FUJ00080690

Document Title: Report on the EPOSS PinICL Task Force

Document Type: Report

Abstract: This document reports on the activities of the EPOSS PinICL Task Force which was in place between 19th August and 18th September 1998 to reduce to manageable levels the EPOSS PinICLs outstanding at that time.

Status: Draft

Distribution: T. Austin M. Bennett
D. McDonnell
Library

Author: J. Holmes D. McDonnell

Comments to: Authors

Comments by: ASAP

FUJ00080690

0 Document control

0.1 Document history

Version	Date	Reason
0.1	18/09/98	Initial draft following Task Force completion
1.0	14/05/01	Raised to v1.0. Administrative catch up.

0.2 Approval authorities

Name	Position	Signature	Date
J. Holmes	Quality & Audit Manager		
T. Austin	Director Systems	N/A	

0.3 Associated documents

	Reference	Vers	Date	Title	Source
[1]	Unreferenced		18/09/98	EPOSS Task Force Briefing Paper	TPA
[2]		3.2		EPOSS Functional Specification	

0.4 Abbreviations

0.5 Changes in this version

FUJ00080690

0.6 Table of content

FUJ00080690

1 Introduction

During the week commencing 17th August the EPOSS /Counter PinICL Stack Reduction Team, known as the Task Force, was established. The objectives, current workload, composition, outline process and targets were presented to the Team on Tuesday 18th with a formal start date of Wednesday 19th August 1998.

This report presents the outcome of the Task Force activity and identifies factors which prevented the original target (zero or near to zero residual PinICLs) being met. During the course of the Task Force it became clear that there are significant deficiencies in the EPOSS product, its code and design, and these are also presented in this report. Finally the report contains recommendations from the authors which we believe should be implemented by the programme to address the shortcomings identified.

2 Scope

The scope of this report is limited to the activities of the EPOSS PinICL Task Force between 19th August to 18th September. It does not consider other PinICL clearance activity taking place elsewhere in the programme.

Although this report is referenced under the Internal Audit project code it is not the result of an audit of the EPOSS Task Force.

3 Management Summary

Before the EPOSS Task Force was initiated the Counter Development Team were immersed in a seemingly impossible task of dealing with PinICLs that were being raised faster than they could be cleared. The Task Force brought about changes in structure and collected together resource from Development, SPTS and T & I into a single coherent unit. It also introduced process changes; the introduction of a 'gatekeeper' to preview PinICLs and target them at the most appropriate person or group; testers who could work alongside developers and with whom consensus could be obtained on a proposed solution before it was 'thrown over the wall'; focused objectives which occasionally caused conflict with other parts of the organisation but helped keep outside interference to a minimum; and shortened communication channels which proved invaluable in turning failure into success in a matter of minutes as opposed to days or weeks.

The Task Force has clearly demonstrated that the deployment of resources at this level and with this structure is what is required immediately and for the long

FUJ00080690

term. A separate report detailing specific recommendations is currently being drafted for consideration within the Systems Directorate.

The EPOSS Task Force was established to address the problem of the escalating number of PinICLs residing in the EPOSS-Dev and Counter-Dev stacks and was planned to operate for the 5 weeks leading to the MOR3 baseline cut on 18th September. The objective was to reduce the PinICL count to zero or low tens by the cut off date and the target set by dividing the current PinICL count by the number of days available. The paper made no concession towards new PinICLs being raised during the period and assumed that the personnel assigned to the exercise would be available 100% of the time and be 100% effective.

The position at 1300hrs on 18th September is that 166 PinICLs have been fixed and closed and 165 remain in WIP. This indicates that the Task Force has failed to meet its primary objective.

However, a review of the Task Force period provides an insight into why it was unable to meet its objective. This management Summary provides an overview of that period and is supported by the main body of the report.

New PinICL Analysis (Sections 5 & 6.1)

Analysis of the PinICL stacks show that since 18th August some 211 new PinICLs have been raised where the product = EPOSS or the assigned team contains 'EPOSS' or the PinICL summary contains 'EPOSS' or 'MiMAN' or MiECCO'.

If the movement of PinICLs between stacks is analysed the results are quite startling as this provides an indication of the number of PinICLs processed by the Task Force.

Note that in measuring movement between stacks a count is made each and every time a PinICL crosses a stack boundary, in or out, and repeats are therefore included.

Stack Name [Numbers as at 1300hrs September 18th]	PinICLs Entering	PinICLs Exiting
EPOSS Pre-Dev [Entry point to Task Force process]	693	664
EPOSS Dev [Holding area for Fix Team work]	580	404
EPOSS Rel [Holding area for link testing]	227	208
EPOSS Post Rel [Holding area for closure testing]	169	157

Team Effectiveness (Sections 6.2 & 6.3)

11 names were identified for the Task Force Fix Team which at 4 weeks duration implied 44 man weeks of effort. The combination of leave, inexperience with EPOSS, attrition and non PinICL work reduced this effort figure to nearer 25.

The Re-work rate between development and link test was ~33% which merely exacerbated the already reduced effort available for fixing PinICLs.

FUJ00080690

The Build Process and Inter-Department Communications (Section 6.4 & 6.5)

The process by which code modules and reference data find their way from a developer's PC to a PIT built rig is lengthy, complicated and prone to error. Each build that was attempted did not pass through the process, end to end, without failing at some point.

The delivery of reference data presented particular problems, especially where multiple baselines are being maintained, and there was obvious confusion between the Task Force, PIT and SPTS regarding responsibilities for the maintenance and delivery of RD in these circumstances and this led to delay in the delivery of MOR2 data.

Communications between the Task Force and PIT did not operate effectively and this introduced unnecessary delay in the build process, especially where the process failed. A further concern is the reliance on Brian Orzel in the process, especially as most build only progressed satisfactorily after personal intervention by Brian.

Additional Functionality and Unplanned Development (Sections 6.6 & 6.7)

Considerable effort was expended in delivering code to support additional functionality required for MOR3. At least one developer was actively involved in this activity throughout the period with extensive support from the Analysis and Test Team. In addition there were instances where the fix to a PinICL required more time and effort than what might be expected to fix a fault. Experienced developers were required for this work and this clearly impacted the PinICL clearance work.

EPOSS Documentation (Section 7.1)

The document suite supporting the EPOSS product consists of three main elements :

a. EPOSS Functional Specification (V3.2) produced in December '97.

b. High Level Design document produced in April '98.

c. Several Low Level Design documents produced in July '98.

All of these were developed by reverse engineering the EPOSS product code at that time.

There are a number of other specifications and associated documentation which also forms part of the EPOSS documentation :

a. Three specification documents (Transfers, Discounts & Balancing) were implemented by the Task Force.

b. A number of detailed problem analysis specifications had to be developed by the Task Force in order to fully understand the problem and how to deal with it.

FUJ00080690

c. >50 Solution Proposals and ~90 Request for Clarifications have been received from POCL following the issue of EPOSS FS V3.2.

EPOSS Code (Section 7.2)

It is clear that senior members of the Task Force are extremely concerned about the quality of code in the EPOSS product. Earlier this year the EPOSS code was re-engineered by Escher and the expectation is that the work carried out in Boston was to a high standard and of good quality. Since then many hundreds of PinICL fixes have been applied to the code and the fear is that code decay will, assuming it hasn't already, cause the product to become unstable. This present a situation where there is no guarantee that a PinICL fix or additional functionality can be made without adversely affect another part of the system.

However, a more worrying concern from the Programme's perspective should be the reliance on the EPOSS product in its current state as a basis for planning and delivery. During the Task Force there was relatively little testing that directly impacted EPOSS and yet >200 PinICLs, roughly 50 per week, were raised. Immediately following the conclusion of the Task Force it is intended to re-run System Test Main Pass and various other test streams. While I am confident that the fixes delivered by the Task Force will prove to be reliable I fully expect the PinICL rate to increase as further testing is carried out.

Lack of code reviews in the development and fix process has resulted in poor workmanship and bad code. Four examples of this are presented in the body of this report and there is no reason to assume that these kinds of problems are not widespread in the product

4 What Was Expected

4.1 Briefing Paper

The briefing paper identified some 280 PinICLs that were to be addressed by the Task Force - 220 from EPOSS Dev and 60 from Counter Dev. It was anticipated that fix work would commence on 24th August and assumed a 7 day working week. This resulted in a required clearance rate of 11/day.

Two teams were established. The Analysis and Test Team (ATT) were to carry out an initial analysis of the problem and provide a 'suggested fix' narrative within the PinICL text. This would enable the Fix Team (FT) to target the solution in the quickest possible time providing a rapid turnaround back to the ATT who would conduct formal unit testing and eventually close the PinICLs on the PIT built ST01 rig. Team members fell into two categories - those that were 100% dedicated to the Task Force and those who had volunteered a portion of their time to help.

Two alternatives to fixing a PinICL were also identified. Closure where the fault could not be re-created or no fault was eventually identified and the Problem

FUJ00080690

Impact Analysis Team (PIAT) where analysis of the PinICL indicated a problem where resolution could be deferred to a later Release.

The final delivery from the Task Force would be a series of Work Packages that could be implemented onto a T & I rig for a full re-run of System Test Main Pass. It was anticipated that zero (or a few) PinICLs would remain to be cleared.

4.2 PinICL Stacks & Process

A special series of PinICL stacks was generated to support the Task Force lifecycle :

a. EPOSS Pre-Dev : Entry funnel to lifecycle and analysis point.

b. EPOSS Dev : Holding stack while fix activity underway.

c. EPOSS Rel : Holding stack post fix but pre link test.

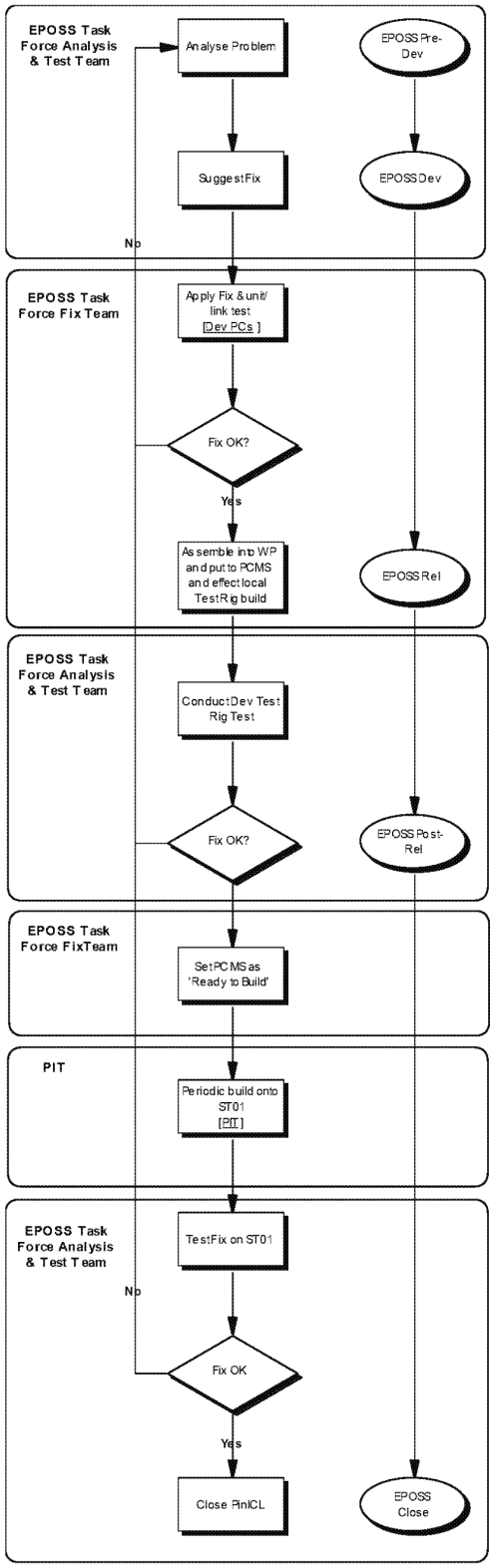
d. EPOSS Post-Rel : Holding stack post link test but pre closure cycle.

e. EPOSS Close : Holding stack for all Task Force 'fixed & closed'.

f. EPOSS Susp : Holding stack for PIAT deferred and other PinICLs for later review.

FUJ00080690

The outline process described in the briefing paper was completed and notified to team members.



FUJ00080690

5 What Was Achieved

The success or otherwise of the Task Force can be measured from a number of perspectives. The original target of 'zero or low tens' of residual PinICLs has not been met so from that perspective the initiative could be considered a failure. On the other hand, it is generally accepted that the Task Force approach, where a self contained team follows the PinICL from analysis through fix to close, has been shown to be more effective and provide greater job satisfaction for those involved.

[NB : The following statistics are taken from the PinICL database and are based on stack events, ie if a PinICL enters or exits a stack an event has occurred and the event count is increased. This means that PinICLs that cycle between stacks will be counted as many times as they cross the stack boundary and should only be used as indicative of]

From a pure numbers perspective the following achievements were made by the Task Force between 19th August and 18th September :

a. ~660 PinICLs entered the process funnel in EPOSS Pre-Dev. This number includes repeats where a PinICL may have been delivered back following a re-route action.

b. ~540 PinICLs entered EPOSS Dev. This number is inflated by the reworks resulting from the 32% unit test attrition rate. (~150 remain on the stack @ 1430hrs 17th September)

c. 114 PinICLs have been closed in EPOSS Close or by the EDSC.

d. 38 PinICLs were closed in EPOSS Pre-Dev following unsuccessful attempts to re-create the problem.

e. 30 PinICLs are awaiting unit or ST01 closure test.

f. 14 PinICLs have been deferred by PIAT to NR2+ and are located in EPOSS Susp.

In terms of non-PinICL work the following significant pieces of work were delivered :

a. Discounts functionality.

b. Transfers functionality.

c. Stock Unit and Office Balancing functionality.

d. Re-written Suspense Account functionality for Mi-Mann.

e. Significant design work around Pick Lists and Session Transfers.

FUJ00080690

6 What Was Found

6.1 PinICL Numbers

On the 19th August the total number of PinICLs distributed between EPOSS Pre-Dev and EPOSS Dev was 331. A review of these identified that 21% had been open for >3 months and that there had been a significant increase in the rate of PinICL being raised in the 3 months up to 19th August.

	EPOSS Pre-Dev			EPOSS Dev		
Raised in	**No Raised**	**Age**	**Number**	**No Raised**	**Age**	**Number**
August 98	101	@ 1 month	311	11	@ 1 month	20
July 98	54	> 1 month	210	5	> 1 month	9
June 98	90	> 2 months	156	1	> 2 months	4
May 98	30	> 3 months	66	2	> 3 months	3
April 98	23	> 4 months	36			
March 98	2	> 5 month	13			
February 98	2	> 6 months	11			
December 97	3	> 8 months	9			
November 97	1	> 9 months	6			
August 97	2	> 12 months	5			
July 97				1	>13 months	1
May 97	1	> 15 months	3			
April 97	1	> 16 months	2			
December 96	1	> 20 months	1			

During the Task Force period a further ~211 PinICLs were raised and deposited in EPOSS Pre-Dev, these being a combination of MOR1, E2E, New Task Force, re-assignments from other stacks and other unidentified sources. Thus a total of ~508 PinICLs were input to the Task Force process during the four week period.

It is by no means clear why so many PinICLs remained unresolved for such a long time. Based on the PinICLs themselves the December '96 entry (PC0001033) was cleared during the Task Force period but took some 20 manhours of effort, the April '97 entry (PC0002757) was transferred to Escher and the May '97 entry (PC0003404) was closed in an hour.

FUJ00080690

6.2 Team Competence & Availability

The briefing paper identified some 11 names in the Rapid Reaction Team. This would suggest ~44 manweeks of effort available. However, of the 11 names 3 were new to EPOSS and within this group 3 weeks was lost due to leave. Of the remainder, 1 was unavailable for the whole period due to the development of new/changed functionality, 1 was unavailable for two weeks due to leave and 1 left after three weeks involvement. Assuming 50% effectiveness for the new comers the actual manweek effort available was nearer 25.

This reduction was further exacerbated by poor quality workmanship from some of the more experienced team members as evidence by an average 33% reject rate from unit test and the failure of every build due to missing RD or code .dlls.

6.3 Development Re-Work Rates

During the Task Force period there were 5 formal Drops of code and reference data to unit test and ultimately PIT. On average the reject rate for PinICLs from unit test was 32%. The effect of these was to re-cycle the PinICL back into EPOSS-Dev and for the developer to spend further time re-correcting his work.

PinICLs	Unit Test	ST01 Test
Untested	1	12
Tested	128	96
Failed	41 (32%)	5 (5%)
Passed	87	91
New Raised	6	5

{Latest figures to be supplied}

Although the 5% failure rate during the ST01 closure cycle is unwelcome it falls within acceptable limits.

6.4 The Build and Delivery Process

The ~30% attrition rate experienced at unit test only occurs after a build of a unit test rig has been made. This in itself is a complex process with sufficient opportunities to fail to ensure that of 5 builds made during the Task Force period each one failed for one reason or another.

Essentially the build process consists of the following steps - the associated 'what could go wrong' situation is presented in italics immediately underneath each step :

FUJ00080690

a. Reference Data is validated through Phil Hemmingway and delivered by the developer into the RD Work Packet established and supervised by Dave McDonnell. Due to the amount of support Phil provides to other RD users there is a known 12 hour lead time for this.

Developer does not provide sufficient time for the validation exercise and either misses a Drop deadline or attempt to meet it and delivers potentially incompatible RD to the WP.

b. Developer delivers code *.dlls to a separate but associated Work Packet established and supervised by Dave McDonnell.

Developer mixes RD and code in wrong WPs.

Developer does not deliver the RD associated with the code at the same time.

c. The WPs are delivered to PCMS for collection by either Brian Orzel's or PIT's rig build process.

WPs incorrectly delivered to PCMS.

d. Brian extracts the WPs from PCMS and incorporates them into his pre-defined unit test rig build process.

WPs incorrectly incorporated into unit test rig build script.

e. The rig is built by the SPTS member assigned to the Task Force.

Finger trouble while building.

f. Once the WP has been unit tested it is marked as 'Ready-For-Build' in PCMS ready for PIT to action.

Items in WP not correctly marked.

WPs not correctly marked.

PIT not notified of WP availability.

Task Force not notified when the build goes wrong.

Task Force not notified when the build works OK.

Most of the 'what could go wrong' scenarios were experienced by the Task Force and while some were of our own making the very complexity of the process, coupled with the urgency of the work in hand, mitigates against a trouble free transition.

6.5 Communications

Communication between departments was inadequate in two particular areas, Task Force with PIT and Task Force with SPTS/RDMC.

The build process outlined above identifies a number of areas where effective communication between the Task Force and PIT is required to ensure that the

FUJ00080690

ST01 rig build progressed without undue delay. Each build was delayed more than necessary by a basic failure of both groups to communicate effectively with each other during the build process.

While the SPTS breakdown was not directly related to EPOSS PinICLs it does have an impact on the team members and the work they have to do. The issue here is to do with the delivery of Reference Data to SPTS for incorporation onto a rig, specifically the 'NRG' files. Counter Dev provide such 'NRD' files as are required by SPTS, including 'D' type and Escher data that is generic to all Post Offices, and 'C' type data which is passed to RDMC for distribution. Associated with an 'NRD' file is an 'NRG' file (which Counter Dev used to supply) but as this was identical in each case an agreement was reached earlier in the year whereby PIT/SPTS scripts would be changed to use the same 'NRG' file each time. However, MOR2 requires a different 'NRG' file to bound the scope of testing and SPTS expected Counter Dev to provide this data. While the Counter Dev team can and do provide generic RD they do not provide any special purpose RD that may be required to bound the scope of a particular test. Historically this has been provided by T & I, presumably by the manager responsible for that phase of testing.

The problem here was to do with an SPTS expectation not being met by Counter Dev and remaining unresolved until it had escalated into an MOR2 threatening situation. The confusion was exacerbated by the emerging role of the RDMC in the testing cycles and it is clear that the interfaces between Counter Dev, the RDMC and SPTS have not been agreed nor have the responsibilities of each of those groups in the supporting Reference Data.

6.6 Additional Functionality

Although it was anticipated that the Task Force would be addressing PinICLs for MOR3 there were three pieces of work that were being developed for incorporation into the MOR3 baseline :

a. Discounts.

b. Transfers.

c. Stock Unit and Office Balancing.

As the most experienced member of the EPOSS team, John Warwick was assigned to develop the code for these functions. This removed him 100% from PinICL fixing. This work was tested in isolation by the Analysis and Test Team (ie. PinICLs were not raised) and resulted in two 3 page reports of bugs and deficiencies. These created sufficient concern within the Team that Vin Patel was assigned to work directly with John and supervise his activities. However, Vin's lack of EPOSS business knowledge meant that Steve Warwick had to be assigned on Day2 of Week4 to personally supervise John's work.

FUJ00080690

6.7 Off-Plan Development

There is an expectation that fixing a fault, while perhaps taking some time to track down the cause of failure, should be a relatively straightforward activity. There were examples during the Task Force where the PinICL resulted in significant re-design as well as coding changes. An example of this was the handling of Suspense Accounts by Mi Man.

In June/July of this year the way that Suspense Accounts was handled by EPOSS was changed. A consequence of this change, which was CP'd according to the Change Control process, was that Mi Man would have to reflect these changes to ensure that migration would work. The link between these two activities, Andrew Morgan, left the project and the work in Mi Man was not done until a PinICL was raised against it. It is estimated that ~40% of the Mi Man code has had to be re-written to accommodate the changes.

7 The EPOSS Product

7.1 Documentation

7.1.1 Documentation Suite

The EPOSS product was originally developed using RAD techniques as part of the Joint Working Agreement in force during Release 1. This approach carries a number of attendant risks, not least of which is the lack of formal specification. During 1997 the product was sent to Escher for significant re-work as the solution arrived at via RAD was deemed not to provide sufficient integrity.

In July 1997 the product was passed across to Escher for the implementation of a solution to the issue of how to maintain the integrity of the accounting data on a distributed system. The original proposal for control of this aspect of the system was to implement the use of a Stock Unit Smart Card which was required to be present in the keyboard during certain key events (logon/logoff, SU balancing etc). The object of the card being to allow the software to determine whether or not all the relevant data for the required activity was present on the node on which the activity was taking place.

The Smart Card solution was rejected both by Alan Ward and Escher on the grounds that it relied on data which was recorded and stored outside of the control of the Riposte environment. The solution which Escher proposed, and which was implemented by November 1997, was to use Riposte Markers to delineate accounting periods. This solution provided predictable and repeatable recording of data within the marked periods but required significant further application development within EPOSS to apply the POCL business

FUJ00080690

rules required to deal with data potentially isolated on nodes which were not connected at the time a balancing or reporting activity was in progress.

The returned product was then reverse documented and V3.2 of the EPOSS Functional Specification produced in December '97. This was put out for review when POCL objected to the level (not enough) of detail in the document and the fact that both generic desktop and specific EPOSS functionality was included. It was then agreed during March/April '98 that both parties would work together to understand the level of detail required. Chris Plunkett would document the result with Graham Seedell's (POCL) help and the result validated (and constrained) by Steve Warwick. The result was V3.3, minus desktop functionality and with extra detail, but not yet agreed and subject to further change.

During April '98 an EPOSS High Level Design document was reverse engineered from the code and circulated for internal review. This document is not consistent with the EPOSS Functional Specification.

Corresponding Low Level Design documents were developed during July '98 by ISTL, again reverse engineered from the code although they were not made consistent with the HLD.

7.1.2 POCL's Involvement

POCL had also identified three major gaps in the EPOSS product, namely Discounts, Transfers and Stock Unit and Office Balancing - referred to as the '3 papers' - and these were required for implementation into EPOSS. Although not introduced via the Change Control process, specifications were developed and code delivered during the Task Force for the MOR3 baseline. The specification content has been introduced into V3.3 although subsequent reductions in scope made during the Task Force have not been factored in.

A third issue raised by POCL was the manner in which the proposed functionality had been presented in the specification. Whereas V3.2 described EPOSS on the basis of the 'accounting cycle', POCL wanted it to reflect their business processes. The result was that POCL were invited to develop 'Solution Proposals' which, if acceptable, would be factored into V3.3 to provide the level of detail requested by POCL. To date some 57 Solution Proposals have been presented by POCL although only 6 have been reviewed and passed for inclusion in the specification.

The final area of difference revolved around the EPOSS Issues List which contained hundreds of 'issues' and had become unmanageable. This was replaced by the 'Request For Clarification' process taken from the original Joint Working Agreement. To date some 90 RFCs have been received from POCL.

7.1.3 Other Pathway Generated Documentation

During the Task Force considerable effort was expended in understanding the root cause of some key areas including Transaction and Event Log handling,

FUJ00080690

Pensions and Counter Revenue. Substantial specification style documentation was developed to support the analysis and these should be considered for inclusion in the final EPOSS Functional Specification.

7.2 Reports

There were three problems consistently encountered with Reports.

a. Non or partially populated.

b. Arithmetically inaccurate.

c. Not conforming to Specification.

These error types could be mixed in any combination and had to be addressed by differing mechanisms. Non or partially populated Reports was usually a Reference Data problem and could be addressed through that medium, the arithmetic inaccuracies could be addressed in code. The non conformance aspects presented a different problem. The current specification for these items, BA/POCL Reports and Receipts v2.5, actually targets NR2+ for the full delivery which meant that that element of the PinICL had to be referred to PIAT for deferral.

7.3 Existing Code

[NB : This section has been produced with the assistance of Dave McDonnell and Martin Smith and their combined experience of structured programming]

Although parts of the EPOSS code are well written, significant sections are a combination of poor technical design, bad programming and ill-thought out bug fixes. The negative impact of these factors will continue and spread as long as the PINICL fixing culture continues. This is partly due to the nature/size of the bug-fixing task and partly due to the quality and professionalism of certain individuals within the team. The problem is probably best illustrated examples :

Example 1 :

This extract from EPOSSCore.dll has been written to reverse the sign of a number and is equivalent to the command :-

d=-d

```
Public Function ReverseSign(d)
If d < 0 Then
    d = Abs(d)
  Else
    d = d - (d * 2)
  End If
ReverseSign = d
End Function
```

FUJ00080690

Whoever wrote this code clearly has no understanding of elementary mathematics or the most basic rules of programming.

FUJ00080690

Example 2 : Unreachable Code and Bad Practice

This extract from EPOSSStockUnit.dll :-

```
If lstockrootnode = 3013 Or lstockrootnode = 3016 Then
    bremedprods = False
    intbalancerootlevel = 5
    lbalancerootenode = 3017
    If lstockrootnode = 2493 Then
      bremedprods = False
      intbalancerootlevel = 3
      lbalancerootenode = 3006
    End If
  Else
     bremedprods = True
     intbalancerootlevel = 5
     lbalancerootenode = 3017
End If
```

The three shaded lines are unreachable code.

'initbalancerootlevel = 5' is set regardless and should be outside the IF statement and is an example of lazy coding.

Nodes hard coded.

Example 3 : Poor Workmanship and Patchwork PinICL

```
If s <> "" Then
  Do
      If s <> "" Then
      {(}Significant body of code removed to save report space{)}
          Exit Do
      End If
    Loop

  End If
 Next
End If
```

The DO WHILE loop should be a WHILE DO loop and is a further example of poorly structured code.

FUJ00080690

Example 4 : Hard Coding

```
If Not bremedprods Then
        stxn = stxn & ObjMake(TXN_PRIMARYMAPPINGS, sPrimaryMappings)
        ' hz 14/7/98 add the suspense container identifier
        sCAMapping = getPersistentObject("CAMappings", ObjAttributeValue(s, "NodeName"))
        Do While ObjAttributeValue(sCAMapping, "Data.Leaf") <> ""
          Select Case Val(ObjAttributeValue(sCAMapping, "Data.Leaf.N"))
            Case 99995026, 99995027, 99995028, 99995029, _
              99995030, 99995031, 99995032, 99995033, _
              99995046, 99995056, 99995057, 99995058, _
              99995059, 99995060, 99995061, 99995062, _
              99995063, 99995064, 99995065, 99995066
                stxn = stxn & ObjMake("SuspenseContainer", "$$")
                Exit Do
            Case Else
            End Select
          sCAMapping = ObjAttributeComplement(sCAMapping, "Data.Leaf")
        Loop
      End If
```

The above is an example of hard coding which may have been originally made for good reason but there is no evidence of review to remove.