them against each other for consistency, and against their own knowledge of the business. SPMs, watching their branch accounts, were a key component of this.	4.3, 6.2
14	Bug Finding and Correction (BFC)	Whenever the system shows any anomalous behaviour, that is investigated, its causes found and corrected. Interim workarounds are deployed. Extra checks may be added to ensure that other similar threats are handled correctly.	6.7
15	Large scale IT architecture (ARC)	In any large IT estate, principles of IT architecture are used to achieve robustness - such as using a distributed network of loosely coupled sub-systems with clearly distinguished functions. The sub-systems are built to well-defined standards with clear interfaces.	4.1, 5.3, 5.4, 6.2, 6.4

POL00029046

16	Quality and change Control (QCC)	Systems are more robust if quality is inherent. This is achieved by organising properly the people who build, maintain and operate the system, by managing them well and by governing what they do through rigorous but effective processes. A system will only continue to be robust if changes are controlled in a way that enhances quality without unnecessary administration.	6.6.2, 6.7.3
17	Managing non-functional requirements (NFR)	Robustness is improved by paying close attention to non-functional requirements and the associated 'ilities' such as manageability, supportability, maintainability and adaptability.	-
18	Security (SEC)	Any system that could be easily subverted would not be robust. Horizon is secured mainly through 'separation of duties', user authentication, access control and audit.	6.5

392 These different types of robustness countermeasure have been described and introduced in sections 4-6 of this report, and I will not repeat the descriptions here. References to the relevant sections for each type of countermeasure are given in the table above.

7.4 My Experience of Robustness Countermeasures

393 As I have described above, the robustness countermeasures are so essential to large commercial IT systems, that they have become a routine part of large IT projects. I have been applying them on such projects since around 1979. I note here some of the countermeasures which are particularly familiar to me.

394 TIN: I designed and developed one of the first relational database management systems (DBMS), and managed its commercial exploitation by my company Logica for several years. During that time, transactional integrity and recovery (TIN) became an essential requirement for our clients. I personally designed the transactional integrity facilities of our DBMS, and was involved in configuring the product for many commercial clients. Since then, TIN has been a routine foundation of most of the software I have been involved with.

395 DEA, RDS, MID, and UEC: I have been involved with computerised accounting systems in several roles both as an architect (for instance, in deployments of SAP) and as a user, when I held line management positions in Logica. In those roles I spent much of my time scrutinising management accounting data (MID), checking it against other data sources (RDS), and ensuring it was an accurate reflection of reality (UEC). All of these were underpinned by the principles of double entry accounting (DEA)

396 RHW, ROC: Particularly when working in the finance sector (investment banking and retail banking), the projects I have worked on have depended on redundant and reliable hardware (RHW) and have had to be robust against many kinds of communication failure (ROC).

397 ARC, NFR, QCC: I have acted many times in major projects where large scale IT architectures (ARC) and non-functional requirements (NFR) have figured centrally in the customer's requirements. Often these projects have involved multiple releases of software, requiring mature techniques for change control (QCC).

POL00029046

398 WOR: In all IT projects, there has to be some consideration of manual 'last lines of defence' (WOR). IT disputes often involve cases where these defences were tested and failed. I have been involved in investigating a major retail banking failure of this nature, and another such failure in government computing.

399 DEP, DDS, DUE, TGP. BFC: I continue to be involved in implementation projects in healthcare IT, developing and testing software in Java and other languages. In these projects, defensive programming techniques (DEP), data-driven software (DDS) and robust, error-proof user interfaces (DUE) are essential - as are thorough testing (TGP) and bug-fixing (BFC) to ensure patient safety.

400 I have not had much personal involvement in building secure kernel software (SEK), or computer security techniques (SEC) - although I am familiar with the underlying mathematical specification methods.

7.5 Effect of Countermeasures on Bugs Which Might Affect Branch Accounts

401 In this section I discuss how various countermeasures acted specifically on any bugs in Horizon which might have had some effect on branch accounts. The discussion of this section is partly based on the KELs which I have sampled. One purpose of this discussion is to state and justify some assumptions I shall make in section 8 when addressing Horizon Issue 1, on the extent of the impact of bugs.

402 In surveys I have made of KELs, I have observed:

i. Many KELs are not about bugs in Horizon. In those cases, Horizon is acting as intended, and the KEL exists to give appropriate advice to an SPM in specific circumstances.

ii. Of the KELs which are related to faults in Horizon, many of those faults self-evidently have no effect on branch accounts; for instance, they may just be about some inconvenience for an SPM, or about some back-end reporting issue.

iii. Of the remaining KELs, which describe bugs with some potential for impact on branch accounts, in many cases it can be easily inferred that some countermeasure would prevent any actual impact on branch accounts. The main countermeasures involved, and how they prevent financial impact, are described below. When some countermeasure would avoid financial impact, sometimes this is stated explicitly in the KEL, but often it is not. I assume this is because KELs were written by people deeply familiar with Horizon, who expected their readers to understand that there would be no impact (as noted at paragraph 66 of Mr. Parker's witness statement).

403 The countermeasures which most frequently prevent any long-term financial impact of bugs in class (iii) are:

- TIN: transactional integrity means that some customer transaction either succeeds in its entirety, or in the case of some error, has no effect on the branch database - as if the transaction had never been started.
- UEC: of any software bug has the same effect as a user error (for instance, as an error in recovering a recoverable transaction, or in cash management) then the normal measures for correction of user

POL00029046

errors (mainly, Transaction corrections, and monthly balancing) will correct the error, just as they correct the many user errors which occur.

- RDS/MID: Many errors produce anomalies which are visible to the SPM (by his inspection of system behaviour, MID) - either immediately, or later in monthly balancing. Once he has reported it, Fujitsu have many ways to look at system logs and other redundantly stored data (RDS), to establish what happened and ensure there is no adverse impact on branch accounts.
- WOR: Once they were alerted to any condition which caused difficulty for the SPM, the various levels of support appear to have been good at finding workarounds for either the SPM or the back office to apply, to ensure there were no harmful effects.

404 When all these KELs are removed from consideration, very few remain - where there is a possible bug in Horizon, and it is not obvious (for one of the reasons above) that it will have no impact on branch accounts. I have only found a very small number of these KELs. Mr Coyne has examined over 5000 KELs, and for none of them has he presented the analysis which would be needed to show that a KEL is in this group - for instance, to show that it is not in groups (ii) or (iii), for instance because of countermeasures.

405 One can classify bugs in Horizon, with possible impact on branch accounts, into the following groups:

a) Bugs whose effect is immediately evident to the clerk in the branch - for instance, because it prevents him from doing something he needs to do to serve a customer.

e) Bugs whose effect is only visible to the SPM when he does monthly balancing and rollover - the bug causing a discrepancy which he can notice and investigate.

f) Bugs which are never visible, either to the SPM or centrally, but which nevertheless affect his branch accounts.

406 In my opinion, there are no bugs of class (c). I shall explain why this is so. Obviously, one cannot deduce it from examining KELs, because if a bug were totally invisible, it would never cause calls to a help desk, so might never trigger creation of a KEL.

407 It is a central principle of Horizon that the Core Audit Database acts as a secure 'gold standard' for branch accounts (countermeasure SEK) and that the audit record can only contain events which originated at the counter - either in customer transactions or monthly balancing. Therefore, it is impossible for any bug to influence the branch accounts without showing itself at the branch, either in customer transactions (class(a)), or in transaction corrections, or in monthly balancing (class(c)), so it is put into the audit database.

408 The same conclusion may also be implied by different considerations.

409 Suppose there was some Horizon bug, not visible to the SPM at any time, which caused money to leak out of his branch accounts. In my opinion, because PO accounts are kept in POLSAP, which uses double entry accounting, any money which leaks out of a branch account must show up in some other account.

POL00029046

Setting aside the possibility that losses in one branch show up as gains for another, this other account must be some PO central account, which aggregates the financial impact across all or many branches.

410 Because many branches are aggregated in this central account, in my opinion even small amounts in single branches would add up to large amounts in this central account - amounts up to 11,000 times larger (summing the amounts across all branches). I believe there are PO managers whose role it is to monitor these accounts and understand them. I have not seen evidence about how these managers operate; but I nevertheless question whether such large amounts of money arising in some central PO account would not be noticed. If they were noticed, this would be a combination of the DEA and MID countermeasures.

411 There are therefore only classes (a) and (b) of bugs with possible financial impact.

412 In class (a) - bugs whose effects are immediately evident to the clerk - there are many KELs which record events which puzzled the clerk, even if their financial impact was small or zero. Therefore, I assume that any such bug would be reported in some non-zero proportion of its occurrences, however small its financial impact. Even if the SPM does not report it immediately, he or his counter clerk notices it at the time; so if there is a later anomaly in his monthly balancing, he may relate the two and then call the help desk.

413 For class (b) - bugs whose impact is only evident at monthly balancing - there is unfortunately a paucity of evidence in KELs about how often they are reported. The purpose of KELs is not to record the branches affected by a bug, or the amounts involved; nor is that the purpose of any related PEAK. Such information is expected to be recorded elsewhere.

There is relevant evidence in one case where we have a fuller analysis - the local suspense account bug. Here, an analysis by Gareth Jenkins,[78] states that 16 branches were affected, with discrepancies of the following amounts:

Branch	Name	Amount
002647	Aberystwyth	-£6.71
002840	Inverness	£140.61
010007	East Dulwich	-£0.01
011458	Willen Village	-£9,799.88
012004	Lower Edmonton	£16.12
054011	Lower Regent Street	£3.34
101832	Dundas	£5.84
104937	Grange	£0.03
104937	Grange	-£49.65
155025	Hounslow	-£113.14
156715	Gilford	£11.55
211844	Rosyth Terminus	£36.20
211844	Rosyth Terminus	-£77.97
243242	Wardles Lane	-£0.51
266418	Bowness Road	£3,186.70
297611	Merthyr Dyfan	£160.92

Table 7.1 - Discrepancies arising from the local suspense account bug

[78] 'Local Suspense Problem', LocalSuspensev05.docx by Gareth Jenkins

POL00029046

414 Only the two branches with the largest discrepancy reported it. This allows me to conclude tentatively that:

- if a branch sees a discrepancy in monthly balancing of £3000 or more, the SPM is very likely to report it
- If the discrepancy is between £100 and £200 (three cases, none reported) , the likelihood of the SPM reporting it is probably less about than 20% (because 0.8^3 = 0.8 cubed = 0.512; if the probability of reporting is 20%, the probability of not reporting on any one occasion is 0.8)
- If the discrepancy is less than £50 (10 cases, none reported) the likelihood of the SPM reporting it is less than about 5%

415 These kinds of estimates of probabilities, made from limited data, are typical of the kind of estimates often required in IT risk analysis (for instance as used in the PRINCE2 project management methods)

416 I can go a little further than this by making weak inferences about how a manager of a small business, such as an SPM, needs to prioritise his time in monthly balancing, and other evidence. The witness statement of Angela Van Den Bogerd at paragraph 187 says: 'Generally, when discrepancies are of a value of several hundreds of pounds, I would expect Subpostmasters to contact NBSC.' If this is correct, it is consistent with my estimates. I assume the following, as best assumptions of SPM behaviour in reporting anomalies in their monthly balancing:

a) If a discrepancy is £1000 or more, the SPM probably needs to investigate it. If he cannot find the cause in his branch, the likelihood of his reporting it through a help line is 80%

b) If a discrepancy is of the order of £300, 30% of SPMs will report it

g) If a discrepancy is of the order of £100, 10% of SPMs will report it

h) For a discrepancy of £10 or less, it is usually not worth the SPM's time to investigate it (because errors in counting cash or stock are often larger than this); so these are reported on less than 1% of occasions.

417 In this respect, I also note that a bug which causes some mean discrepancy (for illustration, £300) causes a range of different discrepancies on different occasions. Therefore, it may sometimes cause discrepancies much larger than £300, which are more likely to be reported.

418 There is an important exception to these expectations. If an SPM knows that there is already some large discrepancy in his accounts (for instance, if he had a larger discrepancy in his previous month's accounts, and has no reason to expect it to have gone away) then in my opinion he is less likely to notice or report smaller discrepancies. The last category (d), which do not reveal themselves immediately to the clerk, and whose financial impact at month-end is of the order of £10 or less, I shall refer to as 'micro bugs'. As was described above, even these micro-bugs, through Double Entry Accounting, are likely to produce larger aggregated effects, which might be noticed in some central PO accounts. I shall discuss micro-bugs further in section 8 and Appendix F relating to it.

POL00029046

419 Apart from the micro bugs, the analysis of this sub-section shows that the SPMs themselves acted as an important countermeasure against bugs impacting their accounts, through their own inspections of Horizon and its data (MID).

420 I emphasise that these are tentative assumptions, based on the evidence above , and on my understanding of how the manager of a small business might prioritise his time. They will enter in my analysis of the financial impact of bugs, in section 8; but before they are used in any calculation, I shall move them in a conservative direction, to make the result more reliable and more favourable to the claimants.

421 Once any anomaly is reported to a help desk, PO and Fujitsu had processes to triage it, and to create a KEL if there was any need to advise branches how to handle a problem, or to take other action in the back office.

422 From my analysis of Fujitsu's support structures and processes in section 6.6, I have found these processes to be fairly efficient. Mr. Parker's witness statement at paragraph 62.8 states that if any anomaly was thought to have the potential to influence branch accounts, it was treated as high priority.

423 Whether or not this is correct, it is my experience that any IT system support team has an incentive to detect commonalities between different calls from users (whatever their cause, whether it be user errors or bugs in the software) so that they can be handled efficiently and effectively - for instance, by creating KELs which describe how to handle them, and maintaining those KELs. My assumption is therefore that these processes resulted in a KEL on more than 90% of the occasions where it was reported and there might be some effect on branch accounts. Once again, I shall move this assumption in a conservative direction, to favour the claimants, before using it in any calculation. If there is any reason to alter it, I shall do so and redo the calculations.

424 Mr Parker's witness statement states at paragraph 61.9 that about 15% of KELs have been archived, and are not in the set of 8390 KELs supplied to myself and Mr. Coyne. They can be recovered with some effort. I have asked to examine some of these archived KELs, to ascertain if there is anything special about them; but I have not yet received them. Meanwhile, assuming Mr Parker's statement is correct, I shall account for them statistically in any numerical calculations I make, as KELs I am unable to examine, just like the other KELs I have not examined for lack of time. If Mr Parker's statement is not correct, I shall need to revise my calculation (by changing one number in the spreadsheet).

425 Taken together, these points mean that the KELs (and their associated Peaks) form a reliable source of evidence about Fujitsu's performance on bug-finding and correction. The factors I have noted above (SPM reporting behaviour, and Fujitsu's creation of KELs) mean that I can sample the KELs and their Peaks to see how well Fujitsu diagnosed and fixed bugs (this includes all bugs, including those that affected branch accounts).

426 My sampling of KELs implies:

- Across all KELs, Fujitsu were generally able rapidly to identify the cause and any fix required

POL00029046

- A small proportion of reported anomalies could not be reproduced in testing, and remained perplexing. In my experience, this is to be expected in any complex system.
- Once they had identified the cause, Fujitsu were generally able to identify all occurrences of any bug in system logs - and often to suggest workarounds while it was being fixed
- The speed with which the fix was made and put into live use depended on its priority (except for reference data fixes, which were generally made very quickly)
- On a small proportion of occasions, fixing one problem caused another, which was observed later (the suspense account bug was one of these - it was a side-effect of a previous fix). Mr Parker's witness statement says that he is aware of only one or two such cases. I note that there may have been other cases of regression of fixes that he was not aware of.

7.6 Assessing How Well Countermeasures Were Applied

7.6.1 Methods of Assessment

427 I have assessed how well each type of countermeasure was applied, in several different ways:

- I have assessed in section 4-6 to what extent Fujitsu implemented the countermeasure, as evidenced by the design documents for Horizon, and documents describing the processes they intended to apply.
- I have assessed in section 6.6 how well some of the countermeasures were tested in Fujitsu's testing processes.
- By examining KELs and Peaks, I have assessed to what extent countermeasures acted in live use of Horizon

428 The Knowledge Error Log (KELs) are over 8,390 records of issues where help desk support was required, and where centrally collated knowledge would be of assistance. In section 6.6, I have described the processes by which KELs were created, maintained, and used in support of the branches.

429 KELs were used to handle a variety of issues, including user errors, software errors, and reference data errors. In my opinion, the KELs and their associated Peaks are the best remaining record of what happened when one of these issues occurred - including the robustness countermeasures which were triggered and helped to mitigate its effects. Therefore, inspecting the KELs is a good way to assess how effectively, or otherwise, the robustness countermeasures did their job in live use. However, the KELs are not a perfect record, because, for instance, they assume a lot of knowledge of Horizon.

430 As I have described above, robustness countermeasures are designed to ensure that several kinds of harmful effect do not ensue - including errors in branch accounts. In assessing the effectiveness of countermeasures, I have focused particularly on this aspect of their use - how well they prevented any harmful effects on branch accounts.

POL00029046

431 I have now examined of the order of 200 out of the 8390 KELs, in various different samples. From this analysis, a picture of the robustness countermeasures in action has emerged. My analyses are in tables in Appendix D. However, I recognise a number of limitations of the analysis:

- The KELs are written by people who were deeply familiar with the architecture and details of Horizon, for other people who were equally familiar. Therefore, they often assume knowledge of Horizon which I do not have. Allusions and terms which were obvious to the KEL reader are no longer obvious to the experts, and it would take a disproportionate effort to disinter them.
- Horizon is a highly complex system, resulting from many thousands of man years of work. KELs refer to many complex parts of Horizon, and it has not been possible in the time I have to follow through all the threads from KELs, understanding in depth all the components they refer to. I have had to rely on the understanding described in sections 4-6 of this report.
- For some KELs - particularly those mentioned only recently in Mr. Coyne's report - there has not been time to analyse them in the depth I would like. I may be able to provide deeper analysis in my supplemental report.

432 So there are some KELs for which I cannot be certain exactly which countermeasures were effective, whether or not they had any impact on branch accounts, or how much impact they had. This does not show any limitation in Fujitsu's analysis of the problem at the time; it only shows the limitations of the analysis I am able to do now.

433 Appendix D contains four tables of my analyses of different sets of KELs. The first two of these tables contain analyses of how those KELs show that countermeasures were applied:

- Appendix D.2 analyses 30 KELs selected at random from the 8390 KELs (actually chosen as every 100th KEL in an alphabetically sorted list), noting some of the applications of countermeasures evidenced by each KEL.
- Appendix D.3 analyses 62 KELs cited in Mr Coyne's report, noting some of the applications of countermeasures evidenced by each KEL.

354 Each table also comments on the potential impact of the KEL on branch accounts, but that is not my purpose in referring to them in this section.

355 I note two further limitations of these analyses of countermeasures in KELs:

- They are not intended to be an exhaustive list of the countermeasures in action. I believe that by further examination of each KEL and its Peaks I could find evidence of other countermeasures in action
- KELs are a biased source of information about countermeasures - giving more information about manual countermeasures, and generally not mentioning automated countermeasures, whose effect is often to forestall the existence of any KEL in the first place. For instance, TIN frequently prevents erroneous or not completed transactions from having any effect - so that no KEL is ever required.

POL00029046

356 Nevertheless I shall briefly summarise the results from the first table of 30 KELs. In these KELs , I have found 54 instances of countermeasures being applied - which in itself shows how widely they were built into Horizon. The countermeasures which are in evidence three or more times are DEP, DDS, RDS, TGP, WOR, MID, and BFC.

7.7 Opinions on Robustness Countermeasures

7.7.1 Reliable and Redundant Hardware (RHW)

434 Sections 4-6 survey the evidence that in the data centres, robust and redundant hardware was used extensively, because the consequences of failures in the back office would be so serious.

435 The hardware in the branches was not as reliable or redundant as the back-end hardware, but in my opinion, this was an acceptable economic trade-off, because the costs of hardware failure in a branch were so much less serious. Hardware failures frequently occurred in the branches - for instance, failures of keypads. The evidence suggests that branch software was designed with robustness countermeasures such as TIN to ensure that hardware failures would rarely, if ever, have adverse effects on branch accounts. Had this not been the case, the number of errors in branch accounts induced by hardware failures would have been unacceptably high.

436 I conclude that, partly because techniques for dealing with hardware failure are so mature and have been so for the full lifetime of Horizon, Fujitsu only had to apply these established techniques, and they did so effectively. I have seen no evidence that they failed to do so.

7.7.2 Robust Communication and Replication (ROC)

437 In the early days of Horizon, the underlying data communications infrastructure to the branches was so unreliable that it was essential to provide reliable replication and communication of data in spite of it, and Riposte effectively provided this. (This was the state of the art in data communications at the time, not a limitation of Horizon) I have seen fewer than 4 KELs where Riposte data replication was suspected not to have worked correctly, in particular circumstances.

438 Even in the later Horizon Online era, failures of the underlying communication infrastructure are still so frequent that it is essential for the layers above it to be robust against such failures. Part of this robustness is provided by standard communication protocols, which Fujitsu used; part was built into the Horizon software, using measures such as DEP and TIN. So in my opinion, the ROC countermeasure is applied effectively in Horizon.

7.7.3 Transactional Integrity and Recovery (TIN)

439 Transactional integrity and recovery have been built into database management systems (DBMS) since 20 years before the start of Horizon, and DBMS are used for essentially all Horizon storage of persistent data. Therefore, transactional integrity has been a core feature of the design of all components of Horizon from the start. This is evident in many of the Horizon design documents. It is clear from those documents

POL00029046

that TIN is a core component of Horizon's robustness against many kinds of failure, such as hardware failures, communication failures, or the user abandoning or cancelling some task in the middle.

440 While it is not mentioned explicitly because it was often so obvious for the intended readers, KELs show many examples where TIN was relied upon to avoid harmful effects. In my opinion, TIN is pervasive and effective in Horizon.

7.7.4 Defensive Programming (DEP)

441 The layered architecture used in Horizon, and described in sections 4-6, provides evidence of defensive programming - where each layer would defensively protect itself against errors in the layers it depended on.

442 The KELs provide many other examples of defensive programming - where the effects of some error (in hardware, software, or user input) was rapidly trapped by some defensive programming measure - by some module testing its inputs - before it had penetrated far into the system. Defensive programming is essential to make it easy to find the origin of problems, by trapping them before they have gone far. The KELs generally show that problems were easily diagnosed - implying that they were defensively detected near their source.

443 The evidence I have seen implies that DEP (like TIN) was pervasive and effective in Horizon.

7.7.5 Data Driven Software (DDS)

444 As was described in sections 3-6, many parts of the Horizon architecture were implemented in a data-driven way - using reference data to define how some generic piece of software would run in many specific applications. This was widely applied, and had advantages for robustness:

- The generic software was more economical to write and test, than specific software for all the applications
- It was repeatedly tested in all the different applications, so was generally highly reliable
- Faults produced by faulty reference data were easy to correct, by fixing the reference data.

445 It also had a potential drawback for robustness, if the reference data was not carefully managed and had errors.

446 KELs show a significant number of faults arising from faulty reference data. These would only rarely affect branch accounts - as they often prevented the use of some transaction at the counter, rather than allow it to be done wrongly - but they may have had the potential to affect branch accounts in some cases. Reference data faults were generally easily diagnosed, and were rapidly fixed by changing the data - so did not have any impact for long.

447 As I described in section 6.2, data driven software is used widely in Horizon. In my opinion, the countermeasure DDS has been highly effective - particularly in enabling the Horizon software to remain very stable over twenty years, using reference data to accommodate changes in requirements.

POL00029046

448 The benefits of this stability, in enabling all other countermeasures to remain stable and effective, have in my opinion been considerable.

7.7.6 Secure Kernel Hardware and Software (SEK)

449 The core audit system was a important secure kernel in the Horizon system. It served as the 'gold standard' record of what transactions had been entered in the branch. All the evidence I have seen implies that it was carefully implemented and has served its purpose correctly through the life of Horizon.

450 Because the core audit system was a backstop countermeasure, which was only used when other ways of investigating any anomaly had not given an unambiguous result, it was only rarely used, and the KELs provide little evidence of its use. This comparative lack of KELs using the audit system provides confirmatory evidence that the other countermeasures were effective.

451 In my opinion the core audit system has at all times been an effective implementation of the countermeasure SEK.

7.7.7 Redundant Data storage (RDS)

452 Horizon was very large and complex system, in which the same data were stored redundantly in many different sub-systems. There are very many examples, in KELs and otherwise, where comparisons of these redundant copies of data, automated or manual, were used to detect anomalies. this makes RDS one of the most important robustness countermeasures used in Horizon.

453 Because it often required human inspection to compare the redundant copies of the same data, the countermeasure of RDS often goes hand in hand with manual inspection of data (MID). Some of the many examples of where RDS was used, with or without MID, are:

- Reconciliation is a automated comparison of two copies of the same transaction data - which is responsible for trapping a large number of errors of various kinds.
- Accounting systems store and display financial data from many different sources, in a variety of different ways and different 'slices' - sliced by time, department, product and so on. Managers spend large amounts of their time scrutinising and comparing these figures, and there are automated comparisons.
- System logs and event logs are a redundant storage of information about what happened in transactions, independent of the transaction data which results in the BRDB and other places. When diagnosing any anomaly, Fujitsu staff placed great reliance on these logs, and comparing them with transaction data (e.g. in Horice) to understand what had happened. They were usually successful in doing so.

454 The KELs provide many examples of where RDS was used, with or without MID, to understand the origins of problems.

455 Where RDS was used, and MID was also necessary, this raises a question. If data were stored redundantly somewhere, why was the comparison not made automatically? There may or may not be a good answer to

POL00029046

this question, in each individual case. This could sometimes have been an automatic countermeasure, and would have trapped an error faster and more automatically than the RDS/MID combination.

456 There are examples in KELs where in my opinion RDS could have been used in a fully automated data comparison , but was not, and required MID. This is a potential criticism of Fujitsu's robustness measures. But the tradeoffs to be made in each individual case are complex, and go beyond my knowledge of Horizon.

457 The large number of instances of RDS which I have seen in Horizon imply to me that RDS was implemented effectively in Horizon.

7.7.8 Manual Inspection of Data (MID)

458 Manual Inspection of data often goes hand in hand with RDS (where the inspection consists of a comparison) , but does not in all cases. It is possible to look at some data and to know 'this is wrong' before you compare it with some other data. Managers often do this with financial data.

459 So while many uses of MID involve RDS, as in the previous sub-section, there are many which do not.

460 One important case, shown in many KELs, is where someone in a support team scans some system logs or event logs, and sees something suspicious. A comparison may be made against some SPM's account of what he did at the time.

461 For this reason and others, in my opinion MID was one of the most important countermeasures in Horizon. MID was , of course, not an automated countermeasure, and was often needed only when automated countermeasures had not prevented a problem. In my opinion, any commercial IT will incorporate some level of MID countermeasures. It is inevitable in the nature of any commercial accounting system that managers at various levels will scrutinise the financial data held in the system, and that this continual cross-checking will be an important check of the quality of the data.

462 The large number of instances of MID which I have seen in Horizon imply to me that MID was implemented effectively in Horizon

7.7.9 Double Entry Accounting (DEA)

463 Since Horizon was an accounting system, and interfaced closely with another accounting system (POL HS or POLSAP), principles of double entry accounting were widely applied - but not universally. Therefore many potential errors were trapped by DEA.

464 There are examples where the use of DEA can be inferred from a KEL, although it is seldom explicitly stated that DEA applied. Wherever a KEL mentions POLSAP, there is usually DEA. Similarly, most changes to the BRDB were made in double entry baskets, which must balance to zero.

465 Some errors can bypass the DEA check. For instance, in the receipts/payments mismatch bug, the change made to BRDB was a double entry change, but the double entry basket was not correct. This led to a discrepancy between BRDB and POLSAP, which could be detected later (RDS/MID).

466 However, in general the DEA countermeasure was applied effectively in Horizon.

POL00029046

7.7.10 Early Detection of User Errors (DUE)

467 Validation of user inputs has been standard practice in building user interfaces since before the start of Horizon. This involves making all checks possible at the time of input - such as data format checks - or constraining the possible user inputs by a variety or means, such as menus of allowed values..

468 Fujitsu applied these standard techniques to the Horizon desktop, both in the development of Horizon and later of Horizon Online.

469 There are many cases of user error which cannot be detected by any IT system. Typical of these is the user simply entering the wrong amount of money for some transaction.

470 In between, there are cases where one could reduce the frequency of user errors by demanding more of the user. Typical of these would be requiring the user to enter the same data twice. The design of these features clearly involves a tradeoff. If some user error is comparatively infrequent, and if its effects can later be corrected (as they usually can be in Horizon - see the next sub-section), then it may not be a good choice to encumber all users with the extra work of a more error-proof interface.

471 In section 6.1, I described how the requirements for DUE are particularly stringent in Horizon, because of the large daily volume of transactions with manual input of data, and the small profit margins of PO, which require high precision in those transactions.

472 In my experience (which includes managing a group of user interface specialists), the design of user interfaces is a deceptively complex topic. Often what appear at first to be obvious improvements to a user interface turn out (on careful examination, or after evaluation in user trials) not to be so, because of complex tradeoffs and user factors which were not appreciated by the designer. It is very easy for a software engineer to believe that he knows what will be good for users, when he does not know. This applies to experts' retrospective opinions, just as to designers' prospective views.

473 In my opinion the facilities of Horizon to prevent mis-keying were as well designed, as far as I am able to assess. To go further than this might require specialist expertise and running user evaluations.

474 Taking these factors into account, in my opinion Horizon was well designed in respect of detecting user errors, and there is no sound basis for thinking it could easily have been improved.

7.7.11 Later Correction of User Errors (UEC)

475 As I have described in sections 4-6, part of Horizon is an accounting system - whose function is to hold a version of the financial state of an organisation, which is always to be kept as accurate as possible. In order to do this, it is essential to have robust processes to correct for the effects of user errors - to check the system's version of the financial state against external reality, and correct it if necessary. Horizon had these , in two main forms:

- Reconciliation and Transaction Corrections: These were applied to the transactions which PO carried out as agents for its clients, such as banks. This had the effect of retrospectively correcting many kinds of errors, including hardware issues (e.g. involving pinpads) or user failures to manually recover recoverable transactions.

POL00029046

- Daily cash balancing: this was an important measure to catch errors in handling cash as soon as possible, so they could be remedied while their possible causes were still fresh in memory.
- Monthly balancing and rollover: This had the effect of correcting many forms of error, including user errors in entering cash or stock transactions into Horizon.

476 In my opinion these formed an essential and effective countermeasure.

477 They had an additional important effect. Not only would they correct for user errors, as above; they would also correct for a wide class of other errors, including software bugs, whose effects were the same as user errors.

478 Some of the KELs acknowledge explicitly that these countermeasures would correct the effects of user errors on branch accounts after a delay. For many other KELs, one can infer that the error would have been corrected by the UEC countermeasure, even though it is not explicitly stated. In my opinion this was part of the support team's knowledge of Horizon, which was so evident that it did not need to be stated. Correction of user errors was a routine and obvious part of Horizon's functionality. It was the main purpose of monthly balancing , and of TCs; this was often not stated in a KEL.

7.7.12 Manual Workarounds (WOR)

479 KELs provide many examples of manual workarounds being suggested for particular problems. These could be simple (e.g. close down the system and restart it) or could be more complex sequences of actions. In many cases, they were not for the SPM to do, but were corrective actions to be taken on some back end system at the data centres. The descriptions of these corrective actions could be complex sequences, and the descriptions almost universally assumed familiarity with the system in question; the terminology may be unfamiliar to outsiders.

480 The KEL typically does not describe how many times the workaround was used , but may give some expectation of how many times it is expected to be necessary (as in 'if this occurs again...'), or for how long it would be needed (until some fix went live)

481 These KELs indicate to me that Fujitsu were in most cases able to understand the origin of a problem , and suggest a manual workaround, fairly rapidly. They give the impression of a support team that knew what it was doing, not thrashing around for possible solutions; there is not much 'try this and see if it works'.

482 The KELs also indicate to me that as soon as Fujitsu understood the origin of a problem (which they usually did) they were then, on many occasions, able to detect its past and future occurrences from examining system and event logs (MID), even if branches did not report it. This also indicates a support team that knew what it was doing.

483 So in my opinion, the MID countermeasure in Horizon was effective.

POL00029046

7.7.13 Testing Good Practice (TGP)

484 An important part of the robustness of Horizon, although not often directly evidenced in KELs, is good practice in testing, to ensure that not many bugs, and only bugs with infrequent occurrence, get though into live use.

485 I have examined Fujitsu's testing practices in section 6.6, and found them to be effective and professional. Particular points relevant to the robustness of Horizon are:

- The use of independent testing teams, whose incentive is to find bugs (rather than to show that the system is correct)
- The use of many test scripts to systematically test 'unhappy paths' where the user does something unexpected or incorrect
- The use of automated regression testing, to ensure that any new release passes all the tests which the previous release had passed.
- Testing of robustness and recovery situations, testing the countermeasures described in this section

486 The other evidence which, in my opinion, supports the effectiveness of Fujitsu's testing is the rarity , in the record of KELs and Peaks, of serious bugs which affected branch accounts in live use. In my opinion, TGP was effectively applied.

487 In this respect I note that Mr. Coyne has looked for such bugs, examining over 5000 KELs, and not found them. For none of the KELs he cites does he give the analysis which would be needed to show an effect on branch accounts. I have done this analysis, and in my opinion most of the KELs he cites have no effect on branch accounts. Fujitsu have analysed the same KELs, in a table attached to Mr Parker's witness statement, where they conclude that few of them had effects on branch accounts. If the court accepted this evidence from Mr Parker (which in my opinion is based on a deeper knowledge of Horizon than my own), it would further confirm that the KELs cited by Mr Coyne do not imply errors in branch accounts. It can also be inferred with some confidence (although not complete confidence) that none of the KELs examined by Mr Coyne said explicitly 'this will affect branch accounts' - or if any of them had said this, in my opinion he would have quoted some of them. KELs are typically fairly short documents, easy to scan.

7.7.14 Bug Finding and Correction (BFC)

488 By inspecting the KELs and their related Peaks, one can usually reconstruct the history of any software bug noted in a KEL and find how long it took to fix it and put the fix in live use.

489 I have not yet done this in a systematic or numeric way. My criteria for selecting KELs, and the KELs I have examined, are listed in Appendix D. From the KELs I have examined:

- For only a small minority of the KELs were Fujitsu perplexed by the nature of a problem - and those were usually the problems which they could not reproduce in test. This happens from time to time for diverse reasons (particularly when the circumstances triggering a bug are rare, i.e. it is not an

POL00029046

important bug), and in my experience is to be expected. For most problems, Fujitsu were able fairly rapidly to identify the cause - typically within days or weeks - and know what was needed to fix it.

- If a problem could be corrected in reference data, it was usually fixed very rapidly
- If a problem required code changes, the speed with which these were done depended on the perceived importance of correcting it. The very few examples (such as the three known bugs) suggest that if a bug had the potential to affect branch accounts, it was treated with high priority. Mr Parker's witness statement at paragraph 62.8 says that ' Incidents with a financial impact on branches are treated as high priority.' If this is correct, it would confirm what the examples and my own experience would suggest.

357 In my opinion, the KELs and Peaks indicate the Fujitsu's level of BFC was broadly very effective. In the few cases where they were unable to identify the cause of a problem or reproduce it in test, it would take a deeper analysis than I have had time to do, to understand the reasons for the difficulty. In my experience, there are always some difficult problems like those.

7.7.15 IT Architecture (ARC)

490 Much of the robustness of Horizon derives from its architecture. As described in sections 4-6, the architecture of the Horizon back end consists of a large number of discrete systems with well-defined functionality and interfaces, rather than some smaller number of monolithic systems.

491 This form of architecture has been good practice in large-scale IT for many year, and Fujitsu simply had to follow it. This they did, which gave many benefits in design, testing, and support. This architecture generally makes it easy for a support team to isolate the cause of a problem into one of the systems.

492 In my opinion, the quality of the Horizon architecture is generally good. This follows not only from my examination of the architecture in sections 4-6, but also from the effectiveness of the support processes which depend on it.

493 In designing the architecture of Horizon, Fujitsu had a major advantage - a fresh start in 1996, unencumbered by any legacy architecture. In many of the large systems I have worked on - particularly in the finance and healthcare sectors - new developments need to be fitted onto some unwieldy legacy architecture - making it yet more unwieldy. While the Horizon architecture is complex, it does not have the arcane over-complexity that I have seen in many commercial organisations.

494 I have also noted that because PO's major requirements have changed little in 20 years, and because detailed changes in requirements have been addressed using DDS and reference data, the Horizon architecture has been very stable over its lifetime.

7.7.16 Quality and Change Control (QCC)

495 Quality is the cornerstone of any system's development and maintenance. Section 6.6.2 introduces how Fujitsu assured the quality of Horizon. The key Quality Control (QC) techniques employed on Horizon were:

- producing documents in accordance with standards and templates;

POL00029046

- reviews of specifications, designs and other significant documents;
- testing of software, including changes.

496 I have reviewed many of the thousands of documents produced during the lifetime of Horizon. They provide clear evidence that those documents have followed the defined document management processes and standards[79] and that they have been reviewed and approved by staff with appropriate levels of responsibility.

497 The KEL and Peaks I have examined include evidence of software testing.

498 Quality is undermined by uncontrolled change. Fujitsu change management was designed to comply with the ISO/IEC 20000-1:2005[80] objectives, e.g. *'To ensure all changes are assessed, approved, implemented and reviewed in a controlled manner'.*

499 Horizon's Managed Service Change (MSC) process[81] controls changes in the live system, new features as well as fixes. I understand that more than 36,000 MSCs have been created. Although the change documents themselves have not been disclosed, the KELs and Peaks that I have examined include evidence that this process is embedded in Horizon's service operations.

500 In their 2011 audit of PO's internal controls[82], Ernst & Young (E&Y) identified exceptions in the Horizon Online change management regime whereby certain user accounts could deploy changes in Horizon without proper authorisation. Fujitsu and PO committed, in their response, to implement a series of corrective actions. Those actions confirmed they were addressing the issue, but I have not seen evidence of whether those actions were completed nor whether E&Y were able to confirm in a subsequent audit that the shortcomings had been rectified.

501 In my opinion, Fujitsu and PO have exercised the QCC countermeasures described above effectively so as to make a significant contribution to the robustness of Horizon.

7.7.17 Managing Non-Functional Requirements (NFR)

502 Non-functional requirements (NFR) such as availability, capacity, performance and scalability are essential considerations for any large system[83]. A system like Horizon cannot be robust unless these qualities are actively managed. Therefore, careful and effective management of the NFRs constitutes another robustness countermeasure - although direct evidence for management of NFRs is unlikely to be seen in working documents such as KELs (unless some NFRs are insufficient, and problems arise from that).

503 Availability encompasses resilience (RHW) because the system will fail unless its hardware, software and network are reliable. System failures reduce availability and disrupt the business supported by Horizon.

---

[79] PA/PRO/010 and PGM/PAS/PRO/0007

[80] https://www.iso.org/standard/41332.html

[81] Described in Appendix C

[82] E&Y report for year ended 27 March 2011 [POL-219218]

[83] Mastering non-functional requirements, Sameer Paradkar, Packt 2017

POL00029046

504 The system can also become unusable if it runs out of space (capacity) or runs too slowly (performance). Unless it can grow in line with demand (scalability), it will breach the required performance or storage levels.

505 Security (SEC) is also treated as an NFR.

506 The Horizon architecture documents[84] show that NFRs have been designed into the system and its management processes. KELs include indirect evidence of successful NFR management. However, I have not examined records of NFR compliance in the live system. See also paragraph 510 for information about Horizon availability.

7.7.18 Other Aspects of Robustness

507 Further evidence on the general robustness of Horizon comes from the lack of interruptions of the Horizon service. In recent years, several major banks, and air traffic control systems, have suffered well-publicised IT 'meltdowns' which led to serious interruptions of service; these systems were not robust. As far as I am aware, in more than 18 years' service, Horizon has experienced very few such interruptions - one on 9th May 2016, cited in the claimants' WS. I have not yet seen sufficient evidence to offer any further opinion.

508 This incident, and others that I have found in KELs, imply that the robustness of Horizon and the business processes around it was not always as good as it might have been - but in the great majority of cases, was good enough. In a typical incident recorded in a KEL and its Peaks, there were several countermeasures (different lines of defence); some were breached, and others were not. On some of these occasions, in my opinion some of the lines that were breached could have been built more strongly, so they would not have been breached. For instance, in some cases there might have been more automatic cross-checking of data, where there was none.

7.7.19 The Effect of Multiple Countermeasures

509 This is my most important conclusion on the robustness of Horizon.

510 I have described 18 different classes of countermeasure - some overlapping - and have given examples of how they prevented errors from affecting branch accounts.

511 These countermeasures did not need to be individually perfect, in order to be highly effective in combination.

512 To illustrate this, suppose (for illustration) that each type of countermeasure is implemented with only 90% effectiveness - so that one bug in 10 gets past it..

513 This would mean that two countermeasures, acting independently, would catch some bug with 99% effectiveness. Only one bug in a hundred would get past both countermeasures. Three countermeasures would catch it with 99.9% effectiveness, and so on.

---

[84] For example, TD/ARC/001 and ARC/SOL/ARC/0001

POL00029046

514 Of course it is not always possible to have several countermeasures at once to detect any bug - as bugs have something of the nature of 'unknown unknowns' - but it is an engineering principle to try to bring as many countermeasures as possible to bear on any possible threat. This principle can result in highly robust systems, in spite of typical rates of errors per line of code (such as one defect in 1000 lines of code) often achieved in the IT industry.

515 In my opinion, Fujitsu successfully applied this strategy of many diverse countermeasures, as is demonstrated in many of the KELs. As a result, Horizon is a highly robust system.

516 In the foregoing discussion, many of the countermeasures are fully automatic and act with no human intervention. Other countermeasures such as MID require human intervention. In my opinion, the fully automatic countermeasures in Horizon were well designed and worked well. However, it is more difficult to find evidence of how well they worked in live use of Horizon, because my main source of information for this - the KELs - is biased in the following sense: the KELs record mainly occasions where some defect evaded the automatic countermeasures, and necessitated some manual countermeasure. So KELs tend not to record the operation of fully automatic countermeasures, only the manual ones. The best evidence for automatic countermeasures in service is the comparative rarity of KELs over 19 years.

7.8 Variations in the Robustness of Horizon Over Time

517 In his report, Mr. Coyne has expressed an opinion that the robustness of Horizon may have varied over time - implying that the number of bugs which could have had impact on claimants' branch accounts may also varied over time.

518 Mr Coyne has described the variability of robustness of Horizon as a possibility, but has not cited evidence that it actually varied.

519 In this sub-section I shall discuss the two related questions raised by Mr. Coyne:

- Variability in the robustness of Horizon over time
- Variability over time in the number of bugs which may have affected branch accounts.

520 The first of these questions is problematic. Neither Mr Coyne or I have suggested any numerical measure of robustness. In my opinion, the topic is so complex (consisting of many different countermeasures, acting together in many different ways) as to admit of no single numeric measure. If you cannot measure something, there is an important sense in which you cannot ask how it varies over time. You can only ask how some consequence of robustness - such as the impact of bugs on branch accounts - varied over time. That is the second question.

521 As I have described earlier in this section, robustness (as opposed to freedom from bugs) consists of effectively applying the many countermeasures described in this section, so that the effects of imperfections, including software bugs, are limited to an acceptable level.

522 Of the 18 robustness countermeasures described in this section, in my experience all of them have been available, and have been a common part of mainstream IT practice, since before the inception of Horizon.

POL00029046

Therefore, they were available to Fujitsu from the start. I have not found any evidence that any of the countermeasures was applied more or less effectively in any period of Horizon's lifetime.

523 It seems to me rather unlikely that , if any countermeasure is applied in some release of Horizon, that it should not be applied in the next release - i.e. that it should be deliberately dropped. In my opinion there was no incentive for PO or Fujitsu to do this. It might be said that some countermeasure could be applied less effectively in a later release, for one of two reasons:

a) If it is an automated countermeasure, because the countermeasure software had some new bug

b) If it is a manual countermeasure, because the team doing it became less effective (e.g. due to a cut in manpower, or a mis-managed organisational change).

524 These are both possible, but I have not seen evidence for either of them. (a) is made more unlikely by the practice of regression testing - ensuring that a new release passes all tests applied to the previous release.

525 Furthermore, as noted in section 6.2, Horizon's requirements, design and architecture have been very stable over its lifetime. This in itself implies that the robustness countermeasures have been similarly stable.

526 Therefore I have seen no evidence that the robustness of Horizon *per se* has changed over time - although, as I have said above, the question itself is rather ill-posed..

527 I turn next to a possible consequence of changes in robustness, which is a change in the rate at which bugs affect branch accounts.

528 The claimants are put forward two hypotheses:

a) That some large part of their claimed £18.7M shortfalls arises from bugs in Horizon

a) that the rate at which this happened has varied over time

529 If both hypotheses are correct, one should be able to see the result, in a variation over time in the rate at which claimants experienced shortfalls.

530 Data on the occurrence of claimed shortfalls by year, taken from Section 3 of the individual claimants' claims, is shown below:

POL00029046

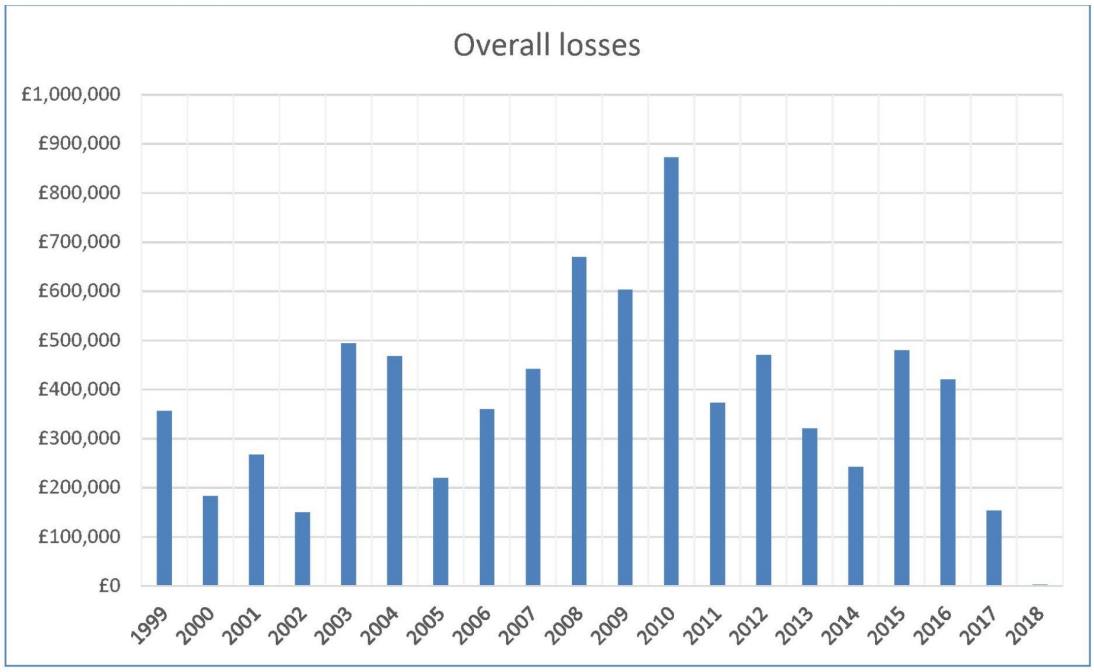


Figure 7.1 - Claimed shortfalls by year

531 It can be seen that there is some variability over the years, but that variation is in my opinion mainly consistent with random fluctuations, with no systematic trend and no huge variability.

532 The main exception to this is a noticeable peak in 2010. In Angela Van Den Bogerd's witness statement at paragraph 183, she says that there was a mandatory cash check in all branches before the change to Horizon Online, which may have caused a temporary spike in declared losses. If this is correct, it might account for the spike in 2010.

533 However, I note that in my opinion, as expressed in section 8, the claimants' shortfalls are not caused by bugs in Horizon, or any lack of robustness in Horizon. So, in my opinion, the graph above says nothing about robustness of Horizon, or about its consequences. So, I have looked for other evidence about occurrence of errors.

534 KELs are written to help address a range of issues raised by SPMs. Many of those issues arise from human errors or other causes, but some of them arise from faults in Horizon. Therefore, a chart of the number of KELs raised in each year may shed some light on the occurrence of faults in Horizon. That chart (which I have made by an automated analysis of the KELs) is shown below:

POL00029046

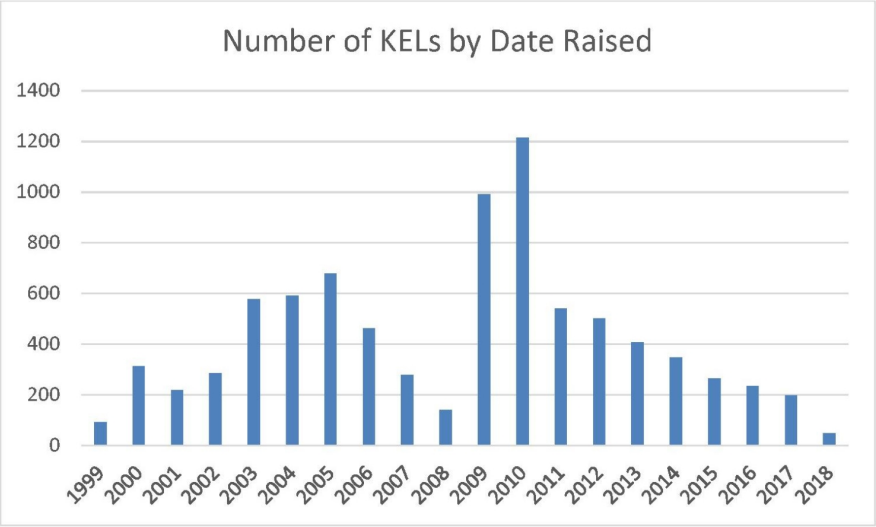


Figure 7.2 - KELs raised by year

535 It can be seen that there is a noticeable surge in the rate of creation of KELs in 2009, 2010 and the years immediately following them. Some of this surge may have arisen because the support team were having to deal with new issues under Horizon Online, because it was different from old Horizon, and were therefore creating new KELs to help them do so.

536 However, this surge post-2009 also raises the possibility that there was a higher level of bugs in Horizon Online - which were teething problems of the new Horizon Online release - and I do not exclude that possibility. How significant is the surge?

537 To answer this question, I anticipate a conclusion from section 8 of this report, about Horizon issue 1.

538 In that section I show - using several separate lines of analysis - that the maximum total impact of all bugs in Horizon, on claimants' accounts over all years, is very small, compared to their claimed £18.7 million total shortfalls. It is much less than 1% of that amount.

539 I have therefore calculated the maximum possible sum of the impact of bugs over all years and have shown that it is very small. If one knows that the sum of a quantity over all years is very small then the fluctuations in that quantity over the years are of little importance. The 'ups' are cancelled by the 'downs'. All that matters is the sum, which I can place an upper limit on. So, any fluctuations over time, in the rate at which bugs in Horizon caused shortfalls, are of little or no importance.

540 I conclude that while there may possibly have been fluctuations over the years in the rate at which bugs in Horizon could have caused shortfalls, any such fluctuations are of no importance, because their sum over all the years of Horizon's life is very small.

7.9 Horizon Issue 4

541 **Issue 4**: To what extent has there been potential for errors in data recorded within Horizon to arise in (a) data entry, (b) transfer or (c) processing of data in Horizon?

POL00029046

542 I find parts of this Horizon issue difficult to interpret. My preferred interpretation is that they are to be seen as selected subsets of Horizon Issue 3.

543 The reference in (c) to 'processing of data in Horizon' is difficult to understand, because essentially all parts of Horizon are involved in 'processing of data' . So my opinion on issue 4(c) is the same as my opinion on issue (3).

544 The reference in (b) to 'transfer of data recorded in Horizon' is difficult to understand, because many parts of Horizon are involved in 'transfer of data' - which is not necessarily restricted to 'transfer of data over communication channels'. If it is so restricted, I refer back to section 6 of this report, and to the ROC countermeasure described earlier in this section. Horizon successfully incorporated the usual measures to protect data in communication. If it is not so restricted, most of the countermeasures described in this section are applicable.

545 The reference in (a) to data entry is more specific. In my opinion, the Horizon user interface incorporated industry-standard measures to detect user errors in data entry wherever possible. Since many errors of data entry cannot be detected automatically, this was as robust as it could be. I refer to my discussion of the DUE countermeasure in this section.

546 So in my opinion all aspects of issue 4 are subsets of issue 3 on robustness, and my opinion on issue 3 applies to issue 4. In summary, this opinion is that Horizon has been a robust system at all times, and that its robustness countermeasures have worked effectively.

547 I note that issue 4 asks about the 'extent' of 'the potential for errors in data'. However, unlike Horizon issue 1, it does not restrict these to errors which impact branch accounts. Therefore, it is not possible for me to quantify this extent in any useful way, since its scope is so broad; and 'extent of potential' is intrinsically difficult to quantify. - being even broader than an extent of actuality.

7.10 Horizon Issue 6

548 **Issue 6**: To what extent did measures and/or controls that existed in Horizon prevent, detect, identify, report or reduce to an extremely low level the risk of the following:

a) data entry errors;

b) data packet or system level errors (including data processing, effecting, and recording the same);

c) a failure to detect, correct and remedy software coding errors or bugs;

d) errors in the transmission, replication and storage of transaction record data; and

e) the data stored in the central data centre not being an accurate record of transactions entered on branch terminals?

549 Like Issue 4, issue 6 raises similar issues of interpretation, and appears to consist mainly of selected subsets of Issue 3.

550 In that Issue 6 addresses 'measures ... [to] prevent, detect, identify, report or reduce ...', Issue 6 seems to be largely about robustness and countermeasures - and to that extent, is subsumed under issue 3.

POL00029046

551 Issue 6(a) seems to repeat issue 4(a), and my opinion is repeated. In summary, this opinion is that Horizon has been a robust system at all times, and that its robustness countermeasures have worked effectively.

552 In issue 6(b), I cannot usefully interpret the reference to 'system level errors' , which might refer to any aspect of the Horizon system; so all robustness countermeasures apply. 'data packet' errors might refer to many different types of data packet. If it is used a widely recognised sense, it refers to data communication, and my opinions on the ROC countermeasure apply. As for issue 4(c), the reference to 'data processing' makes the scope of issue 6(b) as wide as that of issue (3), and my opinion is repeated as in the preceding paragraph.

553 Issue 6(c) refers to software coding errors. This was one of the threats mentioned in my analysis of robustness under issue 3, and is addressed under several robustness countermeasures, notably BFC. My opinion is repeated as for issue 6(a).

554 Issue 6(d) appears to repeat issue 4(a), and my opinion is repeated as for issue 6(a).

555 Issue 6(e) draws out a specific point mentioned above under issue 3. Fujitsu took care to ensure that data stored in the core audit database was an accurate and secure record of transactions entered on branch terminals, and could be called upon as a 'gold standard' when investigating any anomaly. This is addressed under the countermeasure SEK. In this respect, Horizon was robust.

556 So all aspects of issue 6 are subsets of issue 3 on robustness, and my opinion on issue 3 applies to issue 6.

557 As for issue 4, the 'extent' aspect of issue 6 is difficult to address in any quantitative manner.

7.11 Mr Coyne's Opinions

558 In the expert's joint statement, Mr Coyne has agreed that robustness does not equate to perfection or the lack of bugs.

559 His opinions on the robustness of Horizon are stated in paragraphs 5.82 - 5.200 of his report.

560 In paragraph 5.83, he confirms the view in the joint statement that robustness does not equate to lack of bugs.

561 However, in paragraph 5.88 he says he is unable to estimate the level of robustness in Horizon, citing ' the sheer enormity of the task' and implying that he would need to understand all the code to do so. This seems to confuse robustness with being error-free. In my opinion, errors in one part of Horizon may well be successfully mitigated by a countermeasure in another part. It is important to understand the countermeasures, not the whole of the code. Mr Coyne has not done this.

562 This confusion seems to be repeated at his paragraph 5.110, where he states that Horizon is 'neither infallible or totally robust'. In my opinion, robustness has nothing to do with infallibility, and there is no such thing as total robustness. Robustness is a matter of dealing with a variety of threats, including software errors, so as to make their consequences acceptable in a business context. This involves the use of a range of countermeasures, none of which Mr. Coyne has discussed in his report.

POL00029046

563 In para 5.11, Mr Coyne states (contrary to his agnostic paragraph 5.88) that 'the electronic processes in Horizon are relatively robust'. However, in several place he implies that the robustness of Horizon may have varied over time, implying that at some times it may have allowed many errors to affect branch accounts.

564 I have addressed this question in section 7.8. There I showed that , however much the robustness of Horizon may have varied over the years, the sum total over all years of bugs which have affected claimants' accounts is very small compared to their losses. If the sum of a quantity over all years is very small, then fluctuations in that quantity over the years do not matter.

565 In the time available to me since receiving Mr Coyne's report, I have not been able to address all the detailed points he makes in paragraphs 5.82- 5.200. Those that I have been able to address are in Appendix H. I intend to address them in more detail in my supplementary report.

POL00029046

8. THE EFFECT OF HORIZON BUGS ON BRANCH ACCOUNTS

8.1 Horizon Issue 1: My Opinions

566 **Issue 1**: To what extent was it possible or likely for bugs, errors or defects of the nature alleged at §§23 and 24 of the GPOC and referred to in §§ 49 to 56 of the Generic Defence to have the potential to (a) cause apparent or alleged discrepancies or shortfalls relating to Subpostmasters' branch accounts or transactions, or (b) undermine the reliability of Horizon accurately to process and to record transactions as alleged at §24.1 GPOC?

567 The question raised in issue 1 follows on directly from the question raised in issue 3. Given that there were bugs, error or defects in Horizon (as there are in all commercial IT systems), how effective were the robustness countermeasures of Horizon in preventing or limiting any errors in claimant's branch accounts? Issue 1 asks about some of the consequences of Issue 3.

568 In section 7 I examined the robustness countermeasures. I found that in building and supporting Horizon, Fujitsu applied a set of well-established countermeasures, which have been familiar to me throughout my career. I found that they applied them effectively. In particular I looked at the KELs, which record how countermeasures were applied to events that threatened to create discrepancies in branch accounts. I found that in those cases, the countermeasures worked effectively. Only in a small minority of cases was there doubt (from lack of information recorded in the KEL) or any potential to affect branch accounts erroneously.

569 In my opinion on part (a) of issue 1:

iv. **Significant detected defects:** if in some month there was a significant shortfall in any claimant's branch accounts (a shortfall of £300 or more), the chances of that having arisen from a bug or defect in Horizon which has been detected are very small indeed. I have assessed this quantitatively by a retrospective IT Risk Analysis[85], with the following result: the probability of any of the three known bugs introducing a discrepancy in a claimant's branch accounts in any given month is of the order of two parts in a million. To make that probability as large as one part in 10, there would need to be more than 50,000 distinct bugs in Horizon, each of which created errors in branch accounts comparable to one of the three known bugs. The figure of 50,000 bugs is to be compared to the handful of bugs possibly affecting branch accounts which have been disclosed (i.e. the three known bugs) or found by the experts. This figure is derived in section 8.5, by a simple calculation, using evidence which in my opinion has only a small margin of uncertainty. The result is stable under changes of assumptions; if the assumptions change, the result does not change much.

i. **Undetected defects**: the claimants have raised the possibility that shortfalls might be caused by defects in Horizon which were never detected, and may not be known about to this day. Because of the many countermeasures built into Horizon, the potential for any such 'unknown bugs' is very

[85] as described in http://prince2.wiki/Risk

POL00029046

small indeed. Any bug with significant impact on branch accounts would be known about. The net impact of unknown bugs on branch accounts is very small, compared with the impact of defects which are known about and were recorded in KELs.

ii. **Financial Impact of defects**: Because of (ii), the KELs are a good source of information about bugs and the effect they might have had on branch accounts. One can examine the KELs, determine in which of them there might have been an impact on branch accounts, and place a conservative upper limit on the amount of this impact. Doing this sum, correcting for factors such as any inefficiency of the KEL creation process, lack of detail in KELs, and limitations in the sample of KELs I have been able to examine, I have calculated an upper limit on the financial impact of bugs in Horizon on the claimant's accounts. This upper limit is very small. Even using very conservative assumptions, designed to favour the claimants, the total net impact of all bugs in Horizon on the claimant's branch accounts must be less than about 0.1% of the shortfalls claimed by the claimants.

570 Bugs in Horizon cannot account for even a small part of the claimants' shortfalls - either for all claimants taken together, or for any individual claimant.

571 In my opinion this result follows inescapably from the evidence.

572 In my opinion of part (b) of issue 1: the Horizon Core Audit Process was designed to create a secure, accurate and immutable record of what was entered into Horizon at the branch, and to record verifiably who had entered it. In my opinion, regardless of any other processing done in other parts of Horizon, the core audit database was an accurate record of transactions entered in the branch. It was carefully designed and tested, and I have seen no evidence that it ever failed in service. Therefore in any case of doubt about processing done in other parts of Horizon, this record was available to establish the true state of any branch's accounts, based on transactions entered in the branch.

573 These opinions apply both to old Horizon (pre-2010) and Horizon Online (HNG).

574 In section 7, I addressed Horizon Issue 3: To what extent and in what respects is the Horizon System "robust" and extremely unlikely to be the cause of shortfalls in branches? I said there that I would postpone addressing the second part of that issue, 'extremely unlikely to be the cause of shortfalls in branches' to this section. As my opinion on part (a) of Issue 1 makes clear, in my opinion on Issue 3, the robustness of Horizon made it extremely unlikely to be the cause of shortfalls in branches.

8.2 Unknown Bugs in Horizon

575 The claimants have raised the possibility that shortfalls might be caused by defects in Horizon which were never detected, and may not be known about to this day.

576 They may wish to imply that the financial impact of these unknown bugs is essentially unknowable, and so may have been very large. In my opinion this is not so, because of the robustness countermeasures built into Horizon.

POL00029046

577 Part of the purpose of robustness in any financial system is to ensure that far-reaching errors in accounts do not occur. An important part of this is to ensure that if they should occur, they are rapidly detected - and do not persist, unknown, for long periods. Horizon was a typical financial system in this respect. In my opinion its robustness countermeasures worked well.

578 A particularly important countermeasure was the manual inspection of data (MID), by the SPMs themselves, at various times - in customer transactions, in daily cash balancing, and in their monthly balancing and rollover. At these times, one can expect at least some of the SPMs to have been highly vigilant.

579 I assessed this aspect of robustness in some detail in section 7.5, using the evidence available about known defects in Horizon with known financial impact. See the following paragraph.

580 I can go a little further than this by making weak inferences about how a manager of a small business, such as an SPM, needs to prioritise his time in monthly balancing I assume the following, as best assumptions of SPM behaviour in reporting anomalies in their monthly balancing:

a) If a discrepancy is £1000 or more, the SPM probably needs to investigate it. If he cannot find the cause in his branch, the likelihood of his reporting it through a help line is 80%

b) If a discrepancy is of the order of £300, 30% of SPMs will report it

c) If a discrepancy is of the order of £100, 10% of SPMs will report it

d) For a discrepancy of £10 or less, it is usually not worth the SPM's time to investigate it (because errors in counting cash or stock are often larger than this); so, these are reported on less than 1% of occasions.

581 The reader is referred to section 7.5 for more detailed analysis leading to this opinion.

582 I noted there that for any bug which reveals its presence to the SPM before monthly balancing, some SPMs are likely to report it even if its financial impact is small. Several KELs provide evidence for this.

583 Even of this conclusion is made more conservative - assuming, in the claimants' favour, that SPMs do not report bugs as frequently as described above - it means that any bug in Horizon, with significant impact on branch accounts, will, after only one or a few occurrences, be reported by some SPM. Therefore it does not remain an unknown bug.

584 Note from the estimates above that, as the financial impact of one occurrence of a bug decreases, the expected number of occurrences needed for some SPM to report it will increase - but its financial impact on each occurrence is less, so more occurrences are needed for it to have a given financial impact.

585 This means that, whatever may be the average financial impact of a bug on one occurrence, before that bug has had a large total impact (for illustration, say £3000), it is very likely to be reported, and so will be known about. The probability of a bug with financial impact remaining unreported, and remaining unknown, is small.

POL00029046

586 It then follows that the total financial impact of unknown bugs - those that remain unreported - is small, compared to the total financial impact of known burgs.

587 There is one possible exception this analysis. The possible exception is a bug whose financial impact on any one occasion is so small that no SPM will ever report it (e.g. they always believe it arises from error in their branch; or it is not worth their time to report it)- but which occurs on so many occasions that its net financial impact is significant. I refer to these as 'micro-bugs' and shall analyse them on more detail in Appendix F. They do not alter the conclusion above.

588 Note that by the phrase 'unknown bug' I mean 'a bug which was not detected near the time it occurred'. I do not mean 'a bug we do not know about today'. Normally, when a bug with possible financial impact occurred, this resulted in the creation of a KEL, and the KEL is now in evidence. But I do not assume that this process of creating KELs was 100% efficient (in my opinion the efficiency was near 90%, but for calculation I will assume conservatively that it was 50% efficient); so some known bugs are not recorded in KELs, and some 15% of KELs have now been archived, so the experts have not yet seen them. I shall account for this possible inefficiency in the creation and retention of KELs, just as I shall account for KELs I have not yet been able to analyse in detail, by scaling up the financial impact of bugs recorded in KELs that I have analysed. This is the standard engineering sampling approach to measurement - used whenever measurements are known not to be complete.

589 This scaling up of the effects of known bugs is the standard engineering way to account for bugs which were known at the time they occurred, but for which the evidence no longer exists - rather than treating them as some complete unknown which might be arbitrarily large. I have shown here that they cannot be arbitrarily large. Their effects must be much smaller than the effects of known bugs.

8.3 Impact of Bugs on Claimants' Branch Accounts - Qualitative Opinion

590 Issue 1 asks about the financial impact of bugs in Horizon on claimants' branch accounts.

591 Following my analysis of robustness in section 7, it will now be clear that the answer to this question depends on the robustness of Horizon - not on how many bugs there were, but on how well the effects of these bugs were countered and mitigated by the robustness countermeasures, to prevent them from creating discrepancies or shortfalls in branch accounts.

592 Because of this, any simple counting or cataloguing of bugs - for instance, derived from KELs and Peaks, as Mr Coyne has done - does little to answer the question of Issue 1. For any bug or anomaly that might have affected branch accounts, it is necessary to consider the robustness countermeasures, and how they may have operated in that case.

593 In my opinion, the KELs and their associated Peaks are the main source of evidence available about how the countermeasures operated in live use of Horizon. They are the most direct evidence I have, closer to the coal face of Horizon in action than other summary reports or records of meetings. I have now examined more than 150 KELs from this viewpoint - asking what can be inferred from the KEL about what countermeasures were in operation, and how effectively they operated, in the light of my own

POL00029046

experience of the same robustness countermeasures. In my supplemental report, I intend to examine more KELs in this way.

594 While the KELs are by no means a perfect source of information about the operation of countermeasures - as they are often written tersely, assuming a deep knowledge of Horizon and the processes around it, and they often focus on how to help the branch, rather than on spelling out any underlying problem - nevertheless a clear qualitative picture emerges from this examination:

a) Countermeasures play a key role in countering effects such as hardware failures, communication failures and human errors, which occur with much greater frequency than software errors (because software has been extensively rested before it goes live). In this role, the countermeasures are very effective. Many KELs show the effectiveness in this respect of countermeasures TIN, DEP, RDS, DEA, MID, and UEC , as well as others. It can be inferred that for any software bug which has the same effects as a hardware error, communication error, or user error, the same countermeasures will prevent any adverse effect on branch accounts. This is the case for many of the KELs which Mr Coyne and I have examined - although he has not pointed this out.

e) Using system logs and event logs, Fujitsu were able to trace the full sequence of events in the branch, to determine with high reliability the events leading to any anomaly noted in the branch, and to assess whether it arose from human error, or from some other cause which might be a software error.

f) If a software error was suspected - and especially if it might have any effect on branch accounts - Fujitsu were in most cases able to rapidly identify its cause ; to use system logs and event logs to identify any branches affected, before or after the effect was reported; in most cases to reproduce the effect in test, and diagnose the cause. They could usually suggest workarounds for any branch during the period while the bug was being fixed.

g) The KELs were written by a fairly small group of people, who according to Mr Parker's witness statement were mainly in the third and fourth lines of support. The KELs demonstrate a comprehensive grasp of Horizon and the processes around it. Problems were usually rapidly diagnosed, and as the Peaks demonstrate, the processes for correcting the software were tightly managed.

h) My impression from the KELs is of a support team that knew what they were doing, on an IT system that they understood well - trying, and succeeding, to rapidly diagnose and close down any problem, to remove it from their future workload.

595 Therefore in my opinion the robustness countermeasures in Horizon worked well in preventing bugs and other effects from introducing inaccuracies in branch accounts. This applies both to the automated countermeasures within Horizon itself, and to the manual countermeasures applied to any effect which was not countered automatically. In this respect I stress that any system which relied only on automated countermeasures within it, and had no manual countermeasures to back them up, would in my opinion

POL00029046

not qualify as a robust system. I have never seen a major commercial system without manual countermeasures.

596 Because the robustness countermeasures worked well, as the KELs record, the vast majority of anomalies recorded there were not bugs with adverse effect on branch accounts. If a bug had had such adverse effects, it would with high probability be recorded in a KEL.

597 Therefore in my opinion, because the robustness countermeasures worked very well, there were very few bugs which introduced inaccuracies in branch accounts, and their financial impact across the PO branch network was very small.

598 Mr Coyne's report appears to imply otherwise. But he has not analysed the KELs or Peaks to sufficient depth to consider the effects of robustness countermeasures. Therefore his report contains little or no analysis to contradict my opinion. I have examined 62 of the KELs he relies upon[86], and they confirm my opinion. This analysis is shown in a table in Appendix E. My conclusions on robustness, as demonstrated by those KELs, are contained in section 7.6.

599 I have stated this opinion in qualitative terms. However, Horizon issue 1 asks about the extent of impact of bugs, and I go on to assess the extent in quantitative terms.

8.4 Measures of Extent

600 Horizon Issue 1 begins with the phrase 'to what extent'. This implies that the experts are not being asked to decide on a yes/no question, but are being asked to assess the extent of some effect. This raises the question: against what scale of measurement against should the 'extent' be assessed? I need to define the scale of measurement, to make my opinions clear.

601 There are two ways in which this extent might be measured:

- It might be measured by the expected net financial impact of all bugs, over various 'ranges' or scopes - (a) across all branches over the lifetime of Horizon, or (b) across all claimants' branches, or (c) on one claimant's branch in any one month
- It might be measured by the number of bugs over the same scopes - (a) on all branches over the lifetime of Horizon, or (b) on claimants' branches, or (c) on one claimant's branch in any one month

602 I shall express opinions about both of these measures. In my opinion the first of the two measures is more useful, for two reasons.

a) There may be a number of bugs which affect branch accounts, but whose likely financial impact is trivial. If time is spent considering these bugs with non-zero but trivial financial impact, it might divert attention from considering the smaller number of bugs with significant financial impact, which could have made a more important difference to claimants' branch accounts. Focus on the financial impact of bugs will help in narrowing the scope of enquiries.

[86] These were the KELs I found referenced in the body of his report

POL00029046

i) The first sense of 'extent' gives a way to assess not only the relative importance of different Horizon bugs, but also to assess the absolute importance of each one, asking the question: does any bug on its own (or a set of bugs considered together) provide a possible account for some significant part of the shortfalls asserted by the claimants?

j) countermeasures were designed to limit the financial impact of bugs (extent 1) rather than the number of them (extent 2), so the available evidence bears more directly on extent 1

603 For both measures, what I am able to infer from the evidence will be an upper limit on the extent, rather than an estimate of the extent. This is because there are KELs which record incidents which might be caused by bugs, and which might have financial impact. Here, the word 'might' applies with force; there is often not sufficient information in the KEL to conclude that there definitely was an impact, only to conclude that there might have been. This allows me to estimate only an upper limit on the sums over KELs.

604 There are difficulties in measuring the second extent:

a) The robustness countermeasures were generally designed to detect, and to minimise, the first extent (financial impact), with less regard to the second extent - because it is less important for the business. Work was prioritised according to financial impact. So what evidence remains relates more closely to the first extent than to the second.

b) There are difficulties of definition of the second extent, which make it difficult to sum in any quantitative sense. Does the definition refer to occurrences of bugs in any branch at any time, or to distinct types of bug, however many times they may occur? If it is the former, it is almost impossible to estimate. If it is the latter, what is it that defines two distinct types of bug, as opposed to variant occurrences of the same type? As one 'type' of bug may be caused by the effects of several apparently unrelated pieces of source code or reference data, this question of distinct types of bug has no simple answer. I have found no succinct and satisfactory answers to these questions.

c) because of these difficulties, the second extent is not simply additive as the first one is. It is not easy to proceed from scope (a) (all PO branches) to scope (b) (all claimants' branches) to scope (c) (any branch in one month)

605 Therefore, my opinions about the second sense of extent are a great deal more uncertain, and in my view harder to interpret, than my opinions on the first.

606 For the first sense of extent, I can say more about the absolute scale of measurement. There are 561 claimants. Each one has provided in section 8.1 of their claim summary the amount they claim to have repaid to PO and/or owe to PO; and in section 3.1, a list of the separate shortfalls they claim to have experienced, with dates or date ranges for each shortfall. For a few claimants, the amount repaid exceeds the sum of individual shortfalls, so I need to assume that the list of individual shortfalls is incomplete. If we make the sum of shortfalls equal or exceed the repaid amount for each claimant, the sum of all

POL00029046

shortfalls claimed by all claimants is approximately £18.7 million. Without making that correction, it is about £17.1 million.

607 This provides an absolute scale of measurement for any bug or set of bugs in Horizon. Does that bug, or set of bugs, offer a possible account for shortfalls which would make up a significant part of the £18.7m?. When assessing the absolute significance of any bug or set of bugs, I shall apply that criterion.

608 The same criterion can be expressed in a different way. The period of tenure of each claimant in months is known. The sum of these tenure periods over all claimants is just over 52,000 months . Therefore the shortfalls claimed by the claimants amount to an average shortfall of just under £360 for each month of their tenure. For each Horizon bug, therefore, we can ask: (a) over what proportion of the 20 year lifetime of Horizon was that bug active? and (b) during that time, what might have been its average impact on branch accounts, compared to £360 per month?

609 In this way , for any bug or set of bugs, my report is intended to assist the court in measuring those bugs against the claim that bugs in Horizon accounted for a significant part of the shortfalls the claimants experienced.

610 To understand the phrase 'bugs, errors or defects', I need to include various classes of defect, including:

- Software errors in Horizon source code, either in the branch or in the back end, either developed by Fujitsu or in some underlying software product they used (such as Riposte or Oracle)
- Errors in reference data, most of which was maintained by PO staff and which determined how the Horizon software operated
- Errors in operational use of software in the back end - such as errors in scheduling of batch jobs in the back end

611 I shall consider all of these to be included in the definition of Horizon issue 1.

612 There are a number of different ways of assessing the financial impact of bugs, which I shall address in the following sub-sections:

a) The impact of the three Horizon bugs addressed by Mr Coyne in paras 5.4 - 5.14 of his report.

b) Assessments of the net impact of all bugs referred to in KELs

c) Data on claimants' shortfalls as provided by the claimants, to assess the impact of all Horizon bugs (known and not identified) on claimant's branches

d) Evidence cited by Mr Coyne

613 The most important of these analyses , and the ones which gives the most clear-cut result, are the analyses under (b) (which in any case include the analysis under (a)). The first of these results in an upper limit on the impact of all Horizon bugs on claimants' branch accounts. The limit is very small - less than 0.15% of the total shortfalls experienced by the claimants. This implies that bugs in Horizon cannot have accounted for the claimants' shortfalls.

POL00029046

614 The analysis by method (c) is presented only as a backup of the analyses in (b). The upper limit which it leads to is larger than the limit from method (b) (being approximately 8% of the total shortfalls experienced by the claimants). But it has the merit of being based on completely independent evidence (that provided by the claimants themselves), and so provides independent confirmation of the result from (b).

8.5 Scaling of Financial Impacts of Bugs

615 I shall assess the financial impact of bugs in Horizon over three different scopes:

a) Across all PO branches, during the lifetime of Horizon

b) Across all claimant branches, while they held them

c) On a single claimant branch in a single month.

616 It is possible to relate the financial impacts on these scopes, by numerical scaling factors. I calculate those scaling factors in this sub-section.

617 Over the period 2000 - 2018, the PO network has consisted of more than 11,000 branches. The mean number of branches in all years over the period has been about 13,560. This figure is derived from the spreadsheet referred to at paragraph 178 of Angela Van Den Bogerd's witness statement, assuming that the spreadsheet is accepted. If this evidence is accepted, the number of 'branch months' (a single branch, trading for a single month) has been 13,560 * 12 * 19 = 3,091,680. This is the number of monthly branch accounts that have been produced.

618 This means that for a typical PO branch, the scaling factor between scope(a) and scope (c) for the impact of bugs in Horizon is a factor of 3 million.

619 For claimants' branches, rather than typical branches, the scaling factor of approximately 3 million may need to be adjusted for two possible effects:

a) It might be asserted that claimants' branches are more or less likely than other branches to be hit by bugs in Horizon, because of some special property of claimants.

b) Claimants' branches may, on average, be smaller or larger than typical branches across the PO network. If they are smaller, they handle fewer transactions in a month, and so are less prone to Horizon bugs in those transactions.

620 It seems implausible to me that there is some special factor about claimants' branches, which makes them much more prone to bugs in Horizon - bugs which one would expect to strike any branch at random. Nevertheless, I have considered the possibility carefully in Appendix F. I have shown there that there is no significant difference between claimants' branches and other branches, in proneness to bugs in Horizon.

621 It appears, from the spreadsheet attached at paragraph 179 of Angela Van Den Bogerd's witness statement, that the claimants' branches are, in terms of customer transactions carried out per day, smaller than the average across the whole PO branch network.

POL00029046

622 If this spreadsheet is accepted, it implies the following about claimant's branches:

- From summing rows of the spreadsheet, the 561 claimants' branches carried out 558,000 customer transactions per week in 2007.
- This is 558,000/6 = 93,000 transactions per day, assuming a PO branch is open for 6 days a week.
- Across 561 claimant branches, this is an average of 93,000/561 = 165 customer transactions per branch per day.

581 For comparison with this figure, I need to estimate the average size of branches across the PO Network. I have done this using two pieces of evidence. Mrs Van Den Bogerd's witness statement says that across the whole PO network, there are approximately 48 million customer transactions per week, or 8 million per day in 2017 (assuming again that branches are open for 6 days in the week).

582 In section 15 of the architecture definition for Horizon[87], dated in 2003, there is a table of EPOSS Product volumes at that time:

Volume	Historical Peak	Design Volume
Peak Month	100,195,596	120,234,715
Peak Week	33,637,564	40,365,077
Peak 2 Days	14,876,498	17,851,798
Peak Day	8,192,874	9,831,449
Peak Hour	1,171,581	1,405,897
5 Minutes (Per Sec)	328	394

Table 8.1 - Horizon EPOSS Transaction Rates (2003)

583 The best estimate I can make of the average daily volume in 2003 from this table is to divide the peak month by 26 working days. This gives an estimate of approximately 4 million transactions per day in 2003, compared with 8 million transaction per day in 2017.

584 I therefore estimate that the average volume of transactions over the period 2000-2018 has been approximately 6 million transactions per day, mid-way between the 2003 figure and the 2017 figure (assuming Mrs Van Den Bogerd's evidence is accepted. I note that Mr Coyne quotes the same figure from her) .

585 .To estimate how much smaller the typical claimant's branch is than the average PO branch:

- The average transaction rate for all PO branches is 6,000,000/13,650 = 439 customer transactions per branch per day
- So, in terms of customer transactions per day, the typical claimant branch was smaller than the average PO branch by a factor 165/439 = 0.37

623 So claimants' branches, being generally smaller than the PO average, have fewer transactions per month, and so are less likely to be hit by a Horizon bug in a given month. This increases the scaling factor above, between scopes (a) and (c) from about 3 million to about 8 million.

[87] TD.ARC/001

POL00029046

624 I illustrate what the factor of 8 million means using a hypothetical example of a bug which has occurred 16 times over the lifetime of Horizon, with mean financial impact on these occasions of £1000. Call this Bug A. The financial impact of Bug A is similar to that of the Suspense Account bug.

625 If we selected a claimant's branch and a month at random, then the chances of Bug A occurring at that branch in that month are only 16 in 8 million, or 2 in a million - an extremely small probability.

626 Different types of bug occur independently of one another, so their probabilities of occurring are additive. If there were a second bug similar to the hypothetical bug above - call it Bug B - then the chances of either Bug A or Bug B happening to one branch on one month are twice the previous figure - 32 parts in 8 million.

627 If there were 100 similar bugs - called Bug A, Bug B, Bug C, ... Bug Zz - the chances of any one of them happening to one branch on one month are still only 100 * 16 parts in 8 million, or one part in 5,000. This is still a very small probability.

628 It then follows that in order for one occurrence of a bug, of similar financial impact to the Suspense Account bug, to have even a one-in-ten chance of occurring to one branch on one month, there would need to be 50,000 such distinct bugs - because 50,000 * 16 / 8,000,000 = 800,000/8,000,000 = 1/10. There would have to be a Bug A, Bug B, Bug C, and so on, in a list with 50,000 distinct bugs.

629 Even if there were some 'super-bug' - with financial impact ten times larger than the suspense account bug - there would have to be approximately 5,000 such super-bugs to give a one in ten chance of affecting a claimant's branch accounts in a given month. There is no evidence for even one such super-bug - let alone for 5,000 different ones.

630 I have made this calculation in an Excel spreadsheet, which is attached to my report. For convenience it is shown here:

Item	Label	Central Estimate		Source
Mean number of branches in PO network	A	13560		Spreadsheet with AVDB WS
Years lifetime of Horizon	B	19		2000 to 2018
Total branch months	C	3091680		C=A*B*12
Claimants branch size/typical branch size	D	0.37		Spreadsheet with AVDB WS
Scaling factor from all PO to one claimant branch month	E	8355892		E=C/D
Number of occurrences of Suspense Account bug	F	16		Evidence on Suspense Account bug
Mean financial impact per occurrence of Suspense Account bug (pounds)	G	1000		Evidence on Suspense Account bug
Chance of Suspense account bug occurring to a claimant's branch in one month (chance is 1 in N , where N is shown)	H	522243		H=E/F
Number of different bugs, similar to the Suspense account bug, needed to give 1 chance in 10 of causing a shortfall of £1000 in a claimant's branch account in any given month	J	52224		J = H/10

Table 8.2 - Calculation of number of bugs (as defined in row J of the table)

POL00029046

631 The rows have been labelled A, B, and so on - so that the calculation leading to any calculated row can be easily seen and checked - as in the example C = A*B*12.

632 The claimants have never asserted that there are as many as 50,000 distinct bugs in Horizon, with each bug on the same scale of financial impact as the Suspense Account bug. Their case rests on two or three known bugs of this scale, and on the unproven assertion that there may be others.

633 If any claimant were to assert that, for instance, a deficit of £1000 had occurred in his branch in a particular month, caused by a bug similar to the Suspense Account bug, the chances of that assertion being correct are extremely small, because Horizon bugs strike so rarely - unless Horizon contained of the order of 50,000 distinct bugs of that kind (and even then, the chances are only one in ten).

634 Mr Coyne has examined more than 5,000 KELs, and not found definite evidence for even one bug with impact similar to the Suspense Account bug - let alone 50,000 of them.

635 The implication of this result - the very small probability of any error in one months' accounts from a bug in Horizon - is that the accounts for any branch on any month are overwhelmingly likely to be correct (apart from effects such as delayed TCs, which are corrected after a variable delay).

636 In my experience, no commercial IT system could ever go live with as many as 50,000 serious bugs - and certainly could not have the good in-service record over 18 years that Horizon has had.

637 In my opinion the claimants' assertion - that some significant part of their losses was caused by bugs in Horizon - is even more implausible than I have described. It would require not just 50,000 distinct bugs, each with large potential impact on branch accounts - but 50,000 bugs, each of which evaded the many countermeasures built into Horizon, in order to affect branch accounts It would also require branches not to have been compensated for their losses, as they were in the case of the Suspense account bug.

638 Having calculated the scaling factor of 8 million between scope (a) of the extent of bugs (impact across all branches in all the lifetime of Horizon) and scope (c) (impact on one claimant's branch in one month), it is easy to relate these two scopes to scope (b) (impact across all claimant's branches).

639 Evidence submitted by the claimants implies that claimants ran branches for a total of just over 52,000 months. Therefore, the scaling factor between scopes (b) and (c) is a factor of 52,000. Alternatively, the scaling factor between scope (a) and scope (b) is 8,000,000/52,000 = 160.

640 I shall call this ratio - the amount by which the impact of bugs in Horizon (a) on all PO branches is expected to be larger than their impact (b) on all claimant's branches - the **claimant scaling factor**. I shall take it to be 160.

641 There is one immediate consequence of this scaling factor. The total of all shortfalls claimed by the claimants is £18.7 million. If, as the claimants assert, some large part of this (for illustration, say 50%) was caused by bugs in Horizon, the total impact of these bugs across all PO branches would be £18.7M * 0.5 * 160 = £1,496 million. This figure, of almost £1.5 billion, is the sum which PO would have gained from its SPMs, through bugs in Horizon, over the life of Horizon. The figure follows from claimant's claim and the assumption, justified in Appendix F, that the accounts of any branch would be affected by bugs in

POL00029046

Horizon in the same way as those of the claimants. In my opinion, as will be described below, it is very unlikely that there could have been be such a gain for the PO.

8.6 Analyses of the Three Errors Cited By the Claimants

642 I have analysed these three bugs using evidence available to me, and then on November 18th I received the second witness statement of Torstein Godeseth, which addresses those three bugs, and reaches conclusions similar to my previous conclusions. Where relevant I comment on Mr Godeseth's conclusions.

8.6.1 The Receipts/Payments Mismatch Issue

643 This issue is cited in paragraph 5.6 of Mr Coyne's report. It involved a bug in Horizon which was triggered by a rare circumstance (which one would not expect to be exercised in testing) and which had an effect on branch accounts. If the Mr Godeseth's evidence about this bug is not accepted, I shall revise my opinions accordingly. They are based on written evidence - particularly on a written analysis[88] by Gareth Jenkins - as well as Mr Godeseth's witness statement. In my opinion, the two are consistent.

644 At paragraph 35 of Mr Godeseth's witness statement, he says this bug was detected by routine monitoring of system events by the Fujitsu System Support Centre (countermeasures RDS, MID). In the same paragraph, he says that the bug would also show as a discrepancy in POLSAP (another example of RDS)

645 This incident involved a complex sequence of events during branch balancing, which was not detected by some of Horizon's resilience countermeasures, but which was detected by others. It was later the subject of thorough investigation by Fujitsu.

646 The bug only occurred when the user followed a rare sequence of actions during branch balancing. This sequence was to cancel one part of the balancing process (balancing one stock unit), but to proceed after that with balancing other stock units. While not actually forbidden, this sequence of actions would usually have occurred only when the user had misunderstood what he was doing (thinking he was balancing for a 4-5 week Trading Period, when he was actually balancing for a 1-week Balancing Period). Hence the circumstance was rare and had not been exercised in testing.

647 The effect of the bug was to record erroneously in the BRDB that the stock unit whose balancing had been cancelled was in balance; while also recording the imbalance on other systems (including POLSAP). My analysis of the bug in terms of Horizon's robustness countermeasures is as follows:

- Because the bug occurred in one of the countermeasures (the correction of user errors by branch balancing, UEC), that countermeasure created the error, rather than correcting it.
- Because the operation involved was not a double-entry operation on the BRDB, the countermeasure of checking the double-entry constraint DEA did not catch it.
- Horizon kept redundant copies of the information involved and checked them. The checks from this countermeasure produced error messages in logs (RDS).

[88] Receipts&Paymentsv0.4.docx

POL00029046

- Once Fujitsu were alerted to the error, they were able to look at the error messages to find which branches were involved and the amounts involved (MID), and to find out the causes of the error.
- The error was fixed within about 2 months of its first occurrence (BFC).

648 This is a fairly complex incident, which illustrates how many resilience countermeasures there are in Horizon, and how even when some of them do not catch an error, others will do so. Most of the other KELs which I have examined tell a simpler story of how one or other countermeasure trapped some error (an error in software, or a user error).

649 The net quantitative impact of the receipts/payments mismatch was approximately £20,000 across 62 of the 11,000 branches. Paragraphs 42 and 43 of Mr Godeseth's witness statement describe how the branches were compensated. If this is correct, in the event, no SPM suffered any loss.

650 Because this shortfall was carefully investigated, it is known that none of the claimants' branches was affected. Without this knowledge, I would expect on statistical grounds that the net effect on all claimants' branches would be £20,000 divided by the claimant scaling factor, described in section 8.5. The claimant scaling factor is 160.

651 This amount would be approximately £125 spread across all claimants, less than 0.001% of the full shortfall experienced by all claimants. Thus even a prominent and thoroughly investigated bug would have made no significant contribution to the claimants' shortfalls.

652 This accords with the previous result, noted in section 8.5, that a very large number of similar bugs (of the order of 50,000 or more) would be needed to account for the claimants' shortfalls.

8.6.2 The Callendar Square/Falkirk Bug

653 The Callendar Square bug is described in two KELs, JBallantyne5245K and JSimpkins338Q, and in several Peaks PC0075892, PC0083101, PC0086212, PC0103864, PC0126042, PC0126376, and PC0193012. It first arose in 2000, and was not fixed until release S90 in 2006. I shall describe the nature and effects of the bug, based on the KELs and an analysis by Gareth Jenkins, and then summarise my opinion on the significance of the bug for Horizon Issues 3 and 1.

654 The cause of the bug was a failure of data replication in the underlying Riposte software in the branch, which sometimes occurred when transferring stock between stock units. This was caused by a timeout or locking problem somewhere inside the Riposte product, which it was not possible for Fujitsu to fully understand at the time, or for the experts to understand now. Internal design details of system software products like Riposte are generally not made available to developers who use that software.

655 The result was typically that the stock would disappear from the sending stock unit, and not reappear in the receiving stock unit - a failure of double entry accounting (DEA) which was not evident to the SPM at the time. At paragraph 13.6 of his witness statement, Mr Godeseth comments on this failure of double entry accounting.

656 In my opinion , under the later Horizon Online software this failure of DEA might have been immediately manifest as a failure to send a zero-sum basket to the BRDB. But in Horizon, apparently it

POL00029046

was not immediately detected, so in this respect Old Horizon was possibly less robust than Horizon Online.

657 While the failure was not immediately visible to the SPM at the time of the stock transfer, it would always be visible later when balancing. stock units. It was also, as Mr Godeseth says at paragraph 13.7 of his second witness statement, soon visible to Fujitsu in two different ways (a flag from overnight processing, and a system event). If this is correct, it was robustness through the countermeasure RDS.

658 So in the normal course of events, the SPM would see a discrepancy of some large and easily identifiable sum (because stock unit transfers generally involve larger sums than customer transactions) and would know, since he had not made any mistake, to call the help desk. This was countermeasure MID. As is shown be the Peaks, the presence of the Riposte error was easily identifiable from system logs, so the help desk would know it was not a user error and PO could correct any discrepancy if it arose in the branch accounts.

659 Thus for any SPM who was in good control of his branch numbers, and alert to discrepancies, there was little chance of this bug leading ultimately to an error in his branch accounts; he would require it to be corrected. This was ensured by a combination of the countermeasures RDS (e.g. in event logs) and MID (by both SPMs and Fujitsu)

660 At paragraph 15 of Mr Godeseth's witness statement, he says that this bug had impact on branch accounts in 20 cases. For the receipts/payments mismatch bug, there is evidence that affected branches were compensated. Because of this evidence, and because Fujitsu could always spot any occurrence of the bug in event logs, and because neither PO or SPM wanted SPMs to suffer shortfalls from bugs in Horizon, I would expect the SPM to be left with a shortfall (i.e. not compensated) in only a small minority of cases, if any cases. So in my opinion the net shortfall caused by all its occurrences would be possibly zero, and in any event at most a few thousand pounds.

661 Because Fujitsu had designed the counter software assuming that Riposte replication worked correctly, and could not anticipate in what ways it might not work, in my opinion it would have been very difficult for Fujitsu to fix the problem or correct it. Fujitsu were reliant on Escher to fix the problem; and apparently Escher did not do this for some years.

662 To summarise my opinions on the significance of the Callendar Square bug for Horizon issues 3 and 1:

- It was not detected immediately by the countermeasure DEA, when in my opinion it might have been detected (although possibly the Horizon architecture, dependent on Riposte replication, made this very difficult)
- However, it was later detected in branch balancing, and corrected if necessary, by the countermeasures RDS and MID. Overall, Horizon's robustness worked well.
- Therefore, like the other two known bugs addressed in this sub-section, its possible financial impact on claimants' branch accounts was very small indeed

POL00029046

- It took several years to fix, but in my opinion this does not reflect badly on Fujitsu. They were reliant on the Riposte product supplier, Escher, to fix the problem.

8.6.3 The Suspense Account Bug

663 This is referred to in paragraph5.12 of Mr Coyne's report. Like the receipts/payments mismatch, it concerns the process of balancing and rollover of stock units, using suspense accounts.

664 It was analysed in depth[89] by Gareth Jenkins of Fujitsu. His summary of the effect was: *'The root cause of the problem was that under some specific, rare circumstances some temporary data used in calculating the Local Suspense was not deleted when it should have been, and so was erroneously re-used a year later'.*

665 To understand the 'specific, rare circumstances', the effect of robustness countermeasures, and the financial impact, it is necessary first to understand what was supposed to happen, and then to understand how it went wrong.

666 When certain types of data in the branch database are no longer needed for trading, they may nevertheless need to be retained for some time for various purposes, or be kept for longer in an archive. Ultimately, the space taken up by those records in the BRDB needs to be recovered, to stop the database growing without limit. There are therefore policies for initially making records inactive by a process of 'logical deletion' (which means, not actually deleting the record, but marking it as inactive and due for later deletion), for archiving of data, and for ultimate physical deletion of records.

667 These policies are different for different tables of the BRDB. They may need to change from time to time. When they do, there is a risk of transient problems - when the time window of some archiving and deletion policy changes, and records in some table fall between windows. This is what happened in the suspense account bug.

668 A branch will from time to time want to stop using a stock unit. When it does so, the records in the BRDB for the stock unit are first 'logically deleted' - to prevent the stock unit being reused before it can be reused - and later archived and physically deleted. After that, the stock unit can be reused (i.e. another stock unit with the same identity can be created and used).

669 There was a change in the archiving policy in late 2010, which meant that for a short period, the balancing discrepancies for some stock units which were to be deleted became 'orphaned', and escaped the process of archiving and physical deletion. This meant that if, after archiving, the stock unit was recreated and reused, the balancing discrepancy associated with the old stock unit (which had actually been cleared, before the old stock unit was deleted) became wrongly associated with the new stock unit a year later (when the same trading period number between 1 and 12 came up in the next year) - and the SPM was asked, wrongly, to clear it again.

670 Usually, this error would have a small financial impact and would be hard to detect. This is because the balancing discrepancy in a stock unit is expected to be small - arising from mis-counting of stock or cash;

[89] LocalSuspense v0 5.docx

POL00029046

so that if the same discrepancy was then wrongly added to another discrepancy a year later, the difference might not be noticed. For a small discrepancy, an SPM may choose to just accept it, rather than make the effort to try to understand it.

671 Of the 14 branches affected by this bug, for 12 of them the amounts were less than £161 and the SPM did not raise any query. However, for two of the branches the amounts were larger, of several thousand pounds. These large discrepancies arose not because those branches were very bad at counting stock, but for a different reason.

672 Normally, a stock unit should accurately reflect both the cash and the stock in it. When preparing to delete a stock unit, the SPM is expected to move all stock and cash from it into other stock units in the branch, using facilities in the counter software to do this. However, there is a short cut. Because the stock in a branch is not necessarily physically segregated into different stock units, the SPM might make no transfer of stock - but instead, when the stock is sold, simply credit the cash to another stock unit. Then the other stock unit will have a surplus in its cash and stock, and the stock unit to be deleted will have a deficit. These two discrepancies will cancel in the monthly balancing process, so there is no cost to the SPM.

673 This short cut was not recommended by PO in the branch trading manual , but was available to anyone who understood how balancing worked across several stock units - that is, to most SPMs. It saved the trouble of moving stock between units. One of the branches affected had used this short cut, and had a large deficit in the stock unit which was to be removed. Because of the archiving problem, that large deficit wrongly reappeared a year later.

674 The description above agrees with the account at paragraph 48 of Mr. Godeseth's second witness statement.

675 The circumstances for this bug to have a large financial impact on a branch were rare in four respects:

- A branch wishes to stop using a stock unit - which only happens occasionally
- The stock unit is terminated during a short overlap period, caused by a change in archiving policy
- The branch then wishes to re-create the stock unit (to reuse its identity) - another occasional event
- when 'running down' the stock unit before terminating it, the SPM used a short cut leading to a large balancing discrepancy

676 In combination, these circumstances were so rare they had large effects (greater than £200) on the accounts of only two branches.

677 We can consider what this bug implies about the resilience measures in Horizon:

a) The archiving of BRDB data is not done in double-entry transactions, and the table used to compute the initial discrepancy in a stock unit (which was a table used to prepare the BTS) was not subject to double-entry constraints. So the DEA countermeasure did not catch the error.

a) However, the status of each stock unit was also redundantly held on POLSAP, which did apply double entry constraints and did not have the same archiving policies; so once the error was

POL00029046

discovered, the true position for affected branches was easily discovered from POLSAP and corrections were manually applied to the BRDB. This was an example of redundant data storage (RDS), using manual inspection of data (MID) followed by a successful manual workaround (WOR).

b) Because there was no automatic cross-check of stock unit positions between POLSAP and the BRDB, the error was not detected automatically by RDS.

c) There was a delay of a year in manually finding the bug and correcting it - because when the branch most affected reported the problem to PO, PO simply corrected the data centrally and did not inform Fujitsu. It was only a year later, at the next recurrence, that Fujitsu became aware of the problem and fixed it. This business process problem between PO and Fujitsu was a lapse in bug fixing and correction (BFC)

d) In spite of the bug, the core audit process maintained an accurate record of what had happened in the branches (another redundant copy, stored in a secure kernel, SEK), which was useful in diagnosing and fixing the problem.

678 Therefore none of the automatic robustness countermeasures detected the bug. It was detected by an SPM raising a query (as one would expect to happen, for any financially significant effect), and in diagnosing, correcting, and fixing it, redundantly-held copies of the data played an important role.

679 Because (according to the note by Gareth Jenkins) the affected branches were easily identified, and the impact was manually reversed for those branches , no SPM was adversely affected by this bug. Without this correction process, the financial impact of the bug would have been about £10,000 across the 16 branches. As before, no claimant branches were involved ; but if there had been other comparable bugs, their expected impact on the accounts of claimants' branches would be £10,000 divided by the claimant scaling factor, which is 160 (as described in section 8.2), or approximately £60 across all claimants Compared to the total shortfall of £18.7 million suffered by all claimants, this figure is approximately 0.0003%.

8.6.4 Opinion on the Three Identified Bugs

680 Because these three bugs all led to effects on branch accounts, they were all investigated carefully, by staff in Fujitsu with a deep knowledge of Horizon. These investigations indicate to me that, as I would have expected, Horizon is a highly complex system, and to fully understand some errors in it (and the robust handling of those errors) requires a knowledge of many of its component systems to some depth.

681 The experts have not had the time to do this deep analysis for more than a few errors, including these, and it would be unrealistic to expect the reader to understand these to the same depth.

682 The conclusions I draw from analysing these three bugs are:

- There are extensive robustness countermeasures in Horizon, of many types - so that even in the rare case of bugs like these which are not handled by the fully automatic countermeasures, manual

POL00029046

countermeasures enable the bugs to be rapidly diagnosed and corrected, as soon as they are known about.

- Any error in Horizon whose financial impact is greater than about £1000 in one month is highly likely to be reported by a significant proportion of the SPMs who experience it, and so to be the subject of manual investigation. In all the three cases considered here, this manual investigation was successful - which further confirms the robustness of Horizon.
- The expected financial impact of these bugs taken together on claimants' branch accounts was only a fraction of a percent of the total shortfall experienced by all claimants. So bugs like these, even if a very large number of them existed (which I have seen no evidence for), cannot account for the claimants' shortfalls.
- In the two cases where we know there were some shortfalls in branch accounts (Receipts/Payments Mismatch, and Suspense Account), the evidence appears to imply that the branches were compensated for the losses.

683 The last point seems to me to have a general bearing on the claimants' case. There are two kinds of bugs- known bugs, which were noted and detected at the time, and 'unknown bugs' which were not detected at the time. In section 8.2, I showed that the financial impact of unknown bugs is smaller than the impact of known bugs. If, as these examples imply, in the case of known bugs the SPMs were compensated, then known and unknown bugs together would lead to little or no personal loss to SPMs .

8.7 Financial Impact of All Bugs - Main Analysis

8.7.1 Method of Analysis and Conclusions

684 In section 7 of this report, when discussing robustness countermeasures in Horizon, I reached two conclusions:

b) If any bug had impact on branch accounts, (with the exception of micro-bugs, discussed later in this section), on a significant proportion of the occasions when it occurred, it would be reported to the help desk by some SPMs

e) Any anomaly reported by an SPM which had the potential to affect branch accounts would, with fairly high probability, result in a KEL and an investigation by Fujitsu

685 It is therefore possible to use the KELs as a measure of the extent and financial impact of bugs in Horizon - by counting all the KELs which might be bugs with financial impact, summing their maximum possible financial impact, and making allowances for any inefficiencies in the processes (a) and (b).

686 The analysis relies on evidence from KELs rather than other documents. I have examined a large number of KELs which are not cited directly here, and my analysis is contained in tables in Appendix D to this report.

687 The numerical analysis is in principle quite simple. It is to count the bugs in KELs; sum their financial impact; and then make allowances for inefficiency in recording bugs in KELs.

POL00029046

688 In practice, the analysis is made more complex by two factors:

- There are more than 8300 KELs, and it has not been possible in the time available to analyse all KELs to the depth required; so I have had to analyse the KELs on a sampling basis, and to correct the final result for the sampling.
- KELs, and the Peaks they refer to, are not a complete record of the nature of a bug, or its investigation, or its financial impact, or the branches affected; that is not their purpose. KELs and Peaks assume a deep familiarity with Horizon, and are often written in shorthand which assumes that knowledge (as is described at paragraph 66 of Mr. Parker's witness statement) . To infer financial impact from the KELs and Peaks, I have had to make inferences, sometimes with a degree of uncertainty. I have allowed for this uncertainty by expressing the results as upper limits on the financial impact of bug, and attempting to make these upper limits conservative (that is, generous to the claimants).

689 Having done this summation of the maximum possible impact of bugs in KELs, to find the maximum likely financial impact on all claimants' branches, it is necessary to scale the summed financial impact, to account for the following factors: (a) inefficiencies in the KEL creation process, where a Horizon bug existed but did not give rise to a KEL; (b) limitations of the sampling of KELs that I have been able to do in the time available; (c) the claimants' branches being a small proportion of the total PO branch estate. In calculating each of these scaling factors, I have attempted to be conservative, to reach a result which is most favourable to the claimants, and least subject to changes in assumptions.

690 In my opinion, this analysis gives the simplest and most direct route to estimate the total impact on the claimant's branch accounts of all bugs in Horizon, known and unknown..

691 I believe there is no simpler way to estimate the impact on claimants' accounts of all Horizon bugs, than to sum the maximum financial impact of all those I can find, and then to correct the resulting number for those I have not been able to find. This is an application of standard engineering methods.

692 As I shall describe below, this analysis is still at an interim stage, as I have not had time to examine as large a sample of KELs as I should like. As a consequence, as I shall describe below, the results are derived not just from one sampling of the KELs, but from three separate samples. Combining the results from these separate samples is not straightforward, and I have used only a simple and conservative way to combine them (conservative in the sense that it tends to favour the claimants) to derive the main result.

693 The conclusion of this analysis is that the total impact of all Horizon bugs on all claimant's accounts is probably not more than £25,000, with a high degree of confidence, compared to the total shortfall of approximately £18.7 million reported by the claimants.

694 Therefore, in my opinion, bugs in Horizon cannot account for more than about 0.15% of the claimants' shortfalls. This conclusion is robust against the limitations of my analysis to date, but I shall continue to improve its accuracy.

POL00029046

8.7.2 Reporting of Anomalies and the Creation of KELs

695 In section 7.5 of this report, when discussing the effect of robustness countermeasures on Horizon bugs which might affect branch accounts, I estimated the probabilities that SPMs would report the occurrences of bugs, based on evidence from the suspense account bug, and other considerations (part of the MID countermeasure - MID by SPMs). I will use those estimates here, but before doing so, will make them more conservative - adjusted to be more in favour of the claimants - so that the court may place more reliance on the results.

696 The estimates I made there - together with the more conservative estimates I shall use here for calculation - are as follows:

697 I assumed, as 'middle of the road' assumptions of SPM behaviour in reporting anomalies in their monthly balancing:

c) If a discrepancy is £1000 or more , the SPM probably needs to investigate it. If he cannot find the cause in his branch, the likelihood of his reporting it through a help line is 80% (for the computation which follows, I shall assume only 40% of SPMs report)

f) If a discrepancy is of the order of £300, 30% of SPMs will report it (for this computation, I shall assume only 15%)

g) If a discrepancy is of the order of £100, 10% of SPMs will report it (for this computation, I shall assume only 5%)

h) For a discrepancy of £10 or less, it is usually not worth the SPM's time to investigate it (because errors in counting cash or stock are often larger than this); so these are reported on less than 1% of occasions.

698 Thus the estimates used for calculation are twice as conservative as my central estimates, to favour the claimants.

699 Next consider a bug in Horizon, whose total net financial impact on SPMs, from all the occasions when it occurred, was £10,000 or more. For instance, this might have happened in 10 occurrences of £1000 (case(a)), or 33 occurrences of £300 (case (b)), or 100 occurrences of £100 (case(c)), or 1000 occurrences of £10 (case (d)). For the moment I exclude the last 'micro bug' possibility. I will address it later.

700 Had the bug been immediately evident to the SPM, (evident before his monthly balancing) I assume as in section 7.5 that it would have been reported on many occasions, regardless of the size of its financial impact.

701 If the bug was not immediately evident, but its effects were only evident to the SPM in monthly balancing, I shall use the conservative estimates above of the probability of the SPM reporting each occurrence. Using those estimates, this bug would have been reported by SPMs on 10*0.4 = 4 occasions in case (a), on approximately 33*0.15 = 4 occasions in case (b), or 100*0.05 = 5 occasions in case (c).

702 It is then clear that in a mixture of these cases - where the bug occurred with variable financial impact on each occasion - it would still have been reported about 4 times. Therefore the probability of it not being

POL00029046

reported at all is small - say less than 10%. It would have been reported at least once by an SPM, with probability 90%.

703 This is my best estimate 0.9 of the probability that some SPM will report any bug, if it has impact on their branch accounts. For the purposes of calculation, I shall use a more conservative estimate of 0.7. In my opinion, it strongly favours the claimants to assume a 30% chance that no SPM will report such a bug

704 Also in section 7.5, I estimated the probability that , when being informed of an anomaly which might lead to an error in branch accounts, Fujitsu would create a KEL. I estimated that these processes resulted in a KEL on more than 90% of the occasions where it was reported and there might be some effect on branch accounts.

705 Again, I shall use a more conservative estimate for the purposes of calculation. I shall assume only that Fujitsu created a KEL on 50% of the occasions where a reported anomaly might affect branch accounts. From evidence I have seen about Fujitsu's processes, this is a very conservative assumption.

706 This means that for any bug whose impact on all branch accounts was £10,000 or more, the probability of it leading to a KEL was 0.7*0.5 = 0.35. At least 35% of such bugs appeared in KELs. Any bug with smaller financial impact - right down to zero - had some probability of appearing in a KEL - especially if it was immediately visible to the SPM. At least 50% of those bugs would appear in KELs.

707 This means that if I search for KELs describing bugs with potential financial impact, amongst the 8300 KELs that have been provided to me and those that have been archived, I will find evidence of at least 35% of all bugs with financial impact £10,000 or more, and for 50% of any bugs which were immediately evident to the SPM, regardless of the size of their impact. If I search only the 8300 KELs which have not been archived, the figures should be reduced by a factor 0.85 to account for the archived KELs.

8.7.3 Analysis of KELs with Possible Financial Impact

708 As I mentioned above, because of limited time to prepare my report, my analysis of KELs is at present based on three separate samples, which I shall describe in turn. These samples are:

- A sample of 80 KELs selected from the 8390 KELs at random
- A sample of 50 KELs all containing the symbol '£' and therefore more likely to concern possible financial impact - which I have analysed and which subsequently Fujitsu have analysed
- The sample of 5111 KELs examined by Mr Coyne.

709 Because of the very different types of analysis applied to these three samples, it has not been possible to combine them in any straightforward numerical manner, and I am only able to combine the results in a conservative manner which favours the claimants.

Randomly Chosen KELs

710 For my first sample, I selected 80 KELs in a pseudo-random manner from the 8390 KELs (in practice I chose every 100th KEL from an alphabetically sorted list of KELs, so as to avoid any possible bias in my choice). My analysis of these KELs is given in Appendix D. There, the sample of 80 KELs is split into

POL00029046

two tables - a table of 30 KELs (where I have identified in the table the robustness countermeasures in evidence in the KEL) and a table of a further 50 KELs (where I have not identified the countermeasures in the table)

711 For the great majority of these KELs, it was immediately obvious either that they were not bugs in Horizon, or that they would have no effect on branch accounts.

712 For the remaining KELs, I used my knowledge of the countermeasures to assess which countermeasures were applied, and whether or not they would prevent any impact on branch accounts. As a result, I found no KELs with possible impact on branch accounts.

713 From such a limited sample (one KEL in every hundred) I cannot conclude that there were no KELs with any impact on branch accounts in the whole set of KELs.

714 However, if there were 200 such KELs in the whole set (i.e. one KEL in every 40), it is statistically very likely[90] (a chance of about 90%) that at least one of them would have got into my set of 80.

715 From, this analysis, I conclude that there are probably not more than 200 KELs which relate to bugs with possible impact on branch accounts.

KELs including the symbol '£'

716 For the next sample, I have used all those KELs whose text includes the symbol '£'. I chose this set because in my opinion, a KEL with some possible financial impact is more likely to contain the symbol '£', so looking at these KELs may be a faster way to find some KELs with impact of branch accounts. These KELs are easily found by a Windows search, and there are 259 of them. In the time available to me, I have been able to examine 50 of these 259 KELs. The results of this examination (which is tabulated in Appendix D) are:

- For 42 of the 50 KELs, either the KEL does not arise from a bug in Horizon; or if it does, there is no possible impact on branch accounts
- For 8 of the 50 KELs, there is a possible bug with possible impact on branch accounts. I found this by going some way in granting the benefit of the doubt in this respect to the claimants, and admitting the possibility of financial impact even if in my view it was remote.

717 So the sample has borne out my opinion that KELs which mention the symbol '£' are more likely to have financial impact, than KELs chosen at random.

718 Since I made this analysis, the same set of 50 KELs was passed to Fujitsu, and they analysed them. The results are appended to the witness statement of Mr Parker, which I received on 18th November.

719 Because Fujitsu have a deep knowledge of Horizon, of the usage of KELs, and the terminology used in KELs, if this evidence is accepted, in my opinion their analysis is likely to be more accurate than my own.

---

[90] The figure of 90% is derived as follows: if one KEL in every 40 has possible impact on branch accounts, then the chances of any single KEL not having any possible impact is (1 - 1/40) = 0.975. The chances of every KEL in my sample of 80 KELs having no possible impact is then 0.975 raised to the power 80. This is 0.131, which is approximately 1 - 0.9.

POL00029046

720 For the 42 KELs which in my initial opinion had no impact on branch accounts, Fujitsu's analysis agreed with my own.

721 Of the 8 bugs which in my opinion might have had impact on branch accounts, Fujitsu found that only 4 of them had that potential. This also agrees with my analysis, because I was giving the claimants the benefit of the doubt, and Fujitsu have more information to remove doubt.

722 Unfortunately, because I do not yet know by how much the inclusion of a '£' symbol would increase the chances of a KEL signifying a financial impact, I have no way of scaling up Fujitsu's result of 4 KELs with possible financial impact to the whole set of 8390 KELs. However, it does give 4 more KELs with possible financial impact, to help me to assess the extent of that impact.

KELs examined by Mr. Coyne

723 In paragraph 5.114 of his report, Mr. Coyne says: '*Regarding the extent of potential errors within Horizon I have analysed 5114 Horizon Known Error Logs (KELs) to determine the scope of potential bugs or 'PEAKs' (as they are referred to by Post Office and Fujitsu). Of these 5114, I have found that 163 contain PEAKs that could be of significant interest and of these 76 are referred to in the report*'

724 Mr Coyne does not define what he means by '*significant interest*', but it appears to relate to '*the scope of potential bugs* '.

725 Since Mr Coyne's report does not mention any robustness countermeasures, he has evidently not examined any of these KELs from the viewpoint of those countermeasures , to assess whether they could or could not have had any impact on branch accounts.

726 I have examined 62 KELs which I found cited in his report, and I found that fewer than 8 of them might have had financial impact. This analysis is contained in a table in Appendix D.

727 Fujitsu have examined approximately the same set of KELs, identified by PO's lawyers in Mr Coyne's report, and the result is annexed to Mr Parker's witness statement.

728 With their better knowledge of the significance of the KELs, Fujitsu found results similar to mine - apart from the known receipts/payments mismatch, there were very few with possible financial impact. I have not completed comparing Fujitsu's analysis with my own.

729 Furthermore, I believe I can infer that had any of the 5114 KELs examined by Mr Coyne had clearly and explicitly indicated a bug with significant financial impact on branches, Mr Coyne would have quoted from that KEL in his report.

730 On any interpretation, since about half of Mr Coyne's 163 KELs of 'significant interest' have been shown by Fujitsu and by me to have no financial impact, it can be inferred that his search of 5114 KELs has revealed no more than 100 KELs with potential financial impact. In my opinion this is an extremely conservative estimate.

Combining the results of the Three samples

POL00029046

731 My survey of 80 randomly selected KELs revealed none with financial impact. From this I inferred that in the 8390 KELs disclosed to the experts, probably no more than 200 had potential financial impact.

732 From the sample of 50 KELs mentioning the symbol '£', Fujitsu found no more than 4 with potential financial impact. It is not yet possible to scale up this result to the full set of KELs.

733 From Mr Coyne's sampling of 5114 KELs I inferred that no more than 100 have possible financial impact.

734 Taking these results together, and not wishing to rely too much on analyses other than my own, I infer that in the set of 8390 KELs, no more than 200 have potential financial impact. This is in my view a conservative estimate, to be used for calculation; my more central estimate is 100.

8.7.4 Mean Financial Impact of One Bug

735 From the previous analysis, I have found only 7 bugs with possible financial impact on branch accounts - the three known bugs cited by Mr. Coyne and analysed in my section 8.6, and the four which Fujitsu found to have potential financial impact. I summarise these in a table, to estimate the mean financial impact of any bug:

Bug or KEL	Commentary	Approximate Financial impact across all PO branches
Receipts/payments mismatch		£20,000
Suspense account		£14,000
Callendar Square		£3000
AChambers2252R		£3000
AChambers4134R		£200
ballantj020J		£300
AChambers253L		£500
TOTAL Impact		£41000

Table 8.3 - Estimated possible financial impacts of bugs or KELs

736 My estimates of the financial impact of the KELs which in Fujitsu's view might have financial impact are at present very approximate. For the resulting sum at the foot of the table, all that matters is that none of them are large compared to the impact of the three known bugs.

737 From this table, the mean financial impact of any single bug - across all branches in the PO Network - is approximately £41000/7 = £6000.

738 This is a conservative estimate of the actual financial impact of a bug on an SPM, since it is dominated by the two first rows (the known bugs), and in both those cases, evidence suggests that all the branches affected were compensated by PO. If I were to allow for the possibility that PO compensate branches whenever they are aware of a shortfall caused by a bug, the figure would be much less than £6000. My central estimate is therefore £2000.

8.7.5 Calculation of Financial Impact of All Bugs

739 The result of the analysis so far is:

- There are not more than 200 bugs with financial impact in all the KELs

POL00029046

- Of those, the mean financial impact per bug is not more than £6000

740 This makes a maximum financial impact of all bugs in KELs on all PO branches of 200*£6000 = £1.2 million

741 To find the impact of Horizon bugs in all KELs on all PO branch accounts, these results need to be scaled by the following factors:

- Scaled up by a factor 1/0.35 to allow for the fact that not all anomalies may lead to a KEL - either because they are not reported by the SPM (I assume they were only reported in 70% of cases), or because Fujitsu do not create a KEL (I assume they were created in only 50% of reported cases). Then 0.7 times 0.5 is 0.35. This builds in the conservative estimates above.
- Scaled up by a factor 1/0.85, to allow for archived KELs, which the experts have not seen
- Scaled down by a factor 160, as calculated in section 8.5 (the claimant scaling factor) , to allow for the fact that claimant branches were only a small proportion of the whole PO network

742 The result is an impact of £25000 across all claimants.

743 In applying these scaling factors, I have assumed that the probability of a Horizon bug striking a claimant's branch, in any given month, is the same as the probability of that bug striking any other branch. That assumption is justified in Appendix F.

744 The total impact of all Horizon bugs on claimants' branch accounts of £25000 is to be compared with the £18.7 million shortfalls that they have claimed. This figure is approximately 0.1% of their shortfalls.

745 This conclusion rests on a number of assumptions, which I have stated and justified when deriving it. In all cases, I have tried to make these assumptions conservative, erring in favour of the claimants, to make the result as reliable as possible.

8.7.6 Summary of the calculation

746 I have made the calculations above in an Excel spreadsheet, which is attached to my report. For convenience, the spreadsheet is summarised below.

POL00029046

Item	Label	Central Estimate	Conservative Estimate		Source
Mean number of branches in PO network, 1999 - 2018	A	13560	13560		Spreadsheet with AVDB WS
Years lifetime of Horizon	B	19	19		2000 to 2018 inclusive
Total branch months	C	3091680	3091680		C=A*B*12
Claimants branch size/typical branch size	D	0.37	0.37		Spreadsheet with AVDB WS
Scaling factor from all PO to one claimant branch month	E	8355892	8355892		E=C/D
total claimant shortfall (pounds)	F	18700000	18700000		claims
total claimant branch months	G	52000	52000		claims
scaling factor from all PO branches to all claimants	H	161	161		H=E/G
KELs with potential impact on branch accounts	L	100	200		finding from my inspection of KELs
Mean financial impact of KEL with potential impact (pounds)	M	2000	6000		finding from my inspection of KELs
Summed financial impact of KELs wth potential impact (pounds)	N	200000	1200000		N=L*M, or direct sum of KELs impact
Probability(SPMs report bug with impact|bug occurs)	T	0.9	0.7		Evidence on SPM reporting of anomalies - section 7.5
Probability(FJ create KEL|SPMs report bug)	U	0.9	0.5		Evidence on Fujitsu processes for KE creation - section 7.5
Probability(FJ create KEL|Bug occurs)	V	0.81	0.35		V=T*U
Probability(KEL is not archived)	W	0.85	0.85		Evidence on KELs archiving (Parker W
Probablity(KEL is created and not archived|bug occurs)	X	0.6885	0.2975		X=V*W
Summed financial impact of bugs on all PO branches, corrected for KEL sampling,creation and retention (pounds)	Y	290487	4033613		Y = N/X
Summed financial impact of bugs on claimant branches, corrected for KEL sampling,creation and retention (pounds)	Z	1808	25102		Z=Y/H
Financial impact of bugs on claimants, as a percentage of their losses	E1	0.010	0.134		E1 = 100*Z/F
Summed number of bugs, corrected for KEL sampling, creation, and retention	E2	145	672		E2=R/X

Table 8.4 - Estimates of financial impact of all known bugs on claimants branch accounts (as defined in row E1 of the table)

747 The calculations in the spreadsheet, as explained in its rightmost column, follow the descriptions I have given in this sub-section and sub-section 8.5.

748 The spreadsheet shows two alternative calculations, one from 'central' assumptions (which are my best estimate from the evidence) and one from conservative assumptions, intended to favour the claimants, and used to calculate my main result. (this is the right-hand of the two calculation columns, entitled 'Conservative Estimate').

749 It is evident that the conservative result (0.13%) is about ten times larger than the central estimate (0.01%). Yet the conservative result is still less than one percent of the claimants' shortfalls. Even with highly conservative assumptions, bugs in Horizon can account for much less than 1% of the claimed shortfalls. I do not rely on the central estimate.

750 Throughout the analysis of this sub-section, it has been assumed that bugs in Horizon predominantly cause shortfalls in branch accounts, rather than gains to branches. I note here that if a significant proportion of bugs in Horizon were to cause gains to branches, then the total number of bugs required to

POL00029046

produce a given level of shortfall in claimant's branches would be even larger. Of the upper limit which I have calculated on the number of bugs, some bugs would cause gains, and so would reduce yet further the part of the shortfalls experienced by the claimants, which could be accounted for by bugs.

751 It would be straightforward for Mr Coyne to add an extra column to this spreadsheet, to repeat the same calculation with the assumptions that he believes to be correct, and to calculate his version of the estimate E1.

8.8 Alternative Approaches to Estimate The Financial Impact of Bugs

8.8.1 Number of Bugs in Horizon Required to Substantiate the Claimants' Case

752 In section 8.5, 'Scaling of Financial Impact of Bugs' , I gave what I think is the simplest analysis of why, in quantitative terms, bugs in Horizon cannot have accounted for a large part of the claimant's shortfalls. I summarise it here.

753 Because PO have had an average of 13,560 branches over the lifetime of Horizon, the total number of monthly branch accounts has been about 3 million.

754 Therefore, if a bug like the Suspense Account bug has occurred 16 times in the lifetime of Horizon, the chance of it having occurred in any given branch in any given month is about 16 in 3 million. Because the claimants tended to have smaller branches than the average, doing fewer monthly transactions (by a factor 0.37), the chances of the bug occurring in a claimant's branch would be about 2 in 10 million.

755 I have considered a bug similar to the suspense account bug, which occurred about 10 times , and had a mean financial impact of about £1000 per occurrence. How many similar bugs would be needed, to give a one in ten chance of one such bug occurring, with an impact of £1000, on a particular claimant's branch in a particular month?

756 The answer, given by elementary arithmetic which I describe in section 8.5, is that there would need to be 50,000 of these distinct bugs. If there were fewer than 50,000 similar bugs, if any claimant were to assert that in a given month a shortfall of £1000 in his accounts was caused by bugs in Horizon, then the chances of his assertion being correct are less than one in ten.

757 So the claimants cannot credibly assert that their shortfalls were caused by bugs in Horizon, unless there were something of the order of 50,000 such bugs.

758 Only three such bugs have been found. My own search of KELs has found only 8 other possible bugs. Mr Coyne's examination of over 5000 KELs has found no other bugs which definitely caused shortfalls.

759 Thus the claimants' case requires 50,000 bugs in Horizon - but only a handful have been found by the experts. Neither expert can quantitatively support the claimants' case.

8.8.2 Considerations of Call Centre Workload

760 This section contains an alternative analysis of the financial impact of Horizon bugs on the claimants, which rests on some of the same evidence as that relied upon in the previous section, but uses a different approach to calculation.

POL00029046

761 I start from the fact that the sum of all shortfalls claimed by the claimants was approximately £18.7 million, over the period 2000-2018. I next suppose, following the hypothesis put forward by the claimants, that some large part of this amount (say 50%) was caused by bugs in Horizon. This gives a total claimants' shortfall, caused by bugs in Horizon, of approximately £10M during the period. (I assume this as the start of an argument of *reductio ad absurdum*, which follows.)

762 I next assume that bugs in Horizon would affect all branches - claimants and others - in approximately equal measure - as any bug in Horizon would affect all transactions across the PO network at random. (I shall examine this assumption in Appendix F).

763 If that assumption is correct, then the losses from bugs in Horizon suffered by all branches over the lifetime of Horizon, is the loss suffered by the claimants, multiplied by the claimant scaling factor. In section 8.5, I calculated this factor to be 160.

764 In that case, following the hypothesis put forward by the claimants, the impact of Horizon bugs on all PO branches over the period 2000-2018 would be expected to be £10M times 160 = £1,600 million.

765 I next consider the consequences this would have had, across the PO network.

766 In section 7.5, I estimated the probability of SPMs reporting anomalies in their accounts, depending on the size of the anomaly. In this section, I shall again use the same conservative form of these estimates as follows:

767 I estimated, as 'middle of the road' best assumptions of SPM behaviour in reporting anomalies in their monthly balancing:

a) If a discrepancy is £1000 or more , the SPM probably needs to investigate it. If he cannot find the cause in his branch, the likelihood of his reporting it through a help line is 80% (for this sub-section, I assume only 40% of SPMs report)

i) If a discrepancy is of the order of £300, 30% of SPMs will report it (for this sub-section, assume only 15%)

j) If a discrepancy is of the order of £100, 10% of SPMs will report it (for this sub-section, assume only 5%)

k) For a discrepancy of £10 or less, it is usually not worth the SPM's time to investigate it (because errors in counting cash or stock are often larger than this); so these are reported on less than 1% of occasions.

768 Anomalies which were immediately apparent to the SPM, in customer transactions, would be reported with higher frequency.

769 Suppose that the impact of one occurrence of a Horizon bug on a branch's accounts was approximately £1000. To cause a shortfall of £1,800M over an 18-year period, there would need to be 1,800,000 of these occurrences. Using the conservative estimate above, at least 40% of these, or 720,000, would be reported by the SPM.

POL00029046

770 If, on the other hand , the impact of one occurrence was £300, to cause a shortfall of £600M would require 2 million occurrences - of which at least 15%, or 300,000, would be reported (more, if they occurred in customer transactions). Similarly, if each occurrence was of £100, there would be 6 million occurrences, of which at least 2,400,000 would be reported.

771 So, if the hypothesis put forward by the claimants is correct, regardless of the size of impact of occurrences of bugs, the Horizon help desk would have been subjected to about 720,000 or more calls from SPMs, over 18 years. This is a rate of more than 200 calls per day - all raising urgent issues with large financial impact that needed to be resolved.

772 At paragraph 184, at Angel Van Den Bogerd's witness statement she says that call volumes into NBSC have been of the order of 1000 calls per day. It seems to me unlikely that (if her evidence is accepted) such a high proportion as 20% of these should have been about high-value discrepancies in branch accounts.

773 Those issues would have all have given rise to KELs - or to notes on existing KELs and their Peaks saying: "this issue has arisen yet again". The number of KELs, or recurrences of the same issue noted in a KEL or Peak, would have been of the order of 720,000 (rather than the actual 8,000 KELs, with a small number of Peaks per KEL).

774 Also (following the hypothesis put forward by the claimants), most of these issues would need to have been resolved incorrectly - by attributing them to human error in the branch, rather than to a bug in Horizon - in order to lead to a loss in the branch. Investigation of the bug would not have led to any correction in favour of the branch.

775 There are several pieces of evidence that in my opinion are not consistent with the account put forward by the claimants:

- As above, the Horizon help desk was probably not bombarded by 200 or more urgent calls, reporting large anomalies, each day
- I have seen no evidence in KELs or Peaks or reports of 720,000 anomalies in branch accounts of £1000 or more, or of larger numbers of smaller anomalies. The number of KELs relating to any anomaly at all in branch accounts appears to be less than about 100 KELs.
- I have seen no evidence that most anomalies were resolved incorrectly. Rather, because of the many checks and countermeasures in Horizon, it appears that the KELs and Peaks record that most anomalies were diagnosed fairly quickly and corrected.
- I have seen no evidence that the number of anomalies diagnosed as error in the branch, and therefore attributed to the branch, was anything like the figure of 720,000 or more which is implied by the claimant's hypothesis.

776 In my opinion the hypothesis put forward by the claimants, that some large part of their £18.7M shortfalls was caused by bugs in Horizon, is not consistent with this evidence, by a large margin.

POL00029046

8.8.3 Considerations of Central Accounts

777 This formulation of the analysis of the impact of bugs in Horizon starts from the same premises as the second analysis. Following that analysis, it assumes that:

- Of the total £18.7 shortfalls claimed by the claimants, as the claimants say, some large part (for definiteness, I take it to be 50%) was caused by bugs in Horizon - giving £10M caused by Horizon bugs.
- Because the impact of Horizon bugs is expected on average to be the same for claimants as for other SPMs, the impact of bugs on all SPMs is expected to be £10M times a claimant scaling factor, which I have calculated in section 8.5 to be 160.

778 This would imply that over the lifetime of Horizon, a sum of the order of £1.6 billion has leaked out of branch accounts, caused by bugs in Horizon.

779 PO accounts are held on POLSAP, which adheres to the principles of double entry accounting. This means that any amount of money which leaks out of branch accounts must appear in some other account (unless PO were concealing it, by some mechanism which is outside my expertise). Setting aside the possibility that money leaks from one branch to another, in my opinion this would be some central PO account or accounts, which aggregates the amounts from many branches or all branches.

780 This would imply that, in some central PO account or accounts, a figure of £1.6 billion over 18 years (or approximately £8 million per month) is appearing - apparently for no good reason, because it is caused only by bugs in Horizon, which PO do not know about.

781 In spite of the large amount of money which passes through PO accounts - which is of the order of £100 billion per annum - the great majority of this money is pass-through of agency business, which PO does for its clients. Those clients check the amounts paid to them, and it does not seem possible to me that £8M per month could be hidden in those figures. That amount is greater than the total amount of transaction corrections, which (as I describe in section 9) have amounted to something of the order of £2M per month.

782 Therefore the £8M per month which, on the claimant's hypothesis, arises from Horizon bugs, would be impacting some smaller central accounting lines, possibly connected with cash and stock in the branches.

783 In my opinion it does not appear likely that either:

- These smaller accounting lines, (which are where PO makes a large part of its profit) are managed so loosely that an unexplained figure of £8M per month can hide in them
- In the PO's annual financial audit, done by an independent third party, an unexplained figure of £93M could pass without comment.

784 In my opinion, this combination of the DEA and MID countermeasures would have detected the sums arising from Horizon bugs, if they had been anything nearly as large as the claimants assert.

POL00029046

8.9 Impact of Bugs in Horizon on Individual Claimants

785 While the previous analysis has calculated the likely size of the summed impact of Horizon bugs on all claimants, I also need to calculate its likely impact on any individual claimant. The previous analysis implies that the impact on all claimants, spread over all their 52,000 months of tenure, was at most of the order of £20,000. In order to be conservative when considering individual claimants, I shall multiply this figure by a factor 5 - assuming (to give the claimants the benefit of the doubt) that my analysis has somehow omitted 80%, or 4/5, of the bugs with financial impact - or underestimated their financial impact by a factor 5. This is a very conservative assumption.

786 The mean financial impact of bugs in Horizon on any one claimant in any one month of his tenure, is then £100,000/52,000 = £2. A mean loss per month of £2 from Horizon bugs can occur though bugs with higher financial impact, but only if they occur with low probability. For instance, a loss of £200 could occur, but only with probability one part in 100. Or a loss of £2000 could occur, with probability one part in 1,000.

787 This is because, if a loss of £200 occurs in some month, yet the mean loss over all months is £2, there must be 99 other months with zero loss, to make the mean over all months equal £2 (= £200/(99+1)). Thus the loss would be non-zero only for one month in a hundred months - a probability one part in a hundred. By the same analysis, the larger the financial impact of one occurrence, the less likely it is to occur.

788 The mean loss from bugs of £2 per month is to be compared with the mean loss per month claimed by the claimants, of £360. Typically claimants claimed larger losses in individual months. If, then, a claimant were to say: "In a particular month, I suffered a loss of £2000, and I assert that it was caused by a bug in Horizon", then (as above) the probability of that assertion being correct is only one part in 1,000.

789 If the claimant made the same claim about £2000 losses in two different months[91], the chances of that account being correct are one part in 1,000,000; and so on - the more months are involved, the more improbable the assertion becomes.

790 Similarly, a claim that Horizon caused two losses in one month, each of £200, has only a probability $(1/100)^2$, or one part in 10,000 , of being correct. (again, because the probabilities of independent events are to be multiplied)

791 This is standard probability theory, applied to calculate the balance of probabilities of bugs impacting individual claimants.

792 This result concurs with a previous analysis, where I showed that , in order for a claimant to credibly assert that a shortfall in his accounts in some month were caused by bugs in Horizon, there would need to be 50,000 such bugs. No such number of bugs has been found. Only a handful have been found.

---

[91] This is because the two events (two occurrences of a bug) are statistically independent, so their probabilities multiply. This is an application of standard risk analysis.

POL00029046

793 Because so few bugs have been found, the overwhelming probability is that any set of branch accounts in any month is accurate - with no significant shortfalls caused by bugs in Horizon. There may, however, be temporary inaccuracies in branch accounts caused by delayed TCs, which lead to later corrections.

8.10 Financial Impact of All Bugs, Using Data Provided by the Claimants

8.10.1 Analysis In This Sub-Section

794 In this sub-section I shall:

i. Explain in qualitative terms why the data on shortfalls submitted by the claimants, as part of their claim, is not consistent with the assertion that their shortfalls arose from bugs in Horizon

ii. Show some charts derived directly from evidence submitted by the claimants, which is of interest in the light of their claims, and provides some support of point (i)

iii. Describe quantitative upper limits on the part of the shortfalls that can have arisen from bugs in Horizon.

795 The quantitative derivation of the limits (iii) is described in Appendix E .

8.10.2 Qualitative Analysis

796 Claimants held branches for a total of just over 52,000 months. If their shortfalls had been caused largely by bugs in Horizon, then one would expect those bugs to have occurred randomly, and approximately uniformly across those 52,000 months.

797 I illustrate this by an analogy. The total claim is like a field, divided into 52,000 'plots' (monthly branch accounts) of approximately equal area. Bugs in Horizon are like raindrops, falling randomly and uniformly across the field. One would expect approximately the same number of raindrops to fall on each plot (each set of monthly branch accounts), apart from random fluctuations.

798 The picture revealed by the claim is very different from this, and is not at all uniform. There is a large amount of 'clumping' of the claimed shortfalls, in two respects.

- There is clumping across claimants. Some claimants claim very small monthly average shortfalls - of £50 or less; while other claimants had very large average monthly shortfalls, of £1000 or more.
- There is clumping in time. At all time periods, according to their claims, many claimants experienced periods of 36 months or more with no shortfalls. Shortfalls arrived in lumps.

799 In both of these respects , the evidence submitted by the claimants is inconsistent with their claim, that bugs in Horizon were the origin of their shortfalls. It is not like uniform rainfall; it consists of localised deluges. That is consistent with the causation of shortfalls by human errors, which depend on variable factors such as the management of individual branches. This is illustrated with data in the rest of this sub-section.

8.10.3 Evidence used for Analysis, and Graphical Summaries

800 There are 561 claimants. Each one has provided a claim summary, which includes:

- Section 8.1: the total amount of shortfalls they have repaid to PO

POL00029046

- Section 3.1: a listing of all the individual shortfalls they claim to have experienced, with a date or date range for each shortfall

801 For a few claimants, the amount claimed in section 8.1 exceeds the sum of the individual shortfalls in section 3.1. For those cases, I have assumed that some shortfalls are missing from the section 3.1 data. To correct for these as far as possible, for each of those claimants I have added a single 'balancing shortfall' to the section 3.1 data, whose amount is chosen to make the sum of section 3.1 figures equal to the section 8.1 figure, and whose date range is the whole period of tenure of the claimant. This wide date range makes no assumption about when the missing shortfalls arose. From then on, I have used exclusively the section 3.1 figures to assess the shortfalls and the periods during which they occurred.

802 This leads to the following overall figures:

- 561 claimants
- Sum of the periods of tenure of all claimants: 52,077 months
- Mean period of tenure per claimant: 92 months
- Sum of all claimed shortfalls: £18.7 million
- Mean claimed shortfall per claimant per month in tenure: £359

803 A spreadsheet, giving for each claimant the dates of their tenure and the dates and amounts of all the shortfalls they experienced as in section 3.2 of their claims, has been provided to me by the defendant's solicitors, and is the basis of the analyses in this section. It is attached as an Annex to this report.

804 There are some results which can be derived and shown graphically by simple processing of the spreadsheet provided to me, using the facilities of Microsoft Excel. I present these results here to help understanding of the claim.

805 There is a wide range of variation in the average loss per month experienced by each claimant. Some claimants experienced an average loss per month of several thousand pounds, whereas 72 of the 560 claimants each experienced an average monthly loss of £50 or less.

806 This can be seen in the cumulative histogram below. The horizontal axis is monthly average loss of a claimant in pounds. The vertical axis is the number of claimants (out of the 561) whose average monthly loss is less than that figure. Claimants have been sorted in order of ascending average monthly loss - those with the smallest average monthly loss on the left. The diagram does not include the claimants with the highest monthly losses (who would have been to the right-hand end), because those losses (which were several thousand pounds per month) would have compressed the vertical scale so much that one could not see the monthly losses for the claimants with smallest losses. This histogram comes from an earlier analysis, when I only had available the section 8.1 shortfalls repaid, rather than the full shortfalls experienced as in the section 3.1. The qualitative result is unaltered by this difference.

POL00029046



Figure 8.1 - Numbers of claimants with average claimed losses per month less than the values shown

807 This graph shows that claimants are claiming a very wide range of average monthly losses - from a few pounds per month for some claimants, up to £1000 per month for others (and more claimants missed out at the right-hand end of the graph, which if included would continue to shoot upwards, as explained above).

808 This graph on its own calls into question the idea that most of the claimant's claimed losses were caused by bugs in Horizon - because one would expect bugs in Horizon to have affected all claimants equally, apart from random fluctuations. This would have led to all claimants suffering approximately equal losses per month - not to a 'low tail' of claimants with very small losses per month, or a 'high tail' of claimants with very high losses per month. Since the graph shows both a low tail and a high tail, it contradicts the hypothesis of random Horizon bugs impacting all claimants. It is, however, consistent with the idea of losses being mainly caused by human error - with a wide range in the rates of human error in different branches.

809 It appears that the claimants with shortest tenures are claiming the highest average monthly rate of loss:

POL00029046

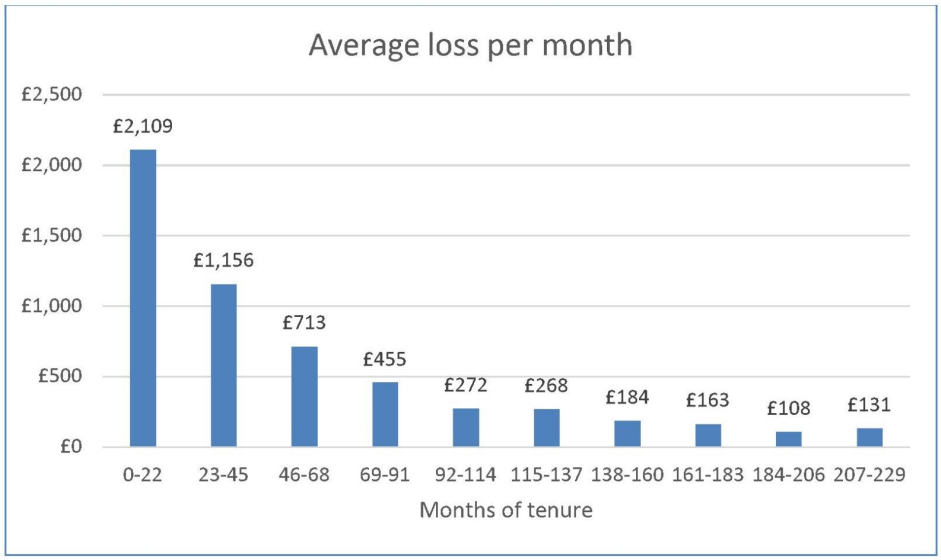


Figure 8.2 - Average losses per month by period of tenure

810 To describe two results from this chart:

- Claimants with tenure between 0 and 22 months claimed to experience an average shortfall of £2,109 per month
- Claimants with tenure between 138 and 160 months claimed an average shortfall of £184 per month

811 Within each range of tenure, there was wide variation in the rate of loss - with many claimants claiming losses much less than £100 per month.

812 This chart is equally not consistent with a hypothesis that losses arose from bugs in Horizon. On that hypothesis, the mean loss per month would not vary with length of tenure, as it does in the chart.

813 One possible interpretation of the chart is that claimants with shorter tenures were less experienced, and so were more prone to make human errors which caused losses.

814 I have also plotted the number of claimants who were claiming losses in each year:

POL00029046

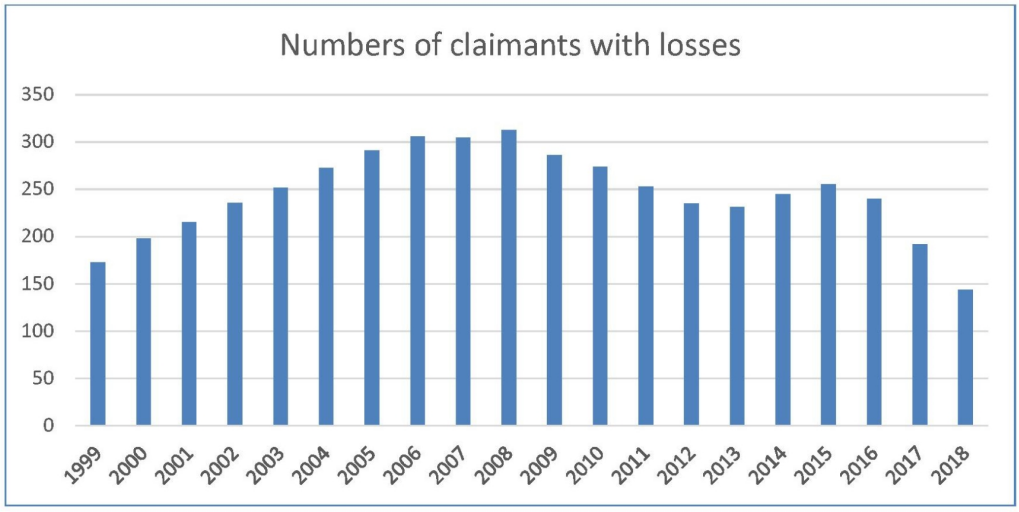


Figure 8.3 - Loss claimants by year

This shows that with minor variations, claimants have claimed losses over the whole period 1999 - 2018. I can plot the amount of losses they have claimed:

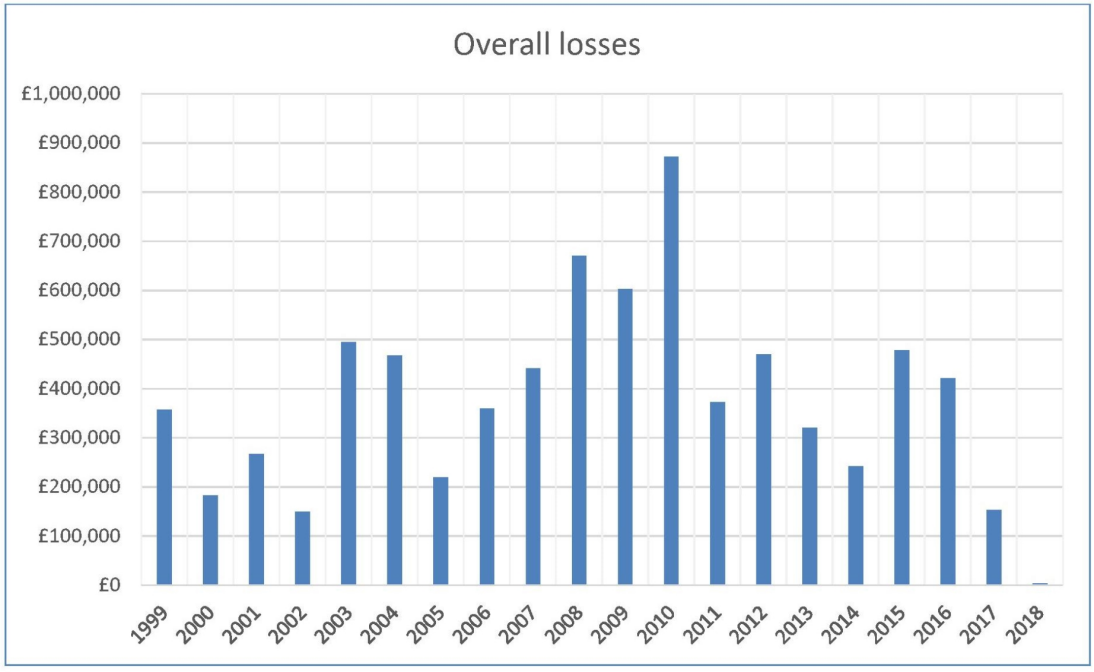


Figure 8.4 – Loss amounts by year

I do not yet know the causes of variation in particular years, but it is clear that shortfalls are claimed to have been experienced from both Horizon and Horizon Online, in all years of their operation. Much of the variation may just arise from random fluctuations.

815 However, the broadly flat nature of this graph, with random-looking fluctuations for year to year, qualitatively contradicts the notion, as put forward by Mr Coyne, that Horizon sometimes had 'bad periods' in which robustness countermeasures did not work well, and claimants suffered large losses as a

POL00029046

consequence. In my opinion, any such 'bad period' would extend over two or three years, while Fujitsu grappled with widespread problems. The graph does not show this pattern.

816 One possible interpretation of Mr Roll's witness statement[92] is that Horizon suffered teething problems in its early years, leading to higher levels of losses to claimants caused by Horizon bugs in those years. As can be seen, the chart above is not consistent with that interpretation. The largest annual losses are claimed for later years.

817 The graph also contradicts the notion that old Horizon (pre-2010) was notably worse than Horizon Online (post-2010).

818 There is an obvious spike in claimants' reported losses in 2010, which one might interpret as arising from the introduction of Horizon Online, and teething problems in the new system. In Angela Van Den Bogerd's witness statement at paragraph 183, she says that there was a mandatory cash check in all branches before the change to Horizon Online, which may have caused a temporary spike in declared losses. If this is correct, it might account for the spike in 2010. Since many claimants show a pattern of not reporting losses for extended periods, followed by large 'lumps' of loss, this second account appears more likely.

8.10.4 Quantitative Analyses

819 In Appendix F, I derive two quantitative results from the evidence described and charted above. I shall only state these results here; see the appendix for the numerical derivations of the results.

820 If all the claimant's claimed shortfalls arose from bugs in Horizon, or even if large part of them did, one would not expect to see a 'low tail' of many claimants with small monthly shortfalls (as in the chart above), much less than the average shortfall of £359 per month, as claimed by all the claimants.

821 This has enabled me to derive an upper limit on the proportion of the claimants' shortfalls which arises from bugs in Horizon. That upper limit is currently approximately 8% - although it needs to be further corrected for the variability in the sizes of claimant's branches. The derivation is in the appendix.

822 This derivation still needs to be further corrected for variations in branch sizes. I have not yet had time to do this, because I only received data on branch sizes on November 19th, when I was finalising this report. I do not expect the correction to be major.

823 While the limit of 8% needs further correction and is a much weaker limit than the limit of 0.1% on the same quantity derived in section 8.7, it is consistent with that limit, and has the merit of being based on entirely different evidence - evidence submitted by the claimants.

824 I next looked at the time dependence of the claimed shortfalls, by dividing the whole time period into three-year time slices. I found that for any three-year slice, a high proportion of those claimants whose period of tenure spanned the whole slice experienced no shortfalls at all during the slice. The details are tabulated in the appendix.

---

[92] 11 July 2016

POL00029046

825 This again is not consistent with the hypothesis that a large part of the shortfalls was caused by bugs in Horizon. Had this been so, bugs would have affected each claimant uniformly over each 36-month slice of his tenure. The chances of any claimant avoiding any bug for 36 months in a row would be extremely small.

826 This analysis can also be used to derive an upper limit on the proportion of the claimants' shortfalls caused by bugs in Horizon. The derivation is in the appendix. This limit is approximately the same number as the first limit - 8%.

827 The second of the two limits depends on the assumption that claimants reported their losses in a timely manner, shortly after the month when each shortfall first arose. It assumes that if a shortfall arose for any reason (including a bug in Horizon) claimants did not delay for many months before reporting it.

8.11 Extent of Bugs - the Number of Different Bugs

828 In section 8.4, I described what measures of 'extent' I would use in addressing Horizon issue 1 quantitatively. I described a first sense of 'extent' - financial impact - and a second sense - number of distinct bugs.

829 I expressed the view that the first sense would be more useful, in that it relates more directly to the claim of financial losses.

830 I also noted some limitations of the second sense of extent: that it is more difficult to infer from the available evidence (because robustness countermeasures are directed to control the first extent, not the second); that it suffers from ambiguity of definition; and that it is difficult to scale it between the three different scopes defined in section 8.4.

831 In spite of these difficulties, I state here my opinions on the second sense of the word 'extent' in Horizon Issue 1.

832 Using the same conservative assumptions as I used for the financial impact of bugs, I have estimated the number of distinct bugs, with impact on branch account in Horizon.

833 In the table in section 8.7, where I summarised these calculations, I estimated at the row marked 'L' that there were no more than 200 bugs in the KELs which might affect branch accounts.

834 To find the total number of such bugs in Horizon, this needs to be corrected for the following factors:

- The probability that no SPM ever reported the bug - estimated conservatively at 0.7
- The probability that Fujitsu never made a KEL - estimated conservatively at 0.5
- The probability that the KEL has not been archived - estimated at 0.85

835 The result of applying these correction factors is that the upper limit of 200 bugs in the KELs becomes an upper limit of 672 bugs in Horizon. This calculation is also included in the spreadsheet of calculations attached to this report, and shown in section 8.7.

836 This count excludes micro-bugs, whose impact on each occurrence, or in each month for one branch, is so small that they may never be reported. They are discussed in Appendix F.

POL00029046

837 Apart from micro-bugs, the estimate is very much an upper limit - where I have counted a KEL as potentially indicating a bug, even when the evidence in the KEL is far from conclusive.

838 In case 672 possible bugs might seem to be a large upper limit, I note again a conclusion I reached in section 8.5. In order to support an assertion, by any claimant, that some shortfall of £1000 in his accounts in some month was caused by bugs in Horizon, there would need to be not just 672 bugs in Horizon - but more than 50,000 bugs, each one with impact similar to the Suspense Account bug. My estimate of a maximum of 672 bugs is very small compared to 50,000, and so in my opinion does little to support the claimants' case.

8.12 Processing and Recording of Transactions

800 Part (b) of Horizon Issue 1 asks whether bugs in Horizon could 'undermine the reliability of Horizon accurately to process and to record transactions'.

801 In sections 4.4 and 6.2.6, I described the Horizon Core Audit system, which in my opinion is a highly reliable record of transactions entered in the branch- an example of the robustness countermeasure SEK. There is a highly reliable and secure chain of communication from the branch to the audit store. This applies to both Old Horizon and to Horizon Online. Any bug in Horizon which is outside this secure chain of communication (for instance, in a back end system such as TPS or DRS) cannot directly create errors in branch accounts. It can only do so indirectly - for instance, by leading to erroneous TCs which are accepted by the SPM, or other errors by the SPM. The great majority of software in Horizon is outside this secure chain.

802 I have seen no evidence for bugs which affect branch accounts, other than the three known bugs and a small number of KELs which might possibly indicate bugs with effect on branch accounts. Therefore in my opinion, the extent to which bugs in Horizon might undermine the reliability of Horizon accurately to process and to record transactions is very small indeed.

8.13 Analyses needed in Support of My Opinions

839 In this section I state the results of two analyses of topics which I need in support of opinions stated earlier in this section.

840 Each result is derived in Appendix F. I shall only state the results here.

8.13.1 Bugs Especially Affecting Claimants

841 In the analyses, I have made of the maximum possible financial impact of bugs, I have assumed that Horizon bugs occur at random, with equal frequency per month (or more precisely, per customer transaction) in the branches of claimants and non-claimants.

842 This is a reasonable assumption, given the nature of bugs in Horizon - that each bug is triggered by some combination of circumstances - say, a particular action, in a particular type of customer transaction - and that, in the absence of further information, that combination of circumstances is equally likely to happen in claimants' branches, as in non-claimants' branches.

POL00029046

843 The opposite assumption - that claimants' branches are somehow different from non-claimants' branches, in a way that triggers more bugs - is not a part of the claimants' case. If it were to become a part, it would be a testable part. In other words, if it were claimed that bugs in Horizon were triggered by some circumstance - which occurred more often in claimants' branches than non-claimants' branches, then that assertion could be tested. It would be possible to compare claimants' branches with non-claimant's branches in this respect, and see if there was any difference.

844 I have tried to think of possible differences of this sort, and I have only been able to find one candidate difference.

845 It might be said that claimants tend to make more errors than non-claimants, and that these human errors particularly trigger bugs in Horizon. Or it might be said that certain bugs in Horizon are successfully handled by non-claimants, but tend to cause claimants to make errors, which cause losses.

846 In the appendix I have examined both these possibilities, and found that neither of them can account quantitatively for a large excess of bug impact, for claimants over non-claimants. In engineering terms, they are both second-order effects (combinations of two unlikely things), and so cannot be large effects. For the derivation of this result, please see Appendix F.

8.13.2 Micro-Bugs

847 The analyses of financial impact of possible bugs in Horizon, which I have given in previous sub-sections, address bugs with all sizes of financial impact, except for a category which I have called 'micro bugs'. These are bugs in Horizon which:

- Have the potential to introduce errors in branch accounts
- Are not immediately visible to the clerk in a customer transaction (if they were, some of them would be reported, however small their financial impact)
- When they introduce discrepancies in the process of monthly balancing, introduce only discrepancies less than £10 per occurrence, which the SPM may put down to human error, or otherwise ignore.

848 This is my definition of a micro-bug.

849 Because of these properties, micro-bugs are less likely to lead to KELs or to investigation by Fujitsu. The question therefore arises: could they have significant effect on branch accounts? In Appendix F, I address that question.

850 There I show that the impact of micro-bugs on claimants' accounts must be much smaller than the impact of known bugs, which are recorded in KELs, and whose impact has been analysed in section 8.7.

8.14 Mr Coyne's Opinions

851 Mr Coyne addresses Horizon issue 1 in his summary of opinions, in paragraphs 3.1 - 3.3 of his report, and in section 5, up to paragraph 5.81.

POL00029046

852 Paragraph 3.1 quotes from the expert joint statement. Paragraph 3.2 makes the uncontroversial statement that bugs in Horizon could have existed for variable periods of time. Paragraph 3.3 talks about the 'sheer volume' of KELs - implying it was large, but not supporting this with the results of any analysis.

853 These summary paragraphs all address bugs in Horizon in general, and do not focus down on bugs which cause discrepancies in branch accounts, as Horizon Issue 1 requires. It is agreed between the parties that bugs in Horizon exist, as Mr Coyne's paras 3.1- 3.3 state.

854 These paragraphs do not address the extent of such bugs, as required by Horizon issue 1, except in the phrase 'sheer volume' of KELs.

855 It seems to me that Mr Coyne's opinions on Horizon Issues 1 cannot yet be contrasted with my opinions in two respects - because in those respects, he has not yet expressed an opinion:

a) He has not expressed an opinion on the many specific robustness countermeasures built into Horizon, and how effectively or otherwise they have acted to prevent discrepancies or shortfalls in claimants' branch accounts

a) He has not expressed an opinion on the quantitative financial impact of bugs in Horizon on claimants' accounts.

856 Regarding point (b), the arithmetic I have used in the calculation of the maximum possible financial impact of bugs is a straightforward application of IT risk analysis. It is contained in a small Excel spreadsheet attached to my report.

857 Mr Coyne may wish to re-calculate this result, based on his own sampling of KELs, and other evidence. Although I have every confidence in my result, it is based on the limited sampling and analysis of KELs which I have been able to make so far. It is an interim result. With time and further effort, it can be improved, and the remaining uncertainties in it can be reduced.

858 I next comment further on Mr Coyne's detailed opinions on Horizon Issue 1, expressed in his paras 5.1 to 5.81.

859 Paragraph 5.1 sets out some areas where bugs may occur - without addressing whether or not those bugs can cause discrepancies in branch accounts. Similarly paragraph 5.3 discusses common failure points in Horizon. It does not address what kinds of failures they might be, or whether they might affect branch accounts, in the light of the many robustness countermeasures in Horizon.

860 The linkage between bugs and branch accounts is addressed in paragraph 5.2 (which recites a point agreed in the expert joint statement) and paragraphs 5.4 to 5.14 (which address the three known bugs admitted by PO).

861 Paragraphs 5.15 to 5.30 discuss a number of issues, without , in my opinion, providing the depth of analysis to enable the reader to determine whether or not these issues affected branch accounts - or if so, the extent to which they did so.

862 For instance, paras 5.20 - 5.26 discuss various issues relating to cash management and pouch delivery. They do not acknowledge that problems managing cash are quite distinct from inaccuracies in branch

POL00029046

accounts, and often do not lead to inaccuracies in accounts. This is because of the countermeasure UEC. If a discrepancy arises during a Trading Period (TP) between cash as recorded on Horizon and physical cash (for instance, by mis-recording the amount of cash remmed in or out), then at the end of the TP, physical cash is counted, and any error in Horizon cash is corrected. This means that the issues cited by Mr Coyne probably lead to no error in branch accounts - and he does not provide any analysis to show that they might do so.

863 Similarly, Mr Coyne's examples of reference data errors (paras 5.30 - 5.34) show that errors occurred - not that they led to inaccuracies in branch accounts. There were many countermeasures such as TIN and UEC to ensure that they would not. Mr Coyne does not discuss these - and his examples do not illustrate bugs which affected branch accounts.

864 The same lack of any analysis showing any impact on branch accounts is shown Mr Coyne's other examples up to para 5.81.

865 In Appendix A to his report, Mr Coyne lists 9 Peaks which he says 'could have a financial impact on branches'. He quotes at some length from the Peaks, without, however, providing his own analysis of why those Peaks could have financial impact.

866 In Appendix H to my report, I provide my analysis of the same Peaks - which shows that in my opinion, none of them (apart from the known receipts/payments mismatch bug) have any significant impact on branch accounts.

867 In the time available since receiving Mr Coyne's report, it has not been possible for me to analyse all the examples he cites to the depth I should like, to assess whether or not they have any potential to affect branch accounts. This would involve me providing a depth of analysis that he does not provide in each case. I have made a preliminary analysis of the KELs he cites; out of 62 KELs I have examined, there appears to be potential to affect branch accounts in only 8 cases. This is only potential and does not establish that branch accounts were affected. As described in Mr Parker's witness statement, Fujitsu have analysed the same KELs, reaching conclusions similar to my own. Fujitsu find fewer KELs which might affect branch accounts - apart from temporary impacts. I do not rely on Fujitsu's analysis, but I note that they have a better understanding of the meaning of KELs than I do.

868 I have not had time to make any similar analysis of the other documents Mr Coyne cites, to assess their context and significance. I will do so in my supplemental report, where I shall also describe any further analysis I have made of the KELs he cites.

869 My finding of 8 KELs with possible impact on accounts, among 62 cited by Mr Coyne, (or Fujitsu's evidence that there were fewer than 8 such KELs) does not alter my opinions about the aggregate financial impact of bugs, as described earlier in this section. This is because I do not know the sampling criteria used by Mr Coyne in selecting those KELs - so I cannot adjust the KELs found by Fujitsu or myself for those criteria.

POL00029046

870 However, I note that Mr Coyne's survey of 5114 KELs, as described in his paragraph 5.114, reinforces my conclusion that the financial impact of bugs was very small. This is because if any of those 5114 KELs had stated in obvious terms that there was a bug with impact on branch accounts, I assume that Mr. Coyne would have quoted it in his report. There are no such direct quotations.

871 Thus, Mr Coyne's opinions of Horizon issue 1 lack any focus on the impact of bugs on branch accounts. They add little to the expert joint statement and the three bugs acknowledged by PO. Mr Coyne has said nothing quantitative about the extent of such bugs (as asked in Issue 1), which might be compared with the claimant's claimed shortfalls.

POL00029046

9. HORIZON ISSUES – RECONCILIATION AND TRANSACTION CORRECTIONS

9.1 The Issues

872 This section addresses Horizon Issues 5 and 15, which concern reconciliation and transaction corrections.

873 **Issue 5**: How, if at all, does the Horizon system itself compare transaction data recorded by Horizon against transaction data from sources outside of Horizon?

874 **Issue 15**: How did Horizon process and/or record Transaction Corrections?

9.2 Summary of My Opinions

875 Issues 5 and 15 are, on the face of it, factual issues, which can be addressed by factual evidence.

876 I repeat my comments in section 1 about my role as an expert and my approach to contested factual matters and evidence.

877 These questions do not invite an opinion on the quality, adequacy, sufficiency or other similar judgment on these processes. In light of this, I provide below my opinion on the evidence I have seen to address the factual questions of whether and, if so, how Horizon undertakes certain activities.

878 I also note that these questions are limited to activity regarding Horizon, and do not extend to other manual business processes operated by Post Office. Save for providing useful context on these other areas, my opinion is limited accordingly.

879 Mr Coyne has gone further than the above scope. He has offered opinions on the adequacy of the reconciliation process in a wider sense - in particular, raising the question of errors in transaction corrections. For the sake of balance, in section 9.6 I offer my own commentary on these matters without prejudice to my understanding of the scope of Horizon Issues 5 and 15.

880 My analysis of the evidence is that:

- For most of PO's clients (for whom PO branches carries out agency business) there is a regular automated process of comparing (reconciling) the transactions as recorded by PO, with the transactions as recorded by the client organisation.
- These comparisons may or may not be carried out within Horizon 'itself' ; but in any event, because of the large volume of transactions, the comparison has to be automated
- Whenever the comparison reveals any discrepancy, there appears to be a human process of deciding where to allocate responsibility for the discrepancy. This has to be a human process, and is therefore subject to errors.
- If responsibility is allocated to a branch, it results in a Transaction Correction (TC), which the branch may accept or query before it enters the branch accounts
- There is also reconciliation of cash remmed from branches to PO cash management, or in the reverse direction

POL00029046

881 The thrust of Mr Coyne's opinions on these issues - for instance in his summary paragraphs 3.13 and 3.28 - is to emphasise that reconciliation, and the creation of transaction corrections, are error-prone processes.

882 The significance of this for the claimants' case appears to be that any such errors might have introduced shortfalls in the claimant's branch accounts.

883 Because of this emphasis by Mr Coyne on errors in TCs, I need to address the topic of errors in TCs, and will do so quantitatively in section 9.6. I have calculated the likely extent of shortfalls in claimants' accounts arising from erroneous TCs, using evidence on:

- annual volumes of TCs (numbers and monetary amounts)
- the distribution of types of TC, in a typical year
- proportions of TCs disputed, and the proportion of disputes upheld
- the number and sizes of branches, both for claimants and other PO branches

887 The result is that an upper limit on the magnitude of the mean discrepancy which might have been introduced by erroneous TCs into any claimant's branch accounts in any month, is of the order of £2. This is to be compared to the mean shortfalls of £360 per month claimed by the claimants.

888 The probability of some larger discrepancy having been introduced in any given month is then very small, For instance, the chances of a discrepancy of £1000 would be one in 500 (500 = £1000/£2). This is explained below.

9.3 Reconciliation, Transaction Corrections and Transaction Adjustments

889 The processes for reconciliation, transaction corrections and transaction adjustments were described in section 6.3 of this report.

890 There I described how, because of the huge volume of transactions carried out every day, the first part of the reconciliation process - the comparison of the PO version of every transaction with the client's version of the same transaction[93] - has to be automated[94]. In my opinion, to do it manually would be a prohibitive workload.

891 I also described that when this automated process discovers a discrepancy, there needs to be an investigation of how the discrepancy arose, and in my opinion that process necessarily has to be a manual process.

892 When this manual process results in a TC[95], that TC is presented to the SPM, and he then has a choice of what to do with it[96]. The options[97] available to the SPM are listed in paragraph 6.55 of Mr Coyne's report.

---

[93] TD/ARC/014

[94] *Reconciliation and Incident Management Joint Working Document*, 18 March 2013 [POL-0032909]

[95] DELLDO014_2.doc, *TPS Transaction Corrections*, 04 April 2005 [POL-0032855]

[96] EPSPG001_0.2.doc, *S80 Impact Release 3 EPOSS Counter Operational Support Guide*, 10 May 2005 [POL-0081677]

POL00029046

893 If the SPM does not wish to accept the TC, the main option available to him in Horizon is to 'seek evidence'. At this stage, in my opinion, there would have to be a cooperative investigation between the branch and the central PO department involved, to investigate the cause of the discrepancy, with the SPM contributing his knowledge about what happened in the branch. I have not seen evidence of how the investigation process takes place. In section 9.6 below I describe evidence about the volumes of these disputed TCs, and the volumes of those upheld.

894 A document 'Horizon Architecture Overview'[98] in 2006 emphasises the need for precision in handling TCs:

Within the branch estate, the majority of the products that are sold by Post Office are on behalf of a third party (a "Client" in Post Office language) – for example payment of a British Gas or BT Bill. The fees paid by the Client for this service are typically related to the amount of manual work that needs to be undertaken by branch staff rather than the value of the transaction – resulting in very low margins (Post Office's turnover is approximately 1% of the £110 billion worth of transactions it handles each year and its margin is a low percentage of this).

One consequence of the low margins is that Post Office has to be extremely careful to minimise the impact of any errors or faults in the solution. One example of this is that for online authorisations every individual transaction is reconciled with the third parties view and all errors are investigated (typical retail organisations would just check that the total for the day is accurate to within an agreed error margin with the third party).

895 In my opinion the Horizon element of the process of reconciliation and TCs was adequately designed, using the necessary mix of automated and manual processes to '*minimise the impact of any errors or faults in the solution* '. I have not seen evidence implying that Horizon was ever the cause of errors. From a technical viewpoint I am not aware of any obvious ways in which it could be made more precise - although I have no knowledge of the business processes for creating TCs used by the central PO departments.

896 In his paragraphs 6.50 - 6.59, Mr Coyne describes how TCs proceed from POLSAP to TRS to the branch and the options available to the SPM for handling them. These descriptions are consistent with my own knowledge.

9.4 My Opinions on Horizon Issues 5 and 15

897 Issue 5 asks how the Horizon system 'itself' compares transaction data against transaction data from outside Horizon.

898 The word 'itself' is problematic - because while reconciliation between transaction recorded on Horizon and the same transactions recorded by PO's clients was extensive and automated, it was done in a variety of different ways and by different IT systems, some of them outside Horizon itself.

899 Regardless of this, for essentially all of the clients for whom PO acted as agents, there was an automated process of comparison of the transactions as recorded by Horizon and as recorded by the client.

---

[97] 1.5 5. Operations Manual version 7 December 2006 - pages 9-13.pdf, *Processing any outstanding Transaction Corrections*, 7 December 2006 [POL-0184501]

[98] TD/ARC/039

POL00029046

900 This process compared millions of transactions per day, and was part of an important robustness countermeasure, relying on redundant data storage (RDS), with automated comparisons of the Horizon version against the client version. If discrepancies were detected, this allowed errors from a variety of sources, notably human errors in carrying out transactions or recovering recoverable transactions, to be corrected (UEC).

901 Issue 15 asks about transaction corrections.

902 If a discrepancy was detected in reconciliation, a correction would need to be made, consistent with the principles of double entry accounting (DEA) - in particular, keeping POLSAP and BRDB in step (another example of RDS). PO's way to make this correction involved manual inspection of data (MID) by PO central staff. If this review found that the cause was an error in the branch, PO would issue a transaction correction (TC), which was followed by review and acceptance or contesting of the TC by the SPM (again MID). This whole process was a constrained double entry process, and kept POLSAP and BRDB in step.

903 The claimants imply that errors in the MID component of reconciliation and TCs may have led to shortfalls in their branches. I have estimated an upper limit on the likely scale of such shortfalls, which is a second-order effect (an initial error, followed by another error in its correction). As a second-order effect, the effect is expected to be small. Based on the volume of TCs, and the proportion of contested TCs, I estimated the magnitude of this amount to be, on average, less than £2 per branch per month - compared with the claimants' claimed losses which averaged £360 per branch per month. So errors in reconciliation and TCs did not contribute significantly to the claimants' claimed losses.

9.5 Mr Coyne's opinions

904 Mr Coyne addresses reconciliation and transaction corrections in section 6 of his report.

905 When describing reconciliation at para 6.13 onwards, he places an emphasis on manual processes, which in my opinion may be misleading.

906 Because the PO handles millions of agency transactions per day on behalf of its clients, it is necessary that the process of reconciliation - detecting any discrepancies between the PO version of those transactions, and the client's version of the same transactions - has to be automated. Manual comparison would in my opinion be infeasible. It is only when transactions with discrepancies are revealed by this automated process, that any manual processes are used - to find the cause of the discrepancy, and to ensure it is allocated to the correct account. To find the cause of a discrepancy, in my opinion inevitably a manual process is needed. This does not emerge clearly from Mr Coyne's paras 6.13 - 6.21.

907 In paragraph 6.38 of his report, Mr Coyne makes a detailed point about one of his requests for information. I respond to that detailed point in Appendix H.

908 The general thrust of Mr Coyne's opinions on issues 5 and 15 is to emphasise that reconciliation and TCs were possibly an error-prone process - and thus to imply that these errors might have contributed to the claimants' losses. This emphasis on possible errors in reconciliation and TCs, appears, for instance, in paras 6.45 and 6.77 of his report.

POL00029046

909 Mr Coyne has not attempted to quantify the number of these errors in TCs, or their impact on branch accounts. I have done so in section 9.4 of this report. I found that the possible financial impact of errors in TCs is probably less than 1% of the shortfalls experienced by the claimants.

910 The individual incidents of possibly incorrect TCs described by Mr Coyne at his paras 6.64 - 6.69 do not alter this opinion. Clearly with several thousand TCs in any month, it is possible to cite small numbers of them that were in error; but as described above, their financial impact is small compared to the claimants' shortfalls.

911 In para 6.3 Mr. Coyne quotes a document from PO about reconciliation which states: "… *not all system faults will lead to corrective action and this is generally done on a contractual and/or cost benefit basis*". The previous paragraph makes it clear that 'system faults' include events such as '*a telephone line being dug up*'.

912 There are six references in Mr Coyne's report to PO making decisions 'on a cost benefit basis' - three of them in the context of reconciliation. These references might be taken to imply that a cost benefit basis is a selfish or short-sighted commercial thing to do, rather than (for instance) putting the interests of SPMs first. If that is the intended implication, then in my opinion it is not necessarily correct.

913 Thousands of business decisions are taken every day on a cost benefit basis, in businesses of all sizes - ranging from the PO central functions to individual PO branches. If reconciliation reveals some small discrepancy in some transaction - say a few pounds and pence - then there is a valid business question of whether to spend the administrative effort required to fully investigate it and take corrective action, or more simply to absorb any loss centrally. In my opinion the likely administrative costs may well dominate in many cases, so it is important for the PO to have guidelines - on a cost-benefit basis - as to what discrepancies should be handled in what way.

914 In just the same way, each SPM will take decisions - on a cost-benefit basis - designed to make best use of his own time. For instance, in monthly balancing, the SPM must decide which discrepancies to investigate, and which to accept without investigation.

915 The claimants may wish to imply that in some cases, the line of least resistance for some central reconciliation function would be to 'blame it on the branch'. In my opinion, it would not be that simple. If PO centrally were to blame a TC on the branch, in cases where it was not in all likelihood the responsibility of the branch, this would lead inevitably to branches disputing more TCs, and I would expect the administrative costs of investigating any disputed TC to often exceed the amount involved.

916 So purely on a cost-benefit basis, it may be in PO's interest to keep their SPMs well supported, and not to blame them unnecessarily for discrepancies - in order to minimise PO central support costs, but also to motivate SPMs, not to distract them with unnecessary disputes and investigations, to enable the SPMs to run successful businesses for the PO, to satisfy PO's customers better, and so on.

917 So, this area involves complex business trade-offs, which the experts have not been asked to investigate. In my opinion, it is not appropriate to portray, or to imply, any over-simplified 'cost benefit' motivation for PO to treat its SPMs badly.

POL00029046

9.6 Financial Impact of Errors in TCs on Claimants' Branch Accounts

918 Because Mr Coyne has discussed Post Office's reconciliation processes and the risk of errors in those processes without offering any analysis of the possible impact of such risks, it may be helpful for me to address this point quantitatively, using evidence available.

919 The defendants have disclosed the following information about TCs[99]:

	CREDIT		DEBIT			
Year	Volume of TCs Issued	Value of TCs Issued	Volume of TCs Issued	Value of TCs Issued	Total Volume	Total Value
2005	1151	-£ 316,059.35	11191	£ 8,412,703.76	12342	£ 8,096,644.41
2006	20799	-£ 5,348,456.00	87692	£ 25,215,930.31	108491	£ 19,867,474.31
2007	31288	-£ 9,190,474.09	100774	£ 32,031,684.88	132062	£ 22,841,210.79
2008	41967	-£ 8,417,508.40	98542	£ 20,971,413.52	140509	£ 12,553,905.12
2009	42999	-£ 7,939,353.32	98376	£ 19,993,591.51	141375	£ 12,054,238.19
2010	46460	-£ 8,118,634.08	103984	£ 19,454,770.24	150444	£ 11,336,136.16
2011	54006	-£ 14,580,500.19	79252	£ 19,086,336.06	133258	£ 4,505,835.87
2012	51246	-£ 11,064,648.41	73128	£ 10,089,399.59	124374	-£ 975,248.82
2013	46544	-£ 10,422,881.17	59332	£ 8,964,914.99	105876	-£ 1,457,966.18
2014	62731	-£ 11,431,411.43	51309	£ 18,989,665.02	114040	£ 7,558,253.59
2015	58814	-£ 53,667,783.90	50338	£ 11,435,707.19	109152	-£42,232,076.71
2016	54837	-£ 9,943,787.13	55114	£ 18,349,729.99	109951	£ 8,405,942.86
2017	48922	-£ 8,353,469.31	68960	£ 15,708,356.78	117882	£ 7,354,887.47
2018	9762	-£ 2,240,040.20	20834	£ 4,102,186.97	30596	£ 1,862,146.77
Grand Total	571526	-£ 161,035,006.98	958826	£ 232,806,390.81	1530352	£ 71,771,383.83

Table 9.1 - Volumes of TCs by year

920 This table shows that there are TCs which credit the branches (left-hand columns) and which debit the branches (right-hand columns) - with a fairly high level of cancellation between the two - £161M credit and £232M debit.

921 Summing the magnitudes of these two gives approximately £400M of TCs flowing through branches, over a 14-year period during which the average number of branches was about 13,600. So, the mean amount of TCs (either credit or debit) was about £290 per branch per month[100] - which was also about one TC per branch per month.

922 How many of these TCs might have been in error? Paul Smith's witness statement describes some approximate numbers of disputed TCs. If this evidence is accepted, it enables me to calculate the approximate financial impact of errors in processing TCs. Since one may assume that any erroneous TC is likely to be disputed (along with many TCs that are correct), the level of disputed TCs is in my opinion an approximate upper limit on the level of erroneous TCs.

923 Where there is evidence on the proportion of disputed TCs upheld, that may give further information on the level of TCs which were erroneous in the first place. When a disputed TC is upheld, I may infer that the TC was in error (i.e. the SPM said it was in error, and after investigation PO agreed); whereas, is it is not upheld, that may indicate that after further investigation, PO concluded that it was not issued in error.

---

[99] Document reference for this spreadsheet TCs.xslx

[100] Here, magnitudes of credits and debits are added - because any of them may contain errors

POL00029046

Although that does not indicate with certainty that the TC was correct, because there was further investigation, I infer that in many cases when a dispute was not upheld, the TC was correct.

924 The levels of disputed TCs and upheld disputed TCs in Mr Smith's witness statement[101] are as follows:

Type of TC	Paragraph of WS	Approximate percentage disputed	Approximate percentage disputed and upheld, if known
Cash, Bureau and Personal Banking	17, 18	2%	0.2%
Personal Banking	19	small	small
Camelot, Debit Card & ATM	20	10%	2%
Santander	23	15%	10%
DVLA	24	small	rare
Drop & Go	25	Small	rare
Postal Orders	26	rare	rare
MoneyGram	27	Rare	rare
cheques	28	1%	rre

Table 9.2 - Proportion of TCs disputed and proportion upheld

925 I proceed on the assumption that these figures (which are the only ones available to me) are accepted by the court. If they are not, a different calculation along the same lines may possibly be appropriate.

926 For Camelot, Debit Card and ATM, I assume that the number of compensating TCs (as described in Mr Smith's witness statement) equates approximately to the number of disputes upheld - that if a dispute was upheld (i.e. if the original TC was found to be in error), a compensating TC was issued.

927 From the table above, the only category of TCs with a possible error rate as high as 10% is Santander. As will be evident from the table below, Santander do not account for a large proportion of TCs. Therefore, the Santander row has little influence on the overall maximum, which I take to be 2% (as Camelot accounts for a large proportion of TCs[102] - see below)

928 In order to understand the maximum possible level of errors in TCs a little better, it is useful to know something about the numbers of TCs of different types. The table below[103] is a TC summary from a month in 2013:

[101][101] The final column includes calculations by me. For instance, the figure of 10% from Santander is (disputes upheld/ errors received from Santander) in the table at para 23

[102] Camelot TCs are approximately 1/3 of the total, so 2% errors in Camelot imply (2/3)% in the total. Therefore assuming 2% errors in the total is conservative, by a factor of 3.

[103] TC Reporting Pack 126874859.xls

POL00029046

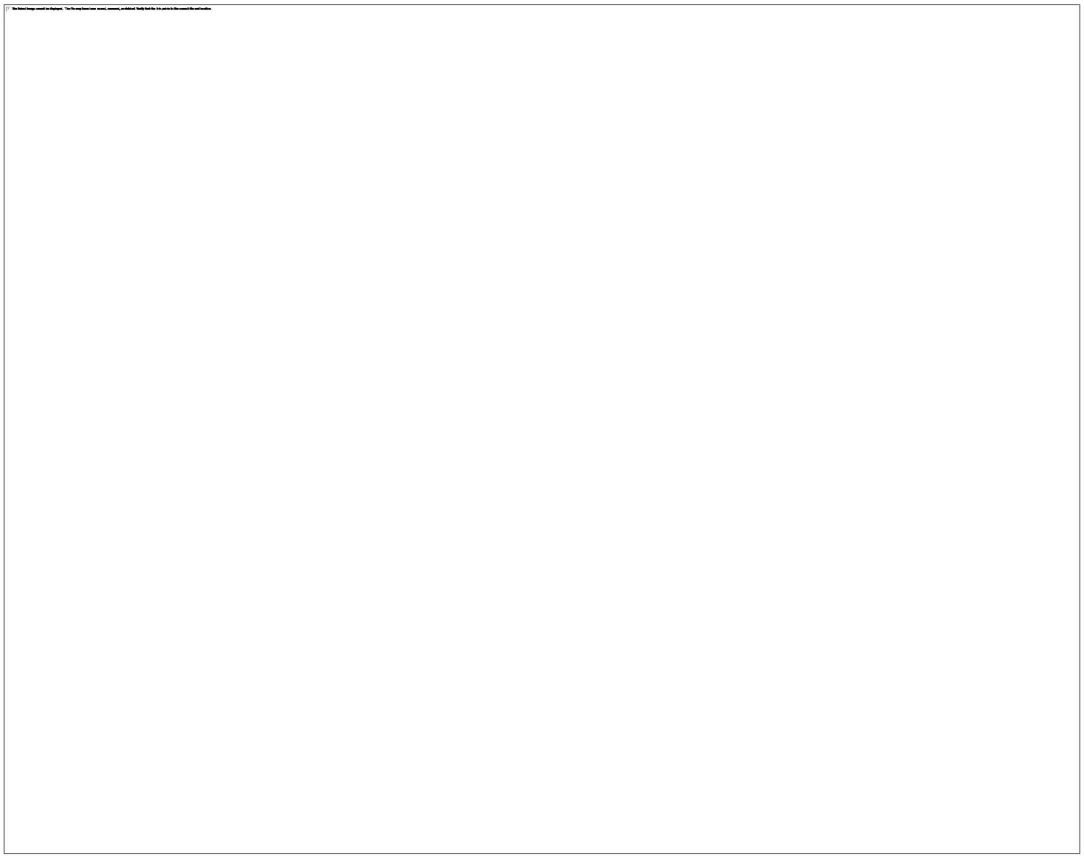


Table 9.3 - Volumes of TCs by origin (2011/2012)

929 The main point to note from this table is that the two biggest categories of TC by volume are Camelot and Cash Rems from Branch. Between them, they account for more than 50% of the volume of TCs, and no other category accounts for more than about 8%. (Mr Coyne confirms this in his para 6.67 and his Appendix C)

930 For both Camelot and cash remming, there are well understood sources of error in the branches. These are described in another Excel worksheet in the same workbook as the table above:

Camelot	Correct accounting procedures followed but incorrect figures entered from Lotto summary to Horizon. (cheque prize payment included for example). Correct figures entered on Horizon, but transaction details not accounted for on same or next day.
Cash Rems from Branch	Pouch remmed in at Cash Centre, contents differ to amount stated on advice note. Resulting discrepancy should be held in Rem Suspense and redeemed when Transaction Correction accepted. Pouch despatched but not remmed out or remmed out twice.

Table 9.4 – Sources of branch errors

931 The same workbook has a league table of the branches with the highest rate of TCs. For the leading branches, the main source of TCs was remming of cash.

POL00029046

932 If there were 2% of TCs issued in error[104], which were resolved incorrectly against the branch, the net effect on branch accounts would be £6 per branch per month. As described above, in my opinion this is a conservative upper limit on the magnitude inaccuracies introduced into branch accounts - which could be of either sign.

933 However, because, as described in section 8.5, claimants' branches were on average three times smaller than typical PO branches (as measured by number of customer transactions per day), one would expect the number of TCs issued to claimant's branches, and the number issued in error, to be three times smaller than the average for all branches - and therefore to be approximately £2 per month. This again is a conservative upper limit on the magnitude of discrepancies.

934 This figure is to be compared with the mean shortfall per month claimed by the claimants - which , as I described in section 8, was £360 per branch per month. A maximum of £2 per month from erroneous TCs is less than 1% of this amount.

935 Therefore, errors in reconciliation and TCs cannot account for a significant part of the claimants' claimed shortfalls.

936 At the level of an individual claimant - if, for example, a claimant were to say that he lost £200 in one month, due to errors in processing TCs - then in the absence of further evidence, the probability of that claim being correct [105]is about 2/200 = 1%.

937 As before, any claim of several erroneous TCs, on one month or in several months, would have a much smaller probability of being correct - because errors in TCs are statistically independent events, so their small probabilities multiply.

938 I note that even this small level of shortfalls in claimant's branches from erroneous TCs is expected to have arisen from human errors in the PO back office, rather than from bugs in Horizon. I do not know how that relates to the scope of the Horizon trial.

---

[104] I assume here that there can be errors, which may cause a shortfall to the branch, in TCs which are either credits or debits to the branch. Thus, the sum of magnitudes of credits and debits, which is £161M + £232M = £393M, scaled down to an individual branch - is the appropriate figure to apply an error percentage to.

[105] This is because, if one month had a TC error of £200, and the average over all months was £2, there would need to be a further 99 months with a zero TC error, to make the average come to £200/100 = £2. So, the £200 TC occurs for one month in a hundred

POL00029046

10. HORIZON ISSUES – FACILITIES AVAILABLE TO SUBPOSTMASTERS

10.1 The Issues and My Opinions

939 In this section I address Horizon Issues 2, 9, 14, which have been grouped together because they all concern facilities and information available to Subpostmasters.

940 **Issue 2**: Did the Horizon IT system itself alert Subpostmasters of such bugs, errors or defects as described in [Issue] (1) above and if so how?

941 **Issue 9**: At all material times, what transaction data and reporting functions (if any) were available through Horizon to Subpostmasters for:

a. identifying apparent or alleged discrepancies and shortfalls and/or the causes of the same; and

b. accessing and identifying transactions recorded on Horizon?

942 **Issue 14**: How (if at all) does the Horizon system and its functionality:

a. enable Subpostmasters to compare the stock and cash in a branch against the stock and cash indicated on Horizon?

b. enable or require Subpostmasters to decide how to deal with, dispute, accept or make good an alleged discrepancy by (i) providing his or her own personal funds or (ii) settling centrally?

c. record and reflect the consequence of raising a dispute on an alleged discrepancy, on Horizon Branch account data and, in particular:

i. does raising a dispute with the Helpline cause a block to be placed on the value of an alleged shortfall; and

ii. is that recorded on the Horizon system as a debt due to Post Office?

d. enable Subpostmasters to produce (i) Cash Account before 2005 and (ii) Branch Trading Statement after 2005?

e. enable or require Subpostmasters to continue to trade if they did not complete a Branch Trading Statement; and, if so, on what basis and with what consequences on the Horizon system?

943 Issues 2, 9 and 14 are on the face of them mainly factual issues, which can be largely resolved by factual evidence, and might not in themselves lead to much expert disagreement.

944 However, they need to be approached in the light of the claimants' case, and certain assumptions apparently built into it, and implied in Mr Coyne's report. These assumptions appear to be that:

- It would have been a good thing to provide SPMs more information about the workings of Horizon than was given to them.
- If there was a fault in Horizon, there should have been some useful automatic way for Horizon to tell SPMs what it was.

POL00029046

- In the case of an anomaly, it was incumbent on the SPM to dispute the cause of the anomaly with PO.
- In doing so, SPMs could usefully use information about the back-end systems of Horizon to infer that some anomaly was caused by a bug in Horizon.
- Because SPMs did not have all this information, but PO did, there was an asymmetry of information between SPMs and PO - which PO used to unfairly attribute the effects of bugs in Horizon to human error by the SPMs.

945 In my opinion, these assumptions all rest on an unrealistic picture of how commercial IT systems are built, used and supported:

- It is not a good thing to give the users information about parts of an IT system which they do not encounter in their daily work, and which they know very little about. They will be perplexed by it.
- To anticipate the small proportion of cases where the IT system is in error, there is no point in trying to educate all the users in details and terminology of the system which will never concern them.
- An IT system can give its users useful warnings and error messages in a variety of situations, but generally not in the case of previously undiscovered bugs in the system.
- When the developers of an IT system discover some bug or defect in it, the best thing to do is to fix it, rather than to create some new error message to the users
- When an IT system gives results, which puzzle its users (for any cause), further automated messages from the system are only of limited help to users. They need support from a human being, who may need to take account of the circumstances and bring to bear a wide variety of knowledge.
- Anomalous results may arise for a wide variety of reasons - from human error, to errors in processing at the back end. Understanding the causes depends inevitably on cooperation between the user (who knows what he did) and support staff (who know much more about back end systems). To portray this cooperation as a dispute is fundamentally misleading.
- Staff and organisations who support an IT system have a strong incentive to understand bugs and to get them fixed, to reduce their future workload. They have no interest in leaving bugs unfixed, so the same problems keep recurring.

946 Putting to one side these assumptions in the claimants' case, my opinions are:

947 **Issue 2**: Horizon did not in general alert SPMs to any significant bugs or other defects in the system itself. Nor should it have done.

948 **Issue 9**: In my opinion, from comparing human errors with software error rates in Horizon[106], most discrepancies are caused by human error. The functions available from Horizon, when used in accordance with PO guidance and procedures, enable SPMs to identify the causes of such discrepancies. SPMs and

[106] This point is addressed, for instance, in Appendix F and below in section 10.2

POL00029046

their staff are the best placed to investigate discrepancies, because they are the only people who have first-hand knowledge of what happens in their branches. The PO and Fujitsu support teams can only use their knowledge of systems and the data stored within them; whereas the SPM can use their knowledge of what happens in the real world.

949 The main concern of an SPM is the successful running of their branch. This means that they may have limited time and patience to investigate discrepancies in their accounts, whatever they may believe is the cause. The reports available to them focus on activities carried out within the branch, their key area of expertise. If they are, nevertheless, unable to identify the problem, their best course of action is to ask for help.

950 **Issue 14** asks a number of specific questions about the facilities of Horizon for SPMs, which I answer in section 10.4

10.2 Approach to the Issues: Assumptions

951 In dealing with these issues, it is particularly important that I clarify what falls within my scope. The scope of these issues is limited to the functionality of Horizon, and not how SPMs use it to run their branches – nor whether the functionality is adequate for this purpose.

952 There are in my opinion certain assumptions behind these Horizon issues; and having read Mr Coyne's report, his report appears to reinforce some of those assumptions. Therefore, I need to address those assumptions before addressing the detail of the issues.

953 An assumption underlying all three issues appears to be that providing more information to SPMs would have been a good thing - enabling them to understand bugs and defects in Horizon (issue 2) identify discrepancies (issue 9) and dispute discrepancies (parts of issue 14).

954 This seems to me to make assumptions about the role and knowledge of SPMs, and about their relationship with PO, which should not be accepted without question - and some of which, once examined, turn out to be unrealistic.

955 In my experience of many types of IT project, expecting too much knowledge of the users - more than they need in order to use the system - is a common mistake in the design of systems, and can often make systems harder to use, and make users more likely to make errors and dislike the system. If SPMs were expected to understand bugs and defects in Horizon (as in issues 2 and 14), that would require them to understand a large amount of detail about the Horizon back-end systems - their names, roles, interactions and so on - all of which appears to have nothing at all to do with the SPM's daily work.

956 Suppose, as asked in Horizon Issue 2, Horizon had automatically produced some error message of the form 'Transaction X has resulted in a discrepancy in data between TPS and DRS', in my opinion the only possible reaction from an SPM would be: "What on earth am I supposed to make of that?". To make any such message meaningful to the SPM would require a large amount of extra documentation and training material, which would be of no use to him in his day-to-day work. He would never spend the time to understand what is happening behind the scenes.

POL00029046

957 In my experience, for any IT system to subject its users to its internal details is usually a mistake. Users typically want to know as little as possible about the internal details of the system. Good design always involves 'information hiding' and keeping things as simple as possible for users. The approach of providing a help desk, where a person could try to understand what problem the SPM was experiencing, and try to help him, was in my opinion the only viable one.

958 In a context unrelated to the Post Office, we have all experienced the frustration we feel when some human help service has been replaced by a machine - for cost-saving reasons. Issue 2 appears to be asking - could PO have given its SPMs automated support in Horizon, in the place of human support?

959 Similarly, there seems to be an assumption behind issues 9 and 14 that, given enough automated information, SPMs could somehow identify the causes of shortfalls (deep inside Horizon), and might have the knowledge and persistence to 'dispute' them with Fujitsu support staff, whose job it is to look at such issues, and who would have a deep knowledge of Horizon internals.

960 This assumption seems to me to be a misconception. It may arise in part because lay people (which for these purposes would include SPMs) do not understand that, for a variety of reasons (such as the need for robustness countermeasures, the needs of many different classes of users, the obduracy of technology, and the evolution of systems over many years), the internal details of any large IT system are always much more complex than you would expect. There is just much more code needed that you would imagine, and the internal behaviour of the system is more complicated. Many failed IT projects are a testament to this.

961 Horizon has taken more than 3,000 man-years of effort to build. To imagine that any SPM can, in his spare time when he is not managing his branch, understand enough about Horizon internals, and how they might or might not go wrong, to debate and dispute the causes with PO and Fujitsu seems to me a misconception. It is an unrealistic view of the knowledge and predispositions of Horizon's main users. In my experience, IT developers are prone to have too high expectations of their users' knowledge; but this level of misconception would seem to be extreme.

962 The true picture, it seems to me, is simpler. SPMs know what happens in their branch, and they should know how to use Horizon. Support staff know all about Horizon, and what may go wrong for a variety of reasons. When some anomalous incident occurs, the best way to understand it is by a cooperation between these two parties, sharing their knowledge. Without that cooperation, it is more difficult for problems to be resolved, and so they may keep on being repeated. The Fujitsu support role requires deep and broad knowledge of Horizon internals - used to filter the large amounts of information available, to find the pieces relevant to some problem. To pass that responsibility over to the SPM would in my opinion be inappropriate - SPMs have neither the knowledge or the time to diagnose the deep causes of problems.

963 A final assumption to be addressed here is that the support function would always start by assuming that any problem had arisen from an error in the branch and would not give sufficient credence to the possibility that it might have arisen from a software error.

POL00029046

964 In my experience, any competent IT support operation is grateful to its users, when they draw its attention to any problem which can be fixed, to reduce the future costs of support. It will use these opportunities to improve the system for all users - not to fob some users off. Repeated evidence in KELs shows that Fujitsu ran such a competent support operation. The great majority of KELs show problems solved.

965 However, this assumption deserves careful consideration, because the evidence shows that human errors in the branch did occur much more frequently than errors induced by software - as one might expect. One measure of errors in the branch is the level of TCs - of which a large proportion arose from human errors, for instance in remming cash in or out, or in manually recovering recoverable transactions. These occurred at a rate of approximately one per branch per month; whereas software errors, as shown by the KELs, were much less common.

966 Furthermore, a software error, once diagnosed, could be permanently fixed across the whole PO network; but human errors would keep on recurring.

967 So, when the support desk was contacted about some problem, the overwhelming likelihood was that the cause really was a human error. The starting assumption, that it was probably an error in the branch, was correct in most cases. Mr Parker's witness statement says of this at paragraph 49: ' *it is a simple truth of support that the majority of issues reported in a system are attributable to user action or user misunderstanding of system functionality. Hence, someone working in a support environment analysing a new issue would examine the possibilities of user error as a first hypothesis but any final conclusion is only generated based on the evidence.* '

968 This agrees with my experience in other applications. In these circumstances, some other evidence would be required, to show if the cause was a software error, rather than human error. In my experience, that evidence would not be one user saying: "I swear I never did that.". It would have to be something else, such as a recurring pattern across several incidents, or evidence from system logs.

969 Evidence from the KELs indicates to me that the Fujitsu support service was effective at spotting recurring patterns, and at delving into logs and other evidence to find the true causes of problems. If there was a software error, in my opinion the possibility of human error could usually be eliminated. Of course, the support service might not get it right every time; and even on the occasions when they correctly attributed a problem to human error (i.e. most occasions), sometimes the SPM might cling to a different account that he had never done anything wrong. This is a natural human reaction.

10.3 Horizon Issue 2

970 **Issue 2**: Did the Horizon IT system itself alert Subpostmasters of such bugs, errors or defects as described in [Issue] (1) above and if so how?

971 In common with most IT systems, Horizon generates messages to report the occurrence of certain errors to users and operators. Error messages displayed on the counter screen or presented on reports alert SPMs to conditions that may indicate the presence of bugs or other defects as described in Issue 1.

POL00029046

972 I agreed the following with Mr Coyne in our Joint Statement: '*The extent to which any IT system can automatically alert its users to bugs within the system itself is necessarily limited. While Horizon has automated checks, which would detect certain bugs, there are types of bugs which would not be detected by such checks.*'

973 Further, as I discuss in section 10.2, it would be counter-productive for Horizon to alert its users with precise details of abnormal conditions beyond their day-to-day experience of the system – for example, in back-end and other systems remote from their counters. To do so, it would need to assume terminology and knowledge well beyond that of a typical SPM.

974 The system does, however, record significant or unexpected events in logs. For instance, in the receipts/payments mismatch, logs were used to identify affected branches. Horizon is operated by specialist staff, who are alerted by the system if certain events occur. Such alerts trigger investigations that may detect bugs, which could potentially affect SPMs and their accounts before any branch users become aware that anything is wrong. The logs may also be checked proactively by the support team in response to a report from an SPM. These measures are amply substantiated in KELs.

975 Horizon and its ecosystem are underpinned by a complex set of software, hardware networks and business processes. In my opinion, it is more rational that any bugs or other defects are investigated and analysed by experienced people following mature processes – rather than expecting that SPMs themselves could diagnose problems if they were given more detailed information.

976 To summarise my opinion on this issue, Horizon did not in general alert SPMs to any significant bugs or other defects in the system itself. Nor should it have done.

10.4 Horizon Issue 9

977 **Issue 9**: At all material times, what transaction data and reporting functions (if any) were available through Horizon to Subpostmasters for:

a. identifying apparent or alleged discrepancies and shortfalls and/or the causes of the same; and

b. accessing and identifying transactions recorded on Horizon?

978 This issue focuses on the functions available to SPMs for identifying and investigating discrepancies. The discrepancies in question are differences between the amounts of cash and stock held by the branch and the amounts calculated by Horizon. Shortfalls are discrepancies where the figures held in Horizon are higher than those declared by the SPM. Discrepancies may also be surpluses in favour of the SPM.

979 In Horizon, balancing means counting all cash and stock holdings and checking that the position matches the figures held within the system. This is the process that identifies any discrepancies.

980 If a discrepancy is detected, the question is: why is it there? There are many reasons why such discrepancies occur. These include the following:

- Transactions may not correctly record the changes that occurred in cash or stock levels, e.g. Horizon was told that £30 was paid out to a customer whereas £50 was actually paid.
- Changes in cash or stock were not recorded at all, e.g. a book of stamps was mislaid.

POL00029046

981 Horizon has always provided SPMs with a comprehensive suite of reports, which can be previewed on screen as well as printed. More than one hundred reports are available[107]. These include:

- reports by stock unit (SU) on a daily or weekly basis, or by user;
- balance reports; and
- journals such as Transaction and Event Logs.

Not all of these are relevant for dealing with discrepancies.

982 It should be noted that the information available in the branch and presented to SPMs has always been stored centrally. For information about business transacted in the branches, PO accesses precisely the same information as the SPMs.

983 David Johnson is PO's Training & Audit Advisor. In his witness statement dated 28 September 2018, he describes the following features of Horizon (HNG)[108]:

> *'8.1 logging in, stock units and basic transactions;*
>
> *8.2 daily cash declarations and periodic balancing;*
>
> *8.3 the cash declaration process;*
>
> *8.4 the reports available in branch to identify causes) of discrepancies;*
>
> *8.5 the balancing process; and*
>
> *8.6 dealing with discrepancies at the end of a trading period.'*

984 These descriptions, if accepted. are relevant, not only to this Issue 9, but also to Issue 14 treated in section 10.5. As they are factual and largely uncontroversial, they may provide additional evidence alongside what I have seen from Fujitsu and PO documentation. However, in this report, I focus on addressing the agreed Horizon issues and avoid commenting on the surrounding business policies, procedures and administration.

10.4.1 Accounting procedures

985 In 2005, PO changed the accounting process in branches. Before 2005, the process was called 'Cash Accounting' and occurred weekly. After 2005, the process was called 'Branch Trading' and was undertaken at the end of every 4 or 5-week period, known as a Trading Period (TP). What follows in this sub-section describes primarily the procedures in place since the introduction of Branch Trading in 2005. The procedures followed in both periods of time are similar. The main differences are discussed in section 10.5.4.

986 PO requires that cash is declared for every SU every day.[109] This entails the branch physically counting the cash held within that SU and entering this information into Horizon. The system will display any

---

[107] SD/DES/005 for the original system and DES/GEN/SPE/0008 for HNG

[108] See paragraph 8

[109] Post Office Onboarding - Counter Guide v6.0

POL00029046

discrepancy between the total figure declared and the system-derived figure. For an individual stock unit, this happens automatically whereas, for a shared stock unit, the user must request it (by touching the Variance button).

987 Branches must perform a balance at the end of either (i) every week during Cash Accounting or (ii) every TP. Balancing can be performed at any time. Larger or busier branches may choose to balance every working day or week. The balance involves a manual count of all cash and stock, a comparison of the cash and stock on hand figures against the system derived figures, the correction of any surpluses or gains (this step being the major change between Cash Accounting and Branch Trading) and then the submission of the accounts.

988 The more frequently branches complete cash declarations and balance reports, the sooner they will be able to identify any discrepancies. It is a fair assumption that staff are more likely to recall an interaction with the customer, which may have caused a discrepancy if that issue is investigate promptly. Daily cash declarations, if completed, will therefore act as an early warning system for discrepancies and likely limit the transactions to be investigated to a single day's trading.

989 There are two categories of report available from Horizon: Counter and Office reports. Counter reports provide details of transactions carried out by a specific SU, whereas Office reports cover all SUs – in other words, the entire branch. In branches with only one SU, both categories of report show the same information.

990 When discrepancies are identified, the main tool used to find their causes is the **Transaction Log**. This allows any user with access to Horizon to obtain a chronological list of the transactions completed in the branch. The log can be used to browse through a list of transactions, or the output can be filtered by selection criteria such as TP, date, time, SU, user, product type or value. Filtering allows the user to limit the information provided, so that the branch staff can 'home in' on any anomalies. Regardless of TP dates and the intervals between balancing branch accounts (Balance Periods), the log can be used to investigate up to 60 days back in time. Prior to 2010 with the original Horizon system, this period was 42 days.

991 A user may spot errors they have made in entering data into Horizon or when handling cash or stock by examining the transaction log. For example, a user may recall giving a customer a cash withdrawal of £100 at a particular time of the day, but by checking the log they may spot that they incorrectly processed the transaction as a deposit. This would create a shortfall of £200 in the branch accounts (Horizon will think that the user has taken a £100 deposit whereas in fact the user has given the customer £100).

992 The following specification shows the detailed layout of the Transaction Log (using illustrative data):

POL00029046

CHARTERIS

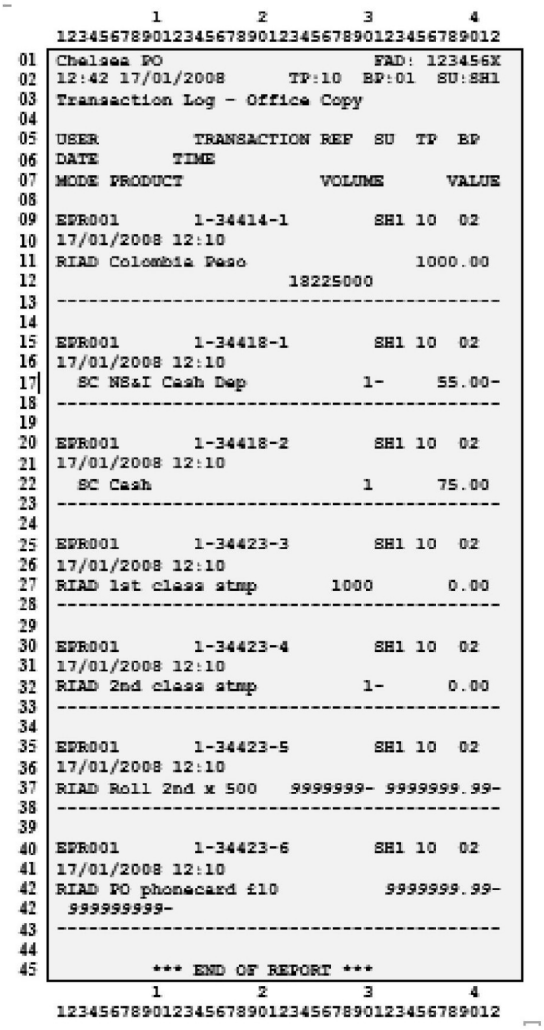


Figure 10.1 - Transaction Log layout

993 The printable area for a counter printer is only 8cm wide. Each transaction printed on the log uses a minimum of four lines. As stated above, all reports can be previewed on screen where users see an image of the printable report. This means that printing of this log can be avoided. Also, as described in paragraph 990 above, the size of the output can be reduced by filtering.

994 The **Event Log** reports, in chronological order, events that have taken place. The Functional Specification dated 2009[110] identifies more than 40 different events, which can be included in reports. These include:

- User logon and logoff, which enables the SPM to identify which user was responsible for each transaction;
- Cash and stock declarations;

[110] SD/DES/005

POL00029046

- Reporting;
- Balancing;
- Cash made good;
- Rollovers;
- BTS creation.

In the original Horizon system and before Branch Trading was brought in, a slightly different set of events was relevant. Reports can be filtered using selection criteria such as date, SU, TP, and user.

995 Thus, the Event Log enables the user to see all the cash declarations that have been made, by which user and whether there were any discrepancies.

996 The **Balance Snapshot** shows details of all receipts and payments since the last time an SU was balanced. The SPM can use this to check his receipt and payment transaction totals, and the stock on hand together with the system-derived stock figures. It may be produced at any time to assist the SPM in balancing by enabling him to check that transactions have been recorded correctly on Horizon. Any stock or cash discrepancies will not be shown.[111]

997 The **Stock On Hand report** shows the derived positions of cash, cheques (if applicable), stock, foreign currency, stamps and other stock on hand. This means that the user can check the physical stock on hand in the branch against the system derived figures at any time and see if there is any discrepancy.

998 Other reports are also used for checking the details of specific transactions identified using the tools described above.

10.4.2 My opinions on Issue 9

999 To address the two parts of this issue explicitly, (b) '*accessing and identifying transactions recorded on Horizon*' is the prime method of (a) '*identifying apparent or alleged discrepancies and shortfalls and/or the causes of the same*'.

1000 I have agreed the following with Mr Coyne in our Joint Statement: '*The causes of some types of apparent or alleged discrepancies and shortfalls may be identified from reports or transaction data available to Subpostmasters. Other causes of apparent or alleged discrepancies and shortfalls may be more difficult or impossible to identify from reports or transaction data available to Subpostmasters, because of their limited knowledge of the complex back-end systems. Identification requires cooperation of PO staff and subpostmasters.*'

1001 In my opinion, the reports available from Horizon, when used in accordance with PO guidance and procedures, would assist an SPM to investigate discrepancies.

10.5 Horizon Issue 14

1002 **Issue 14**: How (if at all) does the Horizon system and its functionality:

a. enable Subpostmasters to compare the stock and cash in a branch against the stock and cash indicated on Horizon?

---

[111] PO Operations Manual: Branch Trading - Balancing and Despatch of Documents, dated 07 December 2006

POL00029046

b. enable or require Subpostmasters to decide how to deal with, dispute, accept or make good an alleged discrepancy by (i) providing his or her own personal funds or (ii) settling centrally?

c. record and reflect the consequence of raising a dispute on an alleged discrepancy, on Horizon Branch account data and, in particular:

i. does raising a dispute with the Helpline cause a block to be placed on the value of an alleged shortfall; and

ii. is that recorded on the Horizon system as a debt due to Post Office?

d. enable Subpostmasters to produce (i) Cash Account before 2005 and (ii) Branch Trading Statement after 2005?

e. enable or require Subpostmasters to continue to trade if they did not complete a Branch Trading Statement; and, if so, on what basis and with what consequences on the Horizon system?

1003 I respond to each question in the following sub-sections.

10.5.1 Comparing stock and cash

a. *How (if at all) does the Horizon system and its functionality enable Subpostmasters to compare the stock and cash in a branch against the stock and cash indicated on Horizon?*

1004 The levels of stock and cash in a branch can only be determined when the SPM physically counts them. When the information is entered into Horizon, the system can make a comparison between the values entered and the corresponding ones derived within the system. As described in section 10.4.1 above, Horizon will make the comparison, either automatically or on request. The values held by Horizon can also be seen using reports such as the Stock On Hand report, which enable the SPM to make the comparison himself.

10.5.2 Resolve discrepancy

b. How (if at all) does the Horizon system and its functionality enable or require Subpostmasters to decide how to deal with, dispute, accept or make good an alleged discrepancy by (i) providing his or her own personal funds or (ii) settling centrally?

1005 I assume that this part of the issue refers to part (a). In that case, the discrepancy in question is between the stock and/or cash declared by the SPM and Horizon's view of those amounts.

1006 The remainder of this section 10.5 focuses on the Branch Trading regime in place since 2005. Cash Accounting is discussed where there were any significant differences.

1007 Horizon requires the SPM to deal with any discrepancies before rolling over their accounts to the next TP.[112] The SPM is free to ignore discrepancies until that point.

1008 At the end of the TP, Horizon reports to the user the amount of any discrepancy. The system invites the user to transfer this amount into the local suspense account and continue to roll over – or to discontinue this operation.

---

[112] See PO Operations Manual on Branch Trading Balancing and Despatch of Documents

POL00029046

1009 If, at the end of a TP, there is a discrepancy (i.e. either a surplus or a shortfall)[113] of less than £150, the SPM must 'make good' the discrepancy – either by removing money from the till (in the event of a surplus) or by adding money to the till (in the event of a shortfall). The ability to make good through Horizon was also available before 2005 under Cash Accounting.

1010 'Making good' causes the derived cash position to remain the same and the actual cash position to change accordingly. The next TP can then begin with a balanced account.

1011 If, at the end of a TP, a branch has a discrepancy of more than £150, they have the option to either make good or settle the discrepancy centrally. The ability to 'settle centrally' was not available under Cash Accounting. If the SPM chooses to settle centrally, they do not have to physically place cash in the till (in the case of a shortfall) at the time. Instead, a message is sent to PO's Finance Services Centre and the discrepancy is moved to a central account held in the SPM's name.[114] In the branch accounts, a transaction is added to adjust the derived cash position (up for a gain; down for a shortfall) to bring it in line with the physical cash position.

1012 An SPM may wish to dispute a discrepancy. The SPM cannot dispute a discrepancy on Horizon or record that they have raised a dispute. My understanding from the material I have read is that the SPM can contact PO for assistance and raise a dispute, but this matter is outside the scope of the Horizon Issues as it is an operational matter, so I say no more about it here.

10.5.3 Recording disputes

c. How (if at all) does the Horizon system and its functionality record and reflect the consequence of raising a dispute on an alleged discrepancy, on Horizon Branch account data and, in particular:

i. does raising a dispute with the Helpline cause a block to be placed on the value of an alleged shortfall; and

ii. is that recorded on the Horizon system as a debt due to Post Office?

1013 Horizon does not record disputes. In response to question (i) of this issue, raising a dispute about a discrepancy with PO (see my comment in paragraph 1012 above about this being outside my scope of work) is not recorded in Horizon – except, possibly later on, if there is a TC. In response to question (ii), the discrepancy is not recorded on Horizon as either a debt due to PO or a credit due to the SPM. It is simply recorded that the discrepancy is being settled centrally. As a result, the branch accounts are restored to balance.

---

[113] In his witness statement of September 2018, at paragraph 50, Mr Johnson says '*If at the end of a TP the BTS shows there is a discrepancy (i.e. either a shortfall or a surplus) of less than £150, the Postmaster must 'make good' the discrepancy.*' This statement implies that the SPM resolves discrepancies after producing the BTS, whereas Horizon insists that he does so before the BTS can be produced.

[114] Further details of this business process are provided in the Witness Statement of Dawn Phillips, the leader of the PO Agent Accounting team who oversees the process of resolving discrepancies that SPMs have chosen to settle centrally.

POL00029046

10.5.4 Accounting statements

d. How (if at all) does the Horizon system and its functionality enable Subpostmasters to produce (i) Cash Account before 2005 and (ii) Branch Trading Statement after 2005?

(ii) Branch Trading Statement after 2005

1014 Each branch is required to perform a full balance of every SU in the branch at the end of each TP. Before the final balance report is produced, the SPM must make declarations of stock on hand, foreign currency, stamps, travellers cheques and cash. After the balance report, a Postage Label report must be produced. The next step is to complete the Suspense Account report for the branch. Once all of the stock units in a branch have been balanced and rolled over to the next TP, the BTS can be produced.

1015 The BTS shows an overall summary by stock unit and it also has a list of stock on hand. It shows any discrepancies found. The user must check the statement and sign it off as accurate.

(i) Cash Account before 2005

1016 Branch Trading was introduced in 2005. Before that, in the original Horizon system, branches had to produce a Cash Account rather than a BTS. The Cash Account fulfilled the same role, but it had to be produced weekly rather than monthly. It could only be done after a series of steps similar to those required today in the build-up to the BTS.

1017 Discrepancies were handled differently until 2005[115]. All discrepancies in excess of £2 had to be posted to the suspense account, immediately after balancing and rolling over to the next Cash Account Period (CAP). The CAP was a week, rather than four or five weeks for TPs.

1018 Therefore, the SPM had to complete his weekly Cash Account before receiving any gains or settling any losses.

1019 Prior to the introduction of TCs along with Branch Trading, Error Notices were used to correct accounting errors made in branches.

10.5.5 Continuing to trade

e. How (if at all) does the Horizon system and its functionality enable or require Subpostmasters to continue to trade if they did not complete a Branch Trading Statement; and, if so, on what basis and with what consequences on the Horizon system?

1020 The PO guide to Branch Trading (balancing and despatch of documents)[116] advises SPMs, in section 29, that '*You must produce a Branch Trading Statement on the last working day on or before the Branch Trading Period end dates shown on your Branch Trading calendar.*'

1021 In his witness statement, at paragraph 48.5, Mr Johnson says that it is possible in Horizon for a branch to continue to trade without completing a BTS.

---

[115] See Horizon System User Guide Booklet G - Transactions other - Part Two (HSUG16-1), section 2.1 POL-0184485

[116] Version 7 December 2006

POL00029046

1022 The PO guide in section 19 also states: '*If you have a stock unit in your branch that has not been rolled over to the next Balance/Trading Period within the last 38 days, the Horizon system will display a screen prompt to remind you to do this.*' It goes on to warn the user of potential system problems unless rollover occurs within 38 days of the previous rollover.

1023 Nothing in any of the technical documents I have seen suggests that, when there is a failure to complete a BTS by the last day of the branch trading period, Horizon prevents the branch from continuing to trade.

1024 I note that Mr Coyne seems to disagree in paragraph 7.39 of his report. He says: '*Subpostmasters are not able to continue trading until Branch Trading Statement process is complete. If the Branch Trading Statement is not completed and therefore, the Monthly Trading Period Rollover is not completed the Post Office will contact the branch in order to rectify the situation.*' This does not address the question of whether Horizon prevents SPMs from trading.

1025 I have not seen any evidence that Horizon prevented SPMs from continuing to trade until they rolled over the branch and produced a BTS. It appears that within the system, the TP was simply extended beyond the dates required by PO with no adverse effects.

10.5.6 My Opinions on Issue 14

1026 Issue 14 is almost entirely factual, asking *how* Horizon supports SPMs in dealing with discrepancies in their branch accounts. I have given my answers above.

10.6 Mr Coyne's Opinions on Issues 2, 9, and 14

1027 Mr Coyne and I agree[117] that SPMs are not able to investigate every discrepancy. Indeed, it would not be reasonable to expect that they could. Nevertheless, in his expert report, Mr Coyne suggests that they need more information to do so:

- In paragraph 8.13, referring to the reports available to SPMs, he points out: '*as these reports are specific to counters and contain no information beyond this, they would not allow a Subpostmaster to determine the cause of an issues* [sic] *that arise at anything beyond counter level'.*
- In 8.20, he suggests that the information available from Horizon '*would not allow a Subpostmaster to determine whether a transaction has reconciled at APS Host or at any other level (harvester, client, etc.)*'.
- Finally, in 8.22: '*In conclusion, Post Office had access to far more comprehensive information* [word omitted] *relation to the Horizon system. If an error occurred beyond counter level, Subpostmasters would need to rely on Post Office to identify and resolve the issue. If that issue or its* [word omitted] *was not properly identified for any reason, then the Subpostmaster would be at risk of being liable for a Transaction Correction.*'

1028 In section 10.2 above, I explain why, in my opinion, SPMs do not need and may not be able to use more information about what happens to their data beyond their branch.

1029 Issue 14 asks about Horizon. In part (c) (i), the experts are asked whether '*raising a dispute with the Helpline cause[s] a block to be placed on the value of an alleged shortfall*'. I take this to mean a block in Horizon. In section 10.5.3 above, I give my opinion that Horizon does not record disputes.

---

[117] See paragraph 944 above

POL00029046

1030 In paragraphs 7.12, 7.17 and 7.42 of his report, Mr Coyne makes a number of claims about processes that operate outside of Horizon. I have not addressed these claims, because those processes appear to lie outside our scope.

1031 In question (c) (ii), the experts are asked whether the shortfall is recorded on the Horizon system as a debt due to Post Office. I conclude that a shortfall is not recorded as a debt to PO on Horizon. Mr Coyne states in his paragraph 7.37, '*A loss is recorded as a debt to the Post Office in the event the discrepancy is upheld by the Post Office following any dispute.*' This appears to imply that he disagrees.

1032 In relation to question (e), Mr Coyne states in his paragraph 7.39: '*Subpostmasters are not able to continue trading until Branch Trading Statement process is complete.*'. I do not agree (in section 10.5.5 above) that Horizon prevents this.

POL00029046

11. HORIZON ISSUES – FACILITIES AVAILABLE TO POST OFFICE

11.1 The Issues

1033 In this section, I address Horizon Issues 7, 8, and 10-13, which concern facilities available centrally to Post Office centrally or to Fujitsu, rather than to Subpostmasters.

1034 **Issue 7**: Were Post Office and/or Fujitsu able to access transaction data recorded by Horizon remotely (i.e. not from within a branch)?

1035 **Issue 8**: What transaction data and reporting functions were available through Horizon to Post Office for identifying the occurrence of alleged shortfalls and the causes of alleged shortfalls in branches, including whether they were caused by bugs, errors and/or defects in the Horizon system?

1036 **Issue 10**: Whether the Defendant and/or Fujitsu have had the ability/facility to: (i) insert, inject, edit or delete transaction data or data in branch accounts; (ii) implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts; or (iii) rebuild branch transaction data:

b. at all;

c. without the knowledge of the Subpostmaster in question; and

d. without the consent of the Subpostmaster in question.

1037 **Issue 11**: If they did, did the Horizon system have any permission controls upon the use of the above facility, and did the system maintain a log of such actions and such permission controls?

1038 **Issue 12**: If the Defendant and/or Fujitsu did have such ability, how often was that used, if at all?

1039 **Issue 13**: To what extent did use of any such facility have the potential to affect the reliability of branches' accounting positions?

11.2 Summary of My Opinions

1040 For **Issue 7**, I have interpreted the word 'access' to mean 'read-only access' - because otherwise, issue 7 would be a subset of issue 10. With that interpretation, both PO and Fujitsu had access to a wide variety of transaction data. They need this access for a wide range of purposes.

1041 On Horizon **Issue 8**, the information required to investigate alleged shortfalls is available to PO from several sources. Their perspective is to look into branch accounts from the outside, with no first-hand knowledge of what has occurred from day to day. On the other hand, they look out to their external clients on whose behalf they are brokering business based on those clients' services and products. By virtue of their role in the end-to-end business, PO has access to information not available to SPMs and vice versa.

1042 **Issue 10** relates to both PO and Fujitsu. It comprises three parts, numbered (i) – (iii). The experts are asked to examine each part in three different respects identified as (a) – (c). Therefore, in principle, the issue calls for 18 opinions although they are not all distinct. Therefore, my opinion on Issue 10 has several parts, which are given in section 11.5, and are summarised in a table there

POL00029046

1043 Part (i) of issue 10 is the most complex, and the answer is difficult to summarise in few words. Part (ii) is simpler, in that Fujitsu necessarily had the ability to implement fixes in Horizon, and these fixes necessarily had the potential to affect branch accounts in the future. Similarly, for part (iii) of Issue 10, Fujitsu had the ability to rebuild transaction data, because this was a very necessary part of the robustness countermeasures. It is important to understand that this rebuilding was an automated process, using a redundantly stored copy of the transaction data (RDS), and did not involve discretionary manual rebuilding.

1044 **Issue 11** asks about permission controls and logs of these processes. As Issue 11 follows from Issue 10, my opinions on it must have several parts, which are described in section 11.6. I only note here that in my opinion any alterations of branch transaction data are necessarily subject to the constraint of DEA (for instance, when they propagate to POLSAP) - and any central user who made any such change would leave many traces of his activity, through several kinds of RDS - like footprints in fresh snow.

1045 Issue 12 asks about how often the facilities under issue 10 were used. In section 11.7, I summarise the evidence I have seen on this topic.

1046 For **Issue 13**, I interpret 'extent' as I have interpreted extent for Horizon Issue 1; and I address issue 13 with respect to parts (i), (ii), and (iii) of issue 10.

1047 I ask the questions with reference to the accounts for a specific claimant in a specific month. If a claimant were to assert that the use of any such facility had introduced a discrepancy into his accounts in any specific month, what is the probability of that assertion being correct?

1048 In summary on Horizon Issue 13 applied to changes under issue 10(i) (insert, inject, edit or delete transaction data or data in branch accounts): for these changes to have any significant chance of affecting a claimant's branch accounts in a given month, there would need to be a huge number of them - probably of the order of 1 million. In my opinion, this is not possible.

1049 I also addressed Horizon issue 13, as applied to parts (ii) and(iii) of issue 10. In both respects, the chances of introducing an error in a claimant's branch accounts in a given month are very small indeed - unless there are a very large number of such changes made in error. Details are given in section 11.8.

11.3 Interpretation of the Issues

1050 In approaching issue 7, I need to choose whether the word 'access' applies to read-only access, or to the ability to access and change records. The term can be used in either sense.

1051 As in his para 3.16, Mr. Coyne uses the term 'access' in the sense of access and change. However, it seems to me that Horizon issue 10 addresses the aspect of changing records, and to make issue 7 distinct from issue 10, it needs to refer to 'access' in the read-only sense.

1052 The only problem is that this makes issue 7 rather trivial - and then perhaps a subset of issue 8. Fujitsu were able to access data in this sense; it was essential, for them to be able to support the system.

POL00029046

1053 Issue 8 asks what data and reports were available to Fujitsu to diagnose problems. It largely concerns the robustness countermeasures RDS and MID, and how they operated. So, there is some overlap with section 7 of this report, which discusses those countermeasures.

1054 Issues 11 and 12 all refer to issue 10, so are linked to it. They are largely factual.

1055 Issue 10 (i) refers to the ability to change transaction data and is in my view the main part of issue 10 which needs an extended answer. Issues 11 and 12 then refer to the controls on that ability, and to evidence about its use.

1056 Issue 10 (ii) asks whether Fujitsu could implement fixes in Horizon (which they could) and whether some of those fixes could affect transaction data. Naturally, many types of fix can affect future transaction data, by changing the behaviour of the software. In the overwhelming majority of cases, these changes are beneficial, in making the transaction date more likely to be accurate.

1057 Issue 10 (iii) asks whether Fujitsu could rebuild transaction data. As I understand it, this would only be done in old Horizon as a part of branch hardware changes, with the SPM involved. The word 'rebuild' implies rebuilding from other data stored redundantly elsewhere. This is described in Mr Parker's witness statement at paragraph 18. If his analysis is correct, issue 10 (iii) refers to a technical robustness measure, using RDS to deal with an identified problem - rather than some discretionary change to transaction data. This is consistent with my understanding from the architecture documents I have seen.

1058 Horizon Issue 13 then asks about the extent of the potential impact on branch accounts. As for Horizon issue 1, 'extent' may be measured in two ways - either as the number of distinct incidents, or as their net financial impact. In my opinion, for Issue 13, the second sense is more useful. As I shall describe below, I have attempted to assess the maximum possible financial impact of all the changes listed under Issue 10, on claimant's branch accounts.

11.4 Horizon Issue 7

1059 **Issue 7**: Were Post Office and/or Fujitsu able to access transaction data recorded by Horizon remotely (i.e. not from within a branch)?

1060 In the previous sub-section, I have explained that I am interpreting 'access' as 'access to read'.

1061 'Transaction data' in Horizon derive from the following sources only[118]:

a) Counters, which record each individual exchange of cash and products with branch customers

b) TCs and TAs as discussed in section 6.3.

c) Balancing Transactions as discussed in section 11.6.

1062 All transaction data from every branch is transferred over the Horizon network to central servers managed by Fujitsu, from where it is distributed to other systems used by Post Office. In Horizon Online, Post Office can remotely examine the data held in the Branch Database (BRDB) in read-only

[118] This list is confirmed by the Witness Statement (27 September 2018) of Torstein Godeseth (Fujitsu's Chief Architect), paragraph 17.2.

POL00029046

mode for business reasons, such as monitoring the levels of cash held in branches.[119] Post Office also access data derived from BRDB in systems such as Credence and Horice, In this way, both Fujitsu and PO have been able to read the data remotely.

1063 Fujitsu needs remote access for support purposes.

1064 The previous two paragraphs are consistent with Mr Parker's witness statement.

11.5 Horizon Issue 8

1065 **Issue 8**: What transaction data and reporting functions were available through Horizon to Post Office for identifying the occurrence of alleged shortfalls and the causes of alleged shortfalls in branches, including whether they were caused by bugs, errors and/or defects in the Horizon system?

1066 This issue relates to 'alleged' shortfalls in branches and their causes, if they were caused by Horizon bugs. The use of the word 'alleged' implies that the experts should consider shortfalls reported by SPMs.

1067 As I explained under the previous issue, PO has access to all branch transaction data. It uses this information for several purposes, which include the following:

- to reconcile business transacted at the counters with PO's clients[120];
- to manage their own business by analysing the details of what happens in branches; and
- to assist them in investigating anomalies reported by SPMs.

1068 In sections 4.3 and 5.4 above, I describe Horizon's back-end architecture. The purpose of the Transaction Processing System (TPS) is to gather the transactions taking place in the branches, and to pass them on both to Horizon's back-end systems such as APS and DRS, and to other IT systems in Post Office. PO's systems include Credence and a succession of systems based on SAP, which have culminated in POLSAP.

1069 PO's access to branch transaction data, via a suite of Horizon reports and via their own management information systems (MIS), serves to improve the robustness of the system. Storing the data in PO's systems and cross-checks between those and Horizon contribute to RDS. Because PO uses the data for its own purposes, MID also comes into play. From my experience of MIS, I would expect PO's MIS to include analytical facilities which are not required and so not available to SPMs. PO would also be able to use standard database reporting tools, on demand, to retrieve and analyse information about branch transactions.

1070 As described in section 4.4, Horizon ensures that an accurate record of all transactions is secured in the audit store[121]. When PO is investigating anomalies reported by SPMs, they use Credence and their other

---

[119] Witness Statement – Stephen Parker (Fujitsu's SSC Manager), paragraph 14

[120] See section 6.3 of this report

[121] See section 4.4 of this report

POL00029046

management information systems in the first instance[122] – but, when they need to confirm the transactions handled in a branch, they can also ask Fujitsu to retrieve the corresponding data from audit.

1071 Horizon's systems software generates events whenever something unexpected happens. These events are detected by the System Management Centre[123] and prompt actions, either automatically or manually by operations staff. The events are recorded and can be analysed by support staff investigating anomalies. This information, available to Fujitsu, is shared with PO as required.

1072 Thus, the information required to investigate alleged shortfalls is available to PO from several sources. Their perspective is to look into branch accounts from the outside, with no first-hand knowledge of what has occurred from day to day. On the other hand, they look out to their external clients on whose behalf they are brokering business based on those clients' services and products. By virtue of their role in the end-to-end business, PO has access to information not available to SPMs and vice versa.

11.6 Horizon Issue 10

1073 **Issue 10**: Whether the Defendant and/or Fujitsu have had the ability/facility to: (i) insert, inject, edit or delete transaction data or data in branch accounts; (ii) implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts; or (iii) rebuild branch transaction data:

a. at all;

b. without the knowledge of the Subpostmaster in question; and

c. without the consent of the Subpostmaster in question.

1074 This issue relates to both PO and Fujitsu. It comprises three parts, numbered (i) – (iii). The experts are being asked to examine each part in three different respects identified as (a) – (c). Therefore, in principle, the issue calls for 18 opinions although they are not all distinct.

1075 I have examined the witness statement of Torstein Godeseth[124]. Where it addresses Issue 10, I find it consistent with my understanding of how Horizon works.

1076 In the text that follows, I provide my opinions on each part of the issue. These opinions are summarised near the end of this sub-section.

(i) Insert, inject, edit or delete transaction data or data in branch accounts

1077 In my opinion, 'inject' means the same as 'insert'. Insert/inject and edit/delete are treated as separate cases. '*Transaction data or data in branch accounts*' excludes reference data. I am taking this Issue 10 to include local as well as remote access.

---

[122] The witness statement of Tracy Mather (dated 16 November 2018) is consistent with my understanding.

[123] See Appendix C.7

[124] Dated 27 September 2018

POL00029046

Post Office

TCs and TAs

1078 In paragraph 1061, I describe three classes of transaction. One of those classes was TCs and TAs.

1079 PO effectively uses TCs as the means to introduce new transactions into a branch's accounts (to correct errors) - but not directly. Therefore, in my opinion, this is not a way for PO to inject transactions. SPMs may dispute a TC and ask for further investigation – whereas they would not have this opportunity with transactions that have simply been inserted into their accounts.

1080 Before TCs were introduced in 2005, Error Notices fulfilled a similar function. These were sent to the branch on paper and manually entered into Horizon by a user in the branch.

1081 TAs are used to allow SPMs to accept transactions involving certain PO clients such as Camelot and Paystation into their accounts but, once again, I do not see this as PO inserting transactions.

Global Users

1082 Within Horizon Online, each branch was set up to include a number of user accounts that could be accessed in all branches (for support purposes)[125] – but only from within branches. PO uses these accounts for certain branch operations such as opening /closing branches and training. So-called 'Global Users' correspond to specific roles, such as:

- Engineer – with test capabilities branch diagnostics and maintenance
- Migrate - used to open new branches
- Setup - used by mobile or relief managers
- Auditor - may view users and stock units but not carry out transactions
- Auditor Emergency Manager - used to run branches.

1083 Global User accounts belong to global branches, which are used to administer those users.[126] These branches are assigned to help desk locations, because the administration is carried out by staff based in those locations. Although the global branch is a virtual branch (i.e. there is no physical branch associated with it), there are physical counters that connect to it. These counters form part of the HNG-X help desk. The virtual branch codes are 999999 and 999998.

1084 Because Global Users are given the capability to run branches, in the same way as SPMs, they are also able to inject transactions into the accounts. Any transaction entered by a Global User is included in the transaction log against that user. The usernames start with an asterisk to differentiate them from other users.

1085 SPMs can only add new transactions to their accounts and not change or remove any existing ones. Therefore, Global Users cannot edit or delete transactions either.

---

[125] See ARC/SOL/ARC/0006, HNG-X Architecture - Global Users

[126] See DEV/APP/SPG/0017, HNG-X Counter Business Application Support Guide (version 7.1, 08 January 2014)

POL00029046

1086 These roles are performed within the branch concerned, rather than remotely. Therefore, it is likely the SPM will know that the PO user is on Horizon. They can also check their logs to find out the details of any transactions carried out.

1087 Once again, because Global Users are on Horizon within the branch, they will normally have co-ordinated their work with the SPM. However, they may not require consent as such.

Fujitsu

1088 Branch transactions are stored in a database. In Legacy Horizon, this was known as the Riposte Message Store.

Database administration

1089 Fujitsu's database administrators (DBAs) are the people who build, maintain and tune the Horizon databases.

1090 In my experience, DBAs cannot do this work without the authority to insert, edit and delete data. All databases require such a role even if, in smaller systems, it is not separated from other technical roles.

1091 As DBAs can affect their customers' business data, it is important that their access is strictly controlled, and that DBA access is limited to a small pool of specialists.

1092 DBAs have the ability potentially to damage the integrity of a database for which they are responsible. In my long experience of databases, such an incident may cause them personally great difficulties, even if the database is quickly restored to health. Therefore, DBAs exercise their powers with great caution. Wherever practicable, they build tools that assist them to operate safely and accurately.

1093 Horizon databases are complex structures, which means that any attempt to modify data directly is fraught with risks. Any responsible DBA avoids this at all costs.

1094 In addition to managing DBA access rights, Fujitsu protects its databases with backups and logging of actions performed. Therefore, any attempt to misuse the power would be recorded and any ill effects could be rectified. In any case, in my experience, any such misuse is highly exceptional.

Balancing Transactions

1095 Fujitsu users from the SSC (Software Support Centre) have the ability to inject additional transactions into a branch's accounts in Horizon Online, using a Balancing Transaction (BT).[127] Those users are not able to amend or delete any transactions.

1096 BTs could be used to rectify any erroneous accounting data that may have been recorded as a result of a bug in the Horizon Counter or BAL. They are inserted using the Host BRDB Branch Transaction Correction Tool.[128] This powerful tool could cause serious problems to the Branch Database in the event of certain bugs. The design document states: '*It is expected that only a small number of skilled staff will run this tool*

[127] See DES/APP/HLD/0020 section 5.6.2

[128] See DEV/APP/LLD/0142

POL00029046

and that they will have detailed guidance as to when and how to use the tool.'[129] It also stipulates: '*It will be used manually by SSC (third-line) support*'. Therefore, usage should be limited to a small group of SSC users who could be made fully aware of the risks. Mr Godeseth, in his statement of 27 September 2018, confirms that this has been the case (in paragraph 58.1).

1097 BTs are clearly visible in the transaction reports that are available to SPMs via Horizon as they are stated to have been carried out on counter number 99[130]. BTs do not require acceptance by SPMs, unlike TCs and TAs.

Transaction injection in Legacy Horizon

1098 Mr Godeseth says (in paragraph 58.10 of his witness statement) that, in the original Horizon system, the SSC could also inject transactions, and that those transactions were clearly distinguished from those entered at the branch because they would have included a counter position greater than 32 when no branches would have had such a high number of counters. It accords with my experience that support staff should have a facility like this, so that branch accounts could be corrected in exceptional circumstances – without resorting to DBAs.

1099 Transactions updated in the Riposte Message Store were replicated between the branch counters and the data centre, in both directions. Therefore, any transactions injected by the SSC at the data centre were automatically replicated to the branch. This means that they became visible to the SPM in his reports - with a distinctive counter position, as above.

1100 Thus, SSC users could update branch accounts without the consent of the SPM but not without his knowledge.

1101 However, Mr Godeseth states[131]: '*All accounting at the counter was carried out based on the data held in the message store. The Riposte product managed the message store and it did not allow any message to be updated or deleted*'. If this is correct, the SSC could not edit or delete transaction data. It is consistent with my understanding of the Old Horizon architecture.

1102 Richard Roll, in his witness statement, says the following in paragraph 18: '*we also had the ability to insert transactions and transfer money remotely without the sub-postmaster knowing. Obviously this was not done by me, however I can recall thinking that a third party may have been able to do that if they could have remotely accessed the system in the way that we could (which may or may not have been possible).*'

1103 Mr Roll worked in the SSC and I established above that (during his tenure with Fujitsu) certain SSC users had the ability to transact injections, although these would have become visible to SPMs. So, in my opinion, Mr Roll could not have made these changes to branch accounts '*without the SPM knowing*'.

---

[129] DES/APP/HLD/0020 section 7.2.12.6

[130] Counter 99 is readily identifiable, because it would indicate that there were 99 serving positions in a branch, which no branch has.

[131] In his witness statement dated 27 September 2018

POL00029046

1104 As I stated above, the purpose of the SSC being able to inject transactions was to correct any errors in branch accounts that may have been caused by a system malfunction. I do not see how Fujitsu or any of its staff could stand to benefit in any legitimate way from creating new transactions in SPMs' account, given the countermeasures I describe in section 7, nor why they would wish to do so unless for nefarious purposes.

1105 As for transferring money, Horizon includes no functionality that allows payments to be made to external parties or accounts (that is done by other PO systems). Moreover, in my experience of financial applications, details of payees are always recorded in some application; in this respect, paying any external party will inevitably leave traces of the payment and the party paid - part of the SEC countermeasure) So, even if there had been a functionality in Horizon allowing Mr Roll to procure that payments were made, he could only have '*had the ability*' to do that if he had been able to subvert the SEC countermeasures built into Horizon and other PO applications. The final sentence of the quotation in paragraph 1102 seems to imply that he was speculating about a theoretical possibility and did not know whether or how money could have actually been transferred.

Privileged users

1106 Under Horizon Online, certain Fujitsu staff (Privileged Users) have access privileges that could be used to edit or delete transaction data in the BRDB. This level of access is needed for system maintenance purposes, such as updating database records to help implement planned system changes. However, Fujitsu has no process that requires transaction data to be amended or deleted. Standard Horizon functionality, such as TCs and BTs, can be used to resolve most errors that may affect branch accounts. There is therefore little need to use privileged access to manipulate transaction data to resolve an error.

1107 Any change to a transaction performed by a Privileged User would be visible to branch staff. The amended transaction would appear in reports and logs that can be viewed in branch, although it would not be flagged as a change by a Privileged User. Theoretically this is a problem, but Privileged Users cannot change the audit record and so the changed record in the BRDB would no longer match an audit extract. This means that an SPM could always find out about changes made by SSC, via a request to the helpdesk.

1108 In my experience, Privileged Users on Horizon have the same role as one would expect to see on any IT system. Such rights are necessary to ensure unforeseen events can be addressed if necessary, as a backstop robustness countermeasure (a combination of MID and WOR) - which is typically used only very rarely. Consent is not required from the SPM for any changes in transaction data.

(ii) Implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts

1109 PO cannot make changes in the Horizon software, but they do maintain its reference data, which is a vital component of the system.

1110 I have agreed with Mr Coyne '*that the very nature of rolling out fixes within any IT system, including those implemented by Fujitsu has the potential to affect transaction data or data in branch accounts.*'

POL00029046

1111 Fujitsu implements fixes in software and reference data. They could certainly implement fixes without the knowledge of SPMs. Indeed, such changes are not generally communicated to SPMs – unless exceptional circumstances apply. In my opinion, there would be no purpose in doing so - and the SPMs would not have the background knowledge of Horizon to understand such communications.

1112 In my opinion, any capability to communicate changes to SPMs needs to be used sparingly, to avoid flooding SPMs with information which is of little concern to them in their daily work.

1113 Fixes could also be applied without the consent of SPMs although, once again, changes in reference data that affect specific branches may be co-ordinated with the SPMs in question.

(iii) Rebuild branch transaction data

1114 In paragraph 1057, I give my interpretation of this part of the issue. The word 'rebuild' implies re-creating from other data stored redundantly elsewhere. Thus, this part of the issue refers to a technical robustness measure, rather than some discretionary change to transaction data.

1115 In Legacy Horizon, the transaction database was stored locally within the branch and replicated with central servers using the Riposte product. It was possible for the data in a particular counter at the branch to become inconsistent with replicated copies or 'corrupt'. The latter term means that the data has been damaged so that it can no longer be used. In that situation, Fujitsu could intervene remotely to correct the problem. Mr Parker says that branch transaction data was not changed in any way.[132] This is consistent with my understanding of the architecture and its use of the RDS countermeasure. The workaround (WOR) involved replicating the correct data from another counter in the affected branch or from the data centre copy. The same technique was used to rebuild the counter database if branch hardware was changed.

1116 In Horizon Online, BRDB is maintained centrally and so rebuilding is not needed for hardware changes.

1117 PO has not had the ability to rebuild branch transaction databases.

1118 The SPM could not use Horizon until the branch database is fully operational. In principle, the data could be rebuilt without the knowledge of the SPM in question, but they would be informed or become aware that they could use Horizon normally again and so they would know that something had happened.

1119 Consent is not formally required but, if the data needed to be re-built before the SPM could use Horizon it is unlikely that they would object to this action.

Summary of opinions on Issue 10

1120 The following table summarises my opinions on each part of Issue 10:

	i. Data amendment	**ii.** Fixes	**iii.** Rebuilds
Whether **PO** have had the ability:	• Insert/inject: yes, Global Users have had that ability.	PO can change reference data, which I consider to	No

[132] Mr Parker addresses this in paragraphs 18 and 55 of his witness statement (dated 16 November 2018). His statements are consistent with my own understanding developed from studying Fujitsu's documentation.

POL00029046

	i. Data amendment	ii. Fixes	iii. Rebuilds
a. At all	• Edit/delete: no.	be part of Horizon.	
b. Without the knowledge of the SPM	No	Yes	Not applicable
c. Without the consent of the SPM	Yes	Yes	Not applicable
Whether **Fujitsu** have had the ability: a. At all	• Insert/inject: yes. - HNG: Only via Balancing Transactions (BTs). - Legacy: By SSC. • Edit/delete - HNG: Privileged Users - Legacy: No.	Yes. Applies all fixes in Horizon (software or reference data), which could affect transaction data.	Yes. - HNG: The database is stored centrally where it could also be rebuilt. - Legacy: On branch hardware changes. Also, via Riposte.
b. Without the knowledge of the SPM	No. Any changes performed by Privileged Users become visible at the branch.	Yes	Yes, unless the hardware was being changed, which would have involved the SPM.
c. Without the consent of the SPM	Yes. Neither BTs nor Privileged Users require consent.	Yes	Yes, although hardware may have been changed at the SPM's request.

Table 11.1 - Issue 10 summary of opinions

11.7 Horizon Issue 11

1121 **Issue 11**: If they did, did the Horizon system have any permission controls upon the use of the above facility, and did the system maintain a log of such actions and such permission controls?

1122 This issue refers back to and therefore is linked to Issue 10, which is in three parts that I treat separately.

Issue 10 (i) Insert, inject, edit or delete transaction data or data in branch accounts

1123 PO Global Users are each assigned to a specific Role. Each role is limited to carrying out specific actions - such as logging on, viewing or entering data.[133] In the Legacy system, Global Users' access rights were strictly controlled by "one shot passwords"[134]. In Horizon Online, the method of administration was improved.

1124 Similarly, only a small group of SSC users (30 according to Mr Godeseth) is permitted to create a Balancing Transaction (in Horizon Online). By definition, the number of Privileged Users, who can edit or delete transaction data in BRDB, is also limited (to about 45).

[133] For more information about Horizon's role-based access control, see Appendix C.

[134] See ARC/SOL/ARC/0006, section 2.1

POL00029046

1125 The High Level Design (HLD) document for the BRDB states[135] that '*Support teams will be restricted to accessing the Branch Database only under an MSC.*' I introduce the MSC process in Appendix C.

1126 The same HLD goes on to confirm:

> '*There is a requirement that the SSC will have ability to insert balancing transactions into the persistent objects of the Branch Database. There are reasons for SSC having to do so e.g. to rectify erroneous accounting data that may have been logged as a result of a bug in the Counter / BAL.*
>
> *SSC will have privileges of only inserting balancing / correcting transactions to relevant tables in the database. SSC will not have any privileges to update or delete records in the database.*
>
> *Any writes by the SSC to BRDB must be audited.*'

1127 In Legacy Horizon, the ability to edit/delete transactions was also limited to SSC users.

1128 SSC's access to the counters has been strictly controlled. Mr Parker says:[136]

> '*Some members of the SSC were (and some remain) able to insert transaction data. SSC access privilege gave the ability to inject transactions, but appropriate change controls were in place and no such insertion would have happened without complying with those controls.*'

1129 This is consistent with my understanding of the role of SSC.

1130 Mr Parker continues in paragraphs 21.2 and 21.2:

> '*Any transaction that was inserted would immediately cause a discrepancy to arise in the branch's accounts. For example, if a transaction were to be inserted which stated that £1,000 of stamps had been bought by a customer who paid cash, that would immediately cause a reduction in stock levels of stamps in that branch and the branch would have £1,000 less in cash than Horizon expected it to have.*
>
> *In other words, although a transaction could be inserted, it would immediately become apparent that this had been done and ultimately it would not benefit any member of staff to behave in this way.*'

1131 If Mr Parker's witness statement is correct in this respect, the DEA countermeasure comes into play. In my understanding, there is also another DEA safeguard in place. Branch transaction data have been captured in POLSAP and its predecessors[137], which have been controlled by PO rather than Fujitsu. A subsequent change in the branch database would have led to a discrepancy between that database and the POL MIS, which could be detected later (RDS/MID).

1132 According to Mr Parker, there is no incentive for anyone to inject transactions unless they are required to correct a branch's accounts.

---

[135] DES/APP/HLD/0020, section 5.6

[136] In paragraph 20.2 of his witness statement

[137] I refer to these collectively as the 'POL MIS'.

POL00029046

1133 As the number of users with any given access rights increases, so the risk of unauthorised or inappropriate usage increases. In my opinion, the number of Privileged Users and SSC users who can create a BT seems high. This may indicate an opportunity to improve security by reducing the numbers of permissions granted.

1134 I reached the following agreement with Mr Coyne: '*Usage of the above tools and facilities* [referring to Issue 10] *should be auditable. However, the maintenance of logs would be dependent upon retention periods and size.*'

1135 Each transaction is associated with a particular user, so it is clear in the records who was responsible for its creation. All transactions are recorded in the audit store, so SPMs could in principle find out if any had been performed without their consent or knowledge.

1136 Privileged usage has been logged since July 2015. Prior to then, only log-ons and log-offs were recorded.

1137 In summary, permission to use the facilities described under Issue 10 was controlled. Usage of those permissions and the resulting actions was also recorded. However, the controls in place have not been perfect. External audits[138] have identified room for improvement.

Issue 10 (ii) Implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts

1138 PO controls changes in reference data themselves, outside the scope of Horizon. The data is delivered into Horizon for distribution to branches (see Appendix C.7). I have seen evidence from July 2017 in a document relating to the PO Operations Board[139] that '*After a number of service impacting incidents over the last 3 months, we have now aligned that all Reference Data changes go through the appropriate change process.*'

1139 It could prove misleading to draw broad conclusions from this single piece of evidence, but I deduce from the document that:

- Horizon service was disrupted in the first half of 2017 by problems arising from changes in reference data.
- Not all such changes were being managed by an effective change process.
- Action has been taken to rectify this weakness.

1140 The corrective action would have rendered Horizon more robust by means of the QCC countermeasure - but points to shortcomings in QCC before this.

1141 I have seen no other evidence about permission controls or records relating to PO's management of changes in reference data.

1142 Fujitsu implements fixes in software by installing new versions of specific components in the live system. Reference data is updated by distributing new datasets as described in Appendix C.7.

---

[138] See Appendix H.7

[139] Operations Board 21 July 2017, [POL-0221328]

POL00029046

1143 The Managed Service Change (MSC) process[140] is used to control changes in the live system, new features as well as fixes. MSC forms part of the overall Manage Change process. It is administered via the TfS and Peak tools available on the SSC website and described in Appendix C.7. The tools create detailed records of the actions performed, i.e. the changes in Horizon.

1144 The MSC process document defines roles and accountability for managing changes. The right to use the tools on the SSC site is limited to authorised users.

1145 MSC succeeded the Operational Change Process (OCP)[141] in 2014. OCP had been in place since Horizon was first implemented and fulfilled a similar role to MSC.

Issue 10 (iii) Rebuild branch transaction data

1146 This operation will be restricted to database administrators or, previously, to engineers for when they were upgrading branch hardware and rebuilding the local database from the central servers.

11.8 Horizon Issue 12

1147 **Issue 12**: If the Defendant and/or Fujitsu did have such ability, how often was that used, if at all?

1148 BTs have only been used once, on 11 March 2010 – and this was not in a branch operated by a Claimant. Torstein Godeseth provides details in his Witness Statement dated 27 September 2018. He also states that, so far as he is aware, Fujitsu has never used its privileged access to edit or delete transaction data. I have not seen any other evidence in conflict with Mr Godeseth's account.

1149 PO changes reference data on a daily basis.

1150 A combined total of 36,000 MSC and OCP records have been created, which amounts to 5-10 per business day on average. There have also been some 20,000 release notes, which equates to approximately 5 releases per working day – including reference data.

1151 I have seen no evidence to confirm how often the following capabilities were used:

- Transaction changes
  - PO Global Users
  - SSC
    - Legacy
    - Privileged Users
- Database rebuilds

11.9 Horizon Issue 13

1152 **Issue 13**: To what extent did use of any such facility have the potential to affect the reliability of branches' accounting positions?

1153 I shall interpret 'extent' for Issues 13 as I have interpreted extent for Horizon Issue 1.

---

[140] Described in SVM/SDM/PRO/1184

[141] See CS/PRD/067

POL00029046

1154 I shall ask the question with reference to the accounts for a specific claimant in a specific month. If a claimant were to assert that the use of any such facility had introduced a discrepancy into his accounts in any specific month, what is the probability of that account being correct?

1155 The answer to this question does not depend strongly on the size of the discrepancy, but I shall assume for definiteness that the question refers to a significant discrepancy, greater than £300. As described in section 7.6, this discrepancy is large enough that at least some SPMs, faced with such a discrepancy, would notice it and query its cause. I am not considering 'micro discrepancies' which any SPM might not notice or attribute to human error.

1156 I shall address Horizon issue 13 with respect to issues 10(i), (ii) and (iii)

1157 To answer the question about 'any such facility' for issue 10(i) (insert, inject, edit or delete transaction data or data in branch accounts), it is simplest to answer it for each facility I have identified above, and then to combine the answers. I shall start with Balancing Transactions (BTs).

1158 I have cited evidence that only one BT has been made in the lifetime of Horizon Online. In section 8.5, I calculated that the number of monthly branch accounts in the history of Horizon, across the whole PO network, is in the region of 3 million. - of which, approximately 1.5 million have been in the time of Horizon Online

1159 Therefore, in the absence of further evidence about why BTs might occur, or their frequency, the probability of a BT affecting a claimant's branch accounts in a given month since 2010 is about one part in 1.5 million. The chances of it being incorrect - introducing a discrepancy in the branch accounts - are even smaller. For that to have happened, the BT would need to have been made in error - and furthermore, the error would need to be not detected or corrected. In my opinion, the probability for all these to happening to one branch in one month may be of the order of one in ten million. These figures show that in order of magnitude terms, BTs have a very small chance of affecting any branch in any given month.

1160 (It might be said that since claimants' branches were smaller than the average PO branch by about a factor 3, the probability of a BT affecting a claimant's branch would also be smaller by a further factor 3. But this would depend on the cause of the BT, and whether that cause was size-dependent, so I shall not assume any dependence on branch size).

1161 I next consider changes to reference data. As described above, these were made frequently by PO. Some of these changes were in error, and they may in principle have introduced discrepancies in branch accounts. However, any such issues were recorded in KELs, and I have already analysed these in section 8 - including analysing several KELs specifically about errors in reference data, which had no effect on branch accounts. My conclusions in section 8 about reference data errors were included in my conclusions about all software errors. The probability of their having any effect on a claimant's branch accounts in any given month was extremely small. For details see that section.

1162 I next consider changes to transaction data made by global users, or by SSC. As above, I have seen no evidence about the number of such changes which have been made. However, by the same analysis as

POL00029046

above, for any one such change, the probability of it affecting one claimant's branch accounts in one month are approximately one in three million. The chances of it doing so erroneously, and not being subsequently corrected, are even smaller - perhaps one in ten million.

1163 It follows that there would need to be a very large number of changes to transaction data made by SSC or global users, with a large proportion of these being in error, to give even a 10% chance of introducing a significant discrepancy in a claimant's branch accounts for one month. The number required is more than a million such changes.

1164 In conclusion on Horizon Issue 13 applied to changes under issue 10(i) (insert, inject, edit or delete transaction data or data in branch accounts): for these changes to have any significant chance of affecting a claimant's branch accounts in a given month, there would need to be a huge number of them - probably of the order of 1 million changes. In my opinion, this is not possible,

1165 I next consider the facilities under issue 10(ii) (implement fixes in Horizon that had the potential to affect transaction data or data in branch accounts). Fixes in Horizon only had the potential to introduce discrepancies in branch accounts if those fixes introduced bugs. I have already addressed bugs in section 8. There I found that in order to give a significant chance of introducing a discrepancy in one claimant's branch accounts in one month, there would need to be of the order of 50,000 separate bugs which affected branch accounts on the same scale as the Suspense Account bug. I do not believe it possible for there to be that number of bugs which affect branch accounts, whether introduced by fixes or otherwise. Therefore, the facilities under issue 10(ii) are not capable of introducing discrepancies in claimant's branch accounts, with any significant probability.

1166 I next consider the facilities under issue 10(iii) (rebuilding branch transaction data). I have not seen evidence as to how many times this has been done. Consider one occasion of rebuilding branch transaction data. As before, the probability of this happening to one claimant's branch in a specific month is one part in 3 million. The probability of it happening erroneously, and not being corrected, are smaller than that.

1167 Therefore, the probability of the rebuilding of transaction data introducing a discrepancy in a claimant's branch accounts in a specific month is extremely small - unless the rebuilding of transaction data has been done on a very large number of occasions. As before, more than 10,000 occasions would be required. I have seen no evidence that transaction data were rebuilt on this number of occasions and consider it extremely unlikely to have happened. Therefore, the rebuilding of transaction data would not introduce discrepancies into a claimant's branch accounts in any given month, with any significant probability.

11.10 Mr Coyne's opinions

1168 Mr Coyne's report enables me to identify new areas of possible agreement between us, since our joint statement of 04 September 2018, as follows:

1) The reports available to PO should have enabled them '*to identify the occurrence of alleged shortfalls in the Horizon system (of those that could be identified), and they were underpinned by formal processes which would provide*

POL00029046

further information in relation to the underlying cause of a given issue, and the best way to resolve the same. In addition, Post Office should have been able to obtain any additional information it required via Fujitsu or the Subpostmasters themselves.' [in paragraph 8.10 of his report]

2) '*Subpostmasters had access to a much smaller pool of information. This is in line with what I would expect to see given that Subpostmasters are the users of the Horizon system, and therefore would not typically be given access to anything beyond what was necessary for them to carry out their 'business as usual' activities.*' [paragraph 8.11]

3) Fujitsu has been able to access counters within a branch so that they can provide support and maintenance. [paragraph 9.4]

4) Fujitsu has been able to access transaction data recorded by Horizon both within a branch and stored centrally within BRDB. [paragraph 9.7]

1169 Alongside point (4) agreed above, Mr Coyne also notes (in paragraph 9.6) that '*it has not yet been identified that transaction data was altered at the counter.*' In other words, despite Fujitsu being able to access the counters, Mr Coyne has seen no evidence of any impact on branch accounts.

1170 Mr Coyne criticises PO's controls on reference data in paragraph 4.21 of his report:

'*Despite the criticality of the integrity of Reference Data, a document from July 2017 suggests that changes to Reference Data were not subject to any appropriate change control process. The document [17] reports; "… we have now aligned that all Reference Data changes go through the appropriate change process". This is consistent with the position that prior to July 2017 Reference Data could be changed without any formal consideration as to what the impact might be.*'

1171 The full quoted sentence reads: *'After a number of service impacting incidents over the last 3 months, we have now aligned that all Reference Data changes go through the appropriate change process.'*

1172 In my opinion, Mr Coyne's concerns about reference data are not supported by the evidence:

- The sentence does <u>not</u> say that changes were not subject to any appropriate change control process. Nor does it say that, before July 2017, reference data were changed with no fomal impact assessment. Instead it only implies that, over the previous 3 months, changes in reference data may have impacted service.
- The sentence also confirms that corrective action had been taken.

1173 I discuss the issue fully at paragraphs 1138 onwards.

1174 In paragraph 9.10 of his report, Mr Coyne cites the witness statement of Richard Roll to confirm that Fujitsu employees could and did remotely access branch accounts to perform modifications. SSC could only insert transactions to modify accounts and not edit or delete the transactions. Insertions were strictly controlled, as explained in section 11.7.

1175 From paragraph 9.15 onwards, Mr Coyne addresses the concept of global branches. From paragraph 0 onwards, I explain my understanding of both global users and their branches.

1176 In paragraph 9.18, Mr Coyne states *'An instance of a global branch would allow Fujitsu to create global users and to input transactions within core Horizon systems as though they had been entered from a physical branch*'. In my opinion,

POL00029046

this is incorrect because a Fujitsu global user could only enter a transaction into a branch's accounts when physically present in that branch. Mr Godeseth confirms this understanding in paragraph 32 of his witness statement dated 16 November 2018.

POL00029046

CHARTERIS

12. DECLARATION

1178 I confirm that I understand my duty to the court and have complied with that duty. I am aware of the requirements of Civil Procedure Rules Part 35, the Practice Direction to the Civil Procedure Rules Part 35 and the Guidance for the Instruction of Experts in Civil Claims 2014.

1179 I understand that my duty in providing this report and giving evidence is to help the Court on matters within my expertise, and that this duty overrides any obligation to the party by whom I am engaged or the person who has paid or is liable to pay me. I confirm that I have complied and will continue to comply with this duty. I have not assumed that any particular version of events is true and I have had regard to the case of Imperial Chemicals Limited v Merit Merrell Technology Limited [2017] EWHC 1763 (TCC) in producing my report.

1180 I confirm that I have not entered into any arrangement where the amount or payment of my fees is in any way dependant on the outcome of the case.

1181 Where any examination, measurement, test or experiment has been undertaken for the purposes of this report I have undertaken it myself.

1182 I confirm that I have made clear which facts and matters referred to in this report are within my own knowledge and which are not. Those that are within my own knowledge, I confirm to be true. The opinions I have expressed represent my true and complete professional opinions on the matters to which they refer.

Signed: _____

Dr Robert Worden
Director

Dated: 07 December 2018