File name: POA Improvements - Overview-20210805 - Sandie Bothick session 2
Client: Morrison & Foerster (UK) LLP
Job ID: UK0155456
Date Due: August 7, 2024
Transcript by TransPerfect

STEVE BROWELL: Just waiting for the prompt to confirm. Okay. So let's make a start. So a few all hands ago, it could have been a month or two, I forget, I shared that there were a series of workstreams being run and that the basis for those was having recently completed some Horizon audit work for the post office and having responded to questions from the public inquiry and a variety of other questions coming in from our legal teams and other areas. Building the answers identified some areas that warranted a bit more thought, that we were perhaps struggling to articulate ourselves well because of certain inconsistencies or people shared that actually things weren't quite working the way that we all thought was best. So we just made a note of them at the time, proceeded to respond, and then captured them all and pulled them into a series of workstreams. Well, there were a variety of observations made, not all of which turned out to be accurate. There were comments made, you know, at the time, but what we did is by using these work streams, in actual fact, I'm only going to talk about four of them here, we then pulled together relevant leaders from the different parts of the business or key contributors and just discussed the points until we had confirmed some things that felt they needed to be either changed, shored up or stopped.

And the real focus here is on TFS, TFSNow, the system, and some of the processes used to interact with it. Peak, the system, and some of the processes used to interact with it, processes and integrations between the two, because, for certainly, for incident management, they are used in tandem. And then looking at ways in which we could use information and processes within the system to then drive some of the important meetings we run, such BIF, customer BIF, and Horizon defects, so that we were more consistent with more reliable data and confidence that we could compare week on week, month or month, etcetera. And the real game here is to make sure we're working in a way that works for us, that we are increasingly system and process driven, not individual and personal style driven, and that we get much better management reporting, and consequently are more confident sharing information with post office. So we pulled together a list of findings and ideas that were either process or system related into team. So it was released in test as well as development, as well as third line support, as well as Mac. Everybody was touched by all of this. It's all been written up into a document that is quite extensive. You might have seen it. It doesn't matter if you haven't at this stage, cause I'm going to summarize the key points here, but that document is kind of a working document. When we finish these streams of work, there will be a new document published, because the main focus for all of this was around management of defects and the progression of things. But, equally, you'll find local documents are being updated relevant to your areas, like incident and problem and duty management and so on. So that what we should be doing is starting to embed all of the working notes that we've produced into standard documents for the teams.

So you shouldn't have to read something separate. It should be something that has been formalized and recorded and probably added to dimensions where appropriate. So some of the motivations here, the emphasis, was about process agreement and consistency, because in the teams and across the teams there were various places at which one team had finished its work and done a great job, but, unfortunately, it didn't lend itself to feeding straight into the next team. So we had some process gaps, or we had some ways of working that created a problem for the next person or that we could have solved if the first person had done it slightly differently. So we

wanted process agreement and consistency, so that we were working in a way that suited us, and obviously working in a way hopefully that was better. Equally our use of TFSNow and Peak, because of time served on the account, various people were using it in ways that they'd become accustomed to, and that was okay until we've now decided that we want to provide management views of it. So we needed to be a bit more specific about what matters in each of the systems, what fields are important, so that we got, again, more consistency with the tooling.

00:04:54.8

And then there was confidence in the data, how reliable is it, so we can track and validate it. And, unfortunately, quite a lot of the entries and certainly the metadata, so not the actual heart of the investigative work and the findings that people produced, but the bits around it, made it difficult to be confident we knew where something was up to in a process, whether it had progressed from last time. You really needed to start reading all of the detail, having some expertise yourself to derive it. So we wanted to get the data into a state that was more reliable, so we could provide tracking and validation, which means, therefore, we get better management reporting, which means those in charge of areas, teams, functions or processes can confidently see what's going on and make good judgments about whether it's working or not and what should be done. And then we've got areas for improvement. There have been many in this conversation, because you spot gaps, you spot things that don't seem to make sense, and by talking them through with everybody, we can identify more areas for improvement, so that we can more confidently articulate how we work. But we wanted more confidence in the post office reporting. We know it's right.

Therefore we're confident to share it, instead of always thinking, oh, hang on, I can't send that, because that needs checking, or I'm not sure that's right, or that's not really been updated, and I'll need to check something. We wanted to get away from that, so that we can just extract the information and present it to the customer where they need to see it. We're not suggesting that they'll see everything. And then the byproduct of that is we can reasonably expect post office to be more responsible and involved in the things they should be aware of. They want more transparency, they want more visibility, and that's okay. But the consequence of that is we therefore expect them to pay attention to the content and join us in some of the discussions and decisions, which they have, you know, been able to avoid in recent years because they haven't had the information, and therefore have chickened out of joining in. So we wanted to get that sorted, because that's an increasing post office interest. They want more information. They want to be more involved, and I guess we want to help them do that with information we can rely on.

So let's be clear, this is we're talking about Belfast HNG-X. We are not talking about the post office cloud at this time. That will come later. And the goal was to evolve what we've got, not to revolutionize. So to work with Peak and TFSNow, to work with the way our teams and organization is structured and then make some improvements, rather than do anything more radical. I don't think it was needed. There isn't time to get to that point, and it isn't, you know, necessary. So it was evolution, not revolution that we went for. And the arrows at the bottom are not a new process. They are how we already work. But I draw the picture, because what we were

keen to be able to understand is from the minute somebody raises a concern by means of an incident, so it is a potential live defect, and we're talking about the live system here, the thing we're responsible for contractually delivering, that the postmaster uses, the post office limited head office uses. That's what we're really worried about, that version of the system. You get a potential live defect with the TFS incident or an investigation peak is raised. We then use our skills and expertise, investigate it, qualify it, might decide there's no action needed. We may decide we need to raise a change. We may decide to make, you know, escalate it to a problem, but eventually we're going to have to decide on a course of action, and that might be to initiate a fix. That point we have a confirmed live defect. This is not as it should be, and we need to do something about it, and then on it goes to development and architecture to identify the solution scope, the size of it, into business impact forum for review and approval, off to customer business impact forum, if necessary, for post office engagement in the decision making. Then it goes into the Peak targeting forum for targeting to a release out to dev ops for the actual work to be done, into test for the actual work to be checked, and then through release to live. The goal in all of this in terms of process and data changes was to be confident I can see or we can see where something is on that continuum at any point, we can see where the bottlenecks are, we can see where things are slower, we can see where things have stopped, and we can actually optimize the things that we know we could do better. And that was the goal. And it was very difficult until we made some changes recently, to be confident where things were on that continuum. We're traversing two systems and probably six or seven teams in order to get to the outcome to make this happen.

00:09:56.7

So there's a few new phrases come into our world, a live defect. This is not a test defect. It's not a development defect live. So it's present on a live system. It is deployed somewhere, the condition. And so the thing we're experiencing is present on a live system. It is or it appears to be inconsistent with the agreed design or service specification. So if it was never meant to do that, that is not a live defect. That is a new feature. But if it was supposed to do that or we think it should have done that, it is, in fact, inconsistent with something we've agreed by way of design or a service delivery specification, and therefore it's a fault that's likely to need fixing. So if we find something, if it is raised as an incident in TFS or Peak and we classify it in this way, it needs some new attributes adding. In TFSNow, it's a configuration item called LiveAffectingDefect, and in Peak, it's a collection called ##LiveAffectingDefect. So we can quickly spot those things which meet those three sub bullets. And it doesn't matter if there's a workaround in place, it still needs fixing. It is still not right. The fact that we've come up with a way round it is great, cause it limits the impact, but that does not mean you don't classify it that way. So that's the one new phrase, live defect. The next one is there is a Horizon defects review meeting, which post office chairs. We used to chair it. It's been in place for about three years, and we have since confirmed some definitions with the customer. Therefore if we have live defects, so they're on the live system, that meet any of these three sub bullets, we need to add another marker, because we have to disclose these to the customer and review them weekly. They are not only shared with the post office, but post office share them with postmasters.

File name: POA Improvements - Overview-20210805 - Sandie Bothick session 2
Client: Morrison & Foerster (UK) LLP
Job ID: UK0155456
Date Due: August 7, 2024
Transcript by TransPerfect

And we were asked quite a few times during the inquiry, what are the defects? Which ones do post office know about? Which ones do the postmasters know about? This is the forum that addresses that concern. So if we have a live defect that affects or has the potential to affect the branch financial outcomes, so they will see balance discrepancies, for example, then we have to add either a HDR-Fin collection or the HDR-CI, configuration item, if it's TFSNow. And if it could affect the way the postmaster interacts with the system or the things they use from it, so the user interface or reporting or functions, or, equally, if a customer, you and I, if we're just going into a branch, could be--would experience something, we have to give it the HDR-Exp experience marker. And those two tags, HDR-Fin, HDR Experience, allow us to classify postmaster branch affecting conditions.

Now, there are various checks going on. Your gut will probably tell you if this is affecting things in that way, but as it moves along the support chain, other people will possibly gain more information, and it may become apparent later in the stage, rather than right at the beginning. But we are bound to identify those things that affect the postmaster and branch and to mark them accordingly, so we can share them. And, again, it doesn't matter if there's a workaround. It's great if there is. We are still duty bound to share them. Even if it's not confirmed, it's suspected, we are required to share it at the earliest opportunity, so that post office is aware and we can work together to either confirm it is or isn't a defect and to then progress to fixing it. And that's important, because it is now a matter of mutual obligation, probably contract when we commit it to print. And all this just stitches the two together. So in some of the instructions you may have been shared and asked to help, you'll have seen this, cause we have been going through an exercise of looking at every single ticket we've got open in TFS or Peak and looking to see if it should be carrying the double hash or the LiveAffectingDefect marker and any of the HDR markers, so that we can now interrogate the system and see where are our live defects, which ones are affecting branch outcomes. So they're two new pieces of phrasing. There are some others.So aside from the Horizon defects review, which is a meeting as well as a tagging system, we have investigation peaks. They are effectively like TFSNow incidents raised internally where we're looking into something or linked to a TFS incident and we're just investigating. We do not know what the cause is yet. We are still making sense of it.

00:14:56.1

The rule, though, is if we are working on something that the customer should be aware of, whether we've only logged it in Peak or we've only logged it in TFSNow, we have to find a way to bond it, so that the customer can see it. We cannot work independent of post office knowledge on faults that we think they should be aware of. That's our decision, what they should be aware of. But the duty is it should have a TFS incident and probably a service now incident, and it's bonded and the updates are flowing, cause that's the mechanism by which we keep them briefed. A defect Peak is different to the investigation Peak, which is the one where we're just working out what's going on. As soon as we know what's up, then we need to create a defect Peak. And that's effectively a fixed ticket that we're going to progress through all of our processes to live deployment. I have a picture of this in a moment. We'll always have potential live defects, so things we think are a live defect, not just something we're investigating that's being worked on.

File name: POA Improvements - Overview-20210805 - Sandie Bothick session 2
Client: Morrison & Foerster (UK) LLP
Job ID: UK0155456
Date Due: August 7, 2024
Transcript by TransPerfect

And then there'll be confirmed ones, confirmed defects. And there's a distinction there, because, obviously, if it's potential is not real, we can't fix it yet. If it's confirmed, then we can fix it, and we can put all the plans in place to move it along that continuum. OTI, if ever you see it written, is the integration between TFS and Peak, largely happens behind the scenes. And KBA, knowledge base article, KEL is a term we wish to extinguish. It's not a fair acronym for what's in that platform. There are lots of hints and tips and guidance, support tools, you know, how to's, what to's, if you see this, explain it like this. There's lots of hints and tips. It's not just errors. So it was some time ago renamed as the knowledge base, which is a very normal support function repository, but we wish to remove any use of KEL and migrate now to it's a knowledge base article and in the knowledge base, just for consistency of language for ourselves. So if I draw a picture, and let's see if this helps bring some of that to life, the customer can raise an incident. If they want Fujitsu's involvement, they bond it. That causes an incident to be raised on our platform and allows us to share updates that replicate across the two platforms.

If in progressing the TFSNow incident, we need access to third and fourth line, then we assign it to a Peak resolve group, and that creates a new Peak in the system, which those in third and fourth line will then continue to work with, and that replicates not everything, but nearly everything across back to TFSNow, meaning it can go back to the customer. So what we've got, it doesn't matter which system you're working on. Hopefully the relevant updates are flowing between them automatically. The customer is kept up to date. Our front line team are up to date, and the back end team can keep working on whatever they're actually investigating and looking into. That's happening today. That's not new. However, when the investigation work is completed here and they have confirmed they know what the fault is and there is something we need to fix, at that point, they clone that Peak. The clone carries a different call type, which is a defect identified. That new Peak's reference will be added into this Peak's last entry. And this Peak will be closed. That allows this incident to be closed. That allows post office to close its incident too, cause it will have the reference to the defects that we will now be managing through our process. This defect Peak is not linked to anything. The updates we apply to this one are internal and private. There are certain fields we will use for reporting purposes, but this is our Peak progressing through our fixing process.

Now, another scenario might be that, I don't know, the test teams or dev architecture have created a Peak, because they've spotted something that they want to look into. If it is something that the customer should be aware of, any work on this should stop. They need to notify our own people to get a TFS incident raised, which is bonded. It then needs to be escalated out to third or fourth line. And then this Peak, the Peak created that's linked to the incident, needs to be maintained, and this Peak must go. We cannot be working on incidents which the customer should be aware of that are not linked to TFS and not bonded. That would be considered not acceptable now. The spirit of transparency is we keep the customer aware and updated. Now, obviously, we may have created the Peak and it doesn't really need to be notified to the customer. It's something we're looking into, some error message, we think that's kind of weird. Why is our platform doing that? So this is very much for us.

File name: POA Improvements - Overview-20210805 - Sandie Bothick session 2
Client: Morrison & Foerster (UK) LLP
Job ID: UK0155456
Date Due: August 7, 2024
Transcript by TransPerfect

00:20:05.5

It's okay that that does not have a TFS incident. That is for us to investigate and look into. If that is the case, if it turns out that it really is a defect and it does need something fixing, you do not need to clone it, because there's only us can see it. You can just change its call type. But, again, nothing, none of these defect Peaks are linked to, by automation anyway, any TFS incident or service now incident. We look after these ourselves internally, and there are reporting mechanisms to make sure that we can, if needed, brief the customer on the status of any or all of them if we need to. So we're severing the link. Once we've determined a fix is needed, then it's in our systems and managed by us. But while we're investigating, we keep the customer briefed by using the replication in the system. Okay. So I'll pick out a few highlights, if you're a TFS user, and I think, yes, you are. Some of you certainly are. Some big things here. We've already said it, if we're working on something that POL need to be notified of, there must be a log TFS incident that's bonded. A Peak only is not correct anymore. The other thing we've been doing, and it's not new, we've been doing it for a while now, is to remove references to knowledge base articles and peaks within the content of the TFS ticket updates, not the associated references, which allows us to see whether--what the actual peak is that it's replicating content from. But we do not, because we wish to discourage, if not remove, the need for post office to want a copy of a KBA or a Peak. That is our system used by our people for our purpose. TFSNow and it's linked to service now is the visibility that post offices should have. So if there are parts of a KBA that we feel need to be shared, put it in the body of the content. If there are parts of a Peak that need to be shared, another Peak, not the one we're actually updating and synchronizing, put it in the body of the content so that the TFSNow incident is an integral reference to everything relevant that the customer needs to see. Now, if it's not a bonded incident, we can do what we like, cause it's our ticket, our systems, and therefore we can work freely. But if it's bonded, then we must make sure that the TFS incident itself is complete and does not require someone to ask for other content in order to get the best picture. We're not hiding content. We're not denying access to content. We're simply making sure that the bit that's necessary is in the incident record and shared with the customer, and the things that are perhaps internal, so guidance notes, what priority to give it, which team to escalate it to, all of those kind of things, how we actually fix it, SQL commands and queries or any other rectification is not of interest to the customer. That's ours. That does not need to be included.

Okay. The summary field, the short description field. Sometimes it's just system generated, particularly for SMC type tickets where they see system events. But the ask here is if we can think about the wording of that, so that it is a better problem statement, if you will, that describes the incident that most people would understand, that will help, because that field is one that we will be sharing when we run reports, with our own management, our own people and the customer. So if it doesn't make sense, someone's going to ask a follow up question. If it does make sense, they probably won't. So whenever we can, the ask would be to make it as clear and articulate as you feel it can be. Also keen to discourage the concept of separate emails to share progress. If there is some further information that someone has that's relevant to an incident that's open, add it to the incident itself, and then the update can be I've already updated the incident to reflect the information I'm about to share. So you could actually just go and have a look, if

you're post office, cause it may be bonded. I want us to try and discourage side assessments and additional information provided by email that is not in the body of the incident. I think we do mostly, but I think we should do it as a matter of policy. This bullet here is more of a health warning. People like me are looking at this now, and I don't understand everything about the system and how you all work. I know it may make sense to you to look at some of the tickets and updates, but I don't always understand. So think about the fact that less qualified individuals may be reading content, and, therefore, you know, if you can just add a few more words to make something that is quite brief a little bit more rounded, we may understand it better.

00:25:01.1

And the other one is it is sometimes incredibly difficult to get a succinct view of where we're up to. You have to read the end to end incident update content, whether that be in Peak or TFS, but you have to read it, and then you have to be knowledgeable enough to join it together and come to a conclusion. And even sometimes when you do that, you're wrong, because there's something else somebody else knew. So the ask is, if we're providing additional comments, the customer visible bit, I know we can put internal notes in, let's think does this help the reader, the customer, understand where we're at, without having to reread everything that's happened before? For the sake of a few more sentences, we may well be able to say, the latest status is this and here's what we're going to do next, rather than assume that they're going to read everything previous and be able to make sense of it. It would be nice if you didn't have to read back more than two or three updates to get a rough idea of what's going on. Now, that's personal, manual. I can't do it with any automation. So it's just an ask, please consider writing a description that helps somebody really understand what's going on. Other bits in for TFS, there's categories and subcategory fields. I don't think we normally change them, but if you do, and it's bonded, it breaks, so just be careful and don't do it. There are certain open and closed categories which we've been using, but I'd like us to look at using them a bit more frequently, so we can get better management reporting of the kind of things that we're seeing and the areas of our, you know, the system and application that seem to be the most problematic or the slowest to deal with or any other attributes that we can actually see.

So it's increasingly important that the open and close categories and codes that we've got are used, so that we can look at TFS and make better decisions about what's happening. I think Sandy and Steve [PH 00:27:07] Bancel, at least, are looking at which ones of those might be useful and what reports they can pull. And then we've set up these configuration items for LiveAffectingDefect and Horizon defect review experience under financial, and those are important, particularly the H Horizon defect review ones, because we report them to the customer. And there's another field that's in the system state, which can be acknowledged, work in progress, researching, fixing progress, yeah. And others. It's important that that is an accurate reflection of the state it's in, because that gives us an idea where it is on that arrow continuum. We're still looking into it. We know what it is, and we're fixing it. And eventually it's either been suspended, resolved closed, whatever the state value we've set. That's important, cause it tells us where it is on the continuum. And although it's sometimes used, it's not always maintained accurately. So that would be good. And we're going to do some work, Sandy's looking at this,

about how long we let things sit in a suspend state, where we have no real interest in the outcome. So we're doing it because POL don't want to close it. I think we have every right to put some rules in place that would allow us to clear down some of those Qs. If it's suspended cause we want the information and we're waiting, that's quite different. So we're looking at that, and we'll communicate that to the customer when we're ready. And I'm hoping to see some of the local work instructions for the teams updated, as well as some of the more formal documents, to bring some of these items more into the content that you rely on to do what you do. And that's a quick summary of those things that affect TFS itself. I can pause briefly if anyone who's more TFS proficient has a reaction to that. That's okay if you don't. I realize it's all new. It may still be sinking in, but obviously I'll share some of this content after. I think some of you certainly use Peak. Is that a fair comment?

MALE 1: Yeah.

STEVE BROWELL: Okay. So I'll continue with some of the things that have changed with the way we use Peak. So firstly, there are some new fields. They weren't there before. There's a field allowing us to capture the POL problem reference. That's the reference that the post office are holding for what fit these Horizon defects. So it is very few of the tickets in the system are incidents and defects in the system that will carry this marker, but we have the ability to hold it.

00:30:01.2

And it just helps when we pull reports, we can actually show a post office reference, which means that they have no excuse for not understanding what the entry relates to. And we can also make sure that if they start quoting other references, and we can't see them, we can say, wait a minute, where did that come from? We're not sure what that means. We added a Fujitsu problem reference field into Peak, excuse me, as well, to allow Matt to be able to associate overtly the Peaks to problem records that he's working with. Again, as those things progress along, then it's possible then to say, hang on a minute, we've got a problem linked to this Peak. So if we take an action, we should be notifying Matt, rather than waiting for him to see. So all that becomes possible because of the fact that we've captured the association and the reference number. Another new field, work around, we've always thought, is there a workaround to this? And if there was one, we've implemented it, and we've added the text into the body of the update. However, unfortunately, I can't query that. So we added a field. So if we do have a workaround in place and there is one, then we set the value to yes. Excuse me. So that means we can see that perhaps there is a little more time being bought to fix it, or the impact has been mitigated with the customer, even if we are still progressing the fix. So that's really useful, cause that helps us, I guess, apply a severity or impact view to it all. So that's a new field.

And then the release management tab probably won't affect any of you in terms of needing to update it. You probably won't even need to look at it, but just so you know what's changed, the tab, which is the top right here when you're in Peak, has been expanded. This is much bigger than it was, and it's been done because the BIF, the CBIF, and the PTF meetings are now held. The outcome of them is held in here, not in separate minutes. It's in the Peak. So we can see what

happened at BIF or these other meetings, and we can see the dates they happened. The updates to this will predominantly be done by people like [PH 00:32:10] Mick Reynolds, Adam Harney and James Guy. So if you see it, that's all. You will not be expected to fill it in. That'll be done by the people who lead those forums. But we have set some specific criteria that determine whether or not this Peak needs customer input before we take it further into targeting. So rather than it being a discretionary thing that some of the attendees would have come to, there are now a series of questions that will be asked at BIF, by the chair of the BIF meeting, to test whether or not the customer should be notified, and we can capture it, record it, and then we know what happened. So that's kind of FYI rather than action for anyone on this call. So there are some new field values in Peak. So we have added a new call type. There was already the A2, and there's a Z underneath the bottom of that one, which isn't really used, but we've added a new one, hash. We'd run out of alphabetic letters, so we had to use special characters.

But this is for those defect Peaks. We have confirmed that there is a fault that needs to be fixed, and it is a defect, so it shouldn't be doing this. We give it the hash marker. And those are the ones that we drive through the process of BIF, customer BIF, PTF, development testing release, and obviously, eventually, closure and live. So there is a new call type to be set when a Peak has moved to the point where we have confirmed the cause. It might be a clone Peak. If it was something that was linked to TFS originally, we will clone it to break the association, but it may well just get that marker because it was raised internally and we just changed its state. So that's a new field value. The collections, there are a few new ones. These hash hash live ones, which we talked about and the HDR ones, again, are important to add. If you are not sure and don't add them, but the ticket or the incident or defect makes its way to BIF, they'll you'll be asked at that forum to confirm. So there are various points at which we may check in case people have forgotten or wrongly tagged it, and so we can fix it. And those are--there's a collections--it's not on my screenshot here, but there's a collections tab at the top. And then you can, somewhere around the center of the screen, you can pick the collection, and you click add it, and it and adds it onto the peak, which means we can query them and see them a lot better. It is a manual activity. It is based on you remembering what a live defect is, and it is based on you remembering what a horizon defect candidate is. But there are various other sweeps of the system constantly happening that look at other attributes on Peaks to determine if it looks like they should have been set. So there are various cross checks going on just in case you've forgot or tagged it incorrectly. So just do your best, and then the other systems and processes should hopefully ensure that we don't miss it or we tidy up any inaccuracies.

00:35:12.7

So renewed importance of Peak fields. So now we're running reports and sharing content, it really becomes important that some of the fields that perhaps--well, they've already been--they've all always been there, but we haven't necessarily made--applied the rigor to the value within them. So the call type does matter. If it's a call type hash, I know it's confirmed and we should be fixing it. It's a defect. If it's not, it's something we're still looking into. So if we're actually trying to fix something we're still looking into, we have an inconsistency, and therefore we're not declaring the data correctly. So this field is important, particularly its use of the new tag or the

new value, hash, which is for the defect identified. The summary field is also important. You'll see why when I show you an extract in a minute. That's typically inherited from TFS for Peaks that are linked, but, equally, if it's created new in Peak, same comment as for TFS, the more understandable it is for most readers, the better. And if we can change it, then we should. There's also a field in Peak called impact, and it used to have problem statement, implication if fixed, implication if not fixed. We've changed that to have three sub elements. So at a template level, if you click into the field and remove any of the data, or it's already empty, as soon as you start to complete it, these fields will appear. They're only really important for those Peaks that we've tagged as being Horizon defect review, which is those we are going to share with the customer. Everything else we're not worrying about for now.

And to put that into scale, we've probably got about a 125 live defects in the system today. 10 of them are HDR tagged, so it's not many. Only 10 of them are we caring about the precision of the way it's described. And if I just zoom in on one, as an example, the better worded they are, the less likely we are to get questions, and the more confidence we instill when we share it with post office. You may not need to write all of this. And if the updates to the body of the content are good, it's very easy for someone to pull this together. And it will typically be myself, Sandy, Adam or Steve Bancel who attend the post office meeting that will make sure that this is worded in a way that puts us in the best light and avoids unnecessary follow on questions. But if any effort has already been made, we'll use what's already there and build on it to make it even clearer for the customer. And you'll see why on the next screen, cause I'll show you what we share with them. Priority field is important, because obviously that determines a number of disclosure thing. So if it's priorities As and B, we treat them obviously differently to Cs and Ds. The mapping is priority one, [INDISCERNIBLE 00:38:09] Sev1, 2, I think, in TFS to Sev3, 4. The team it's assigned to, cause that's who we're going to chase for updates and the product group and product value's a bit like open and closed categories in TFS and associated CIs. This helps us understand what are the problematic areas of our solution and which one seems to take the longest and are there any trends in growths or, you know, declines in faults recorded in different areas? All of this is helpful to get really good data about what's going on. This is an example of the content we will share with the customer, and we do--this is an example from one recently shared for one of the Horizon defect review meetings.

We share the call reference, which is the defect Peak number, the summary field, whether the workaround has been set, and that business impact field or the impact field, and that's all they get. They do not get a copy of the Peak. They do not get a copy of any related knowledge base articles that we may have written. They equally do not get a copy, in this case, of the problem record. They simply get the latest status update for their purposes of their tracking. And this is simply to show them that we are still looking after this. It is making progress, but we know where it's up to. And therefore here is your update, post office. And that's what we share. Hence the importance of knowing, certainly, the summary field makes sense. The workaround, if there is one, is recorded. And the three step update for the impact is recorded. As you can see, INC72 FAD--I have absolutely no idea what that's trying to describe, just from that summary. So there's an example of something which I think probably was raised by the customer. So I guess what we get their description.

File name: POA Improvements - Overview-20210805 - Sandie Bothick session 2
Client: Morrison & Foerster (UK) LLP
Job ID: UK0155456
Date Due: August 7, 2024
Transcript by TransPerfect

00:40:03.5

But unfortunately, for people like me, I'm reading it, thinking, I have no idea what that means. I can read this and make sense of it. We wrote that. But you can see the challenge. If you're just going to make a conclusion from just this summary of what you think it means, it isn't always helpful. Hence the reason if we can change it, that would be really helpful. A couple of other points, there's a field called root cause. It is used, but not particularly consistently. It should be able to help us understand what is causing the issues and the incidents and defects that we're dealing with. So the encouragement for Peak users is please set a value that's correct and appropriate. Don't ignore the field. But there are certainly some root causes, which is it's the user knowledge or a user procedure or the user themselves, which I will treat as no fault found, as in there's nothing wrong with the way the system is operating. Even if we thought it was a defect, even if we thought it was branch affecting, we can eventually close it out and put that clause on it and say, actually, after investigation, we confirmed it's not us. It's you, if you will. And response category is a bit like closed codes, closed categories for Peak. It determines progress through from being opened, pending, final and closed. But when it gets to final and we're approaching closing it out, there are certain values that matter in the new world. All of the other ones can still be used. These are not new. They're existing. But if one of these is used, it has an important meaning. So if we put final program approved, no fix required, that's because possibly a customer BIF, we've decided that or the customers decided I don't want to do anything about that. I'm happy to leave it where it is.

We also get some of those where, you know, the fix is quite light, but all these others fit into what I will call a no fault found classification. So I will ignore them when we're trying to share the scale of defects we're dealing with, and the ones warranting fixed, they will be taken off the list. There are some target release values. I don't know if you ever use those, it's probably mostly release management, but there were a couple that were never really used, so we we're going to remove them just for clarity and simplicity. These are the no fault found categories in Peak. I know that the bottom one is not a no fault found. It's a no further work required by Peak resolver groups. It could well require Unix, Network Security, [PH 00:42:41] Wind Tell. So it could be that Peak have said not with us. They route it back. The TFS ticket can then be reassigned to a different resolver group. So those actually are not no fault found, but we'll pull reports off TFS now in order to determine those. These are all described in the application support strategy document, which has got that number, which describes our first, second, third and fourth line incident processes, particularly for application support. So that document already describes these. I've just picked out those that we have got specific filters on now, and that really matter if that value is set.

Finally then, the BIF meeting. I don't know how many of you--you probably know of it, but you may not attend it. There is a BIF action flag that can be set when a developer is ready for it to be discussed for approval. Equally the relevant scans of the system, if it looks like a Peak is stuck, if it's unclear why, having identified the defect has yet to be presented at BIF, the system, or me, may automatically set the flag to force the discussion to stop the position where it's stuck with

development and we are unclear what's going to happen next. So sometimes the flag will be set for the person to cause a discussion at one of our forums to determine what's the delay, what's the problem. You've seen on that screen, that moving from BIF to customer BIF is now a matter of ticking some answers to some questions. It's not discretionary. It's system driven based on data selections. We capture ManDays, but we don't use that to determine if the customer should be notified. It doesn't matter if it's a day's work or 20 days' work. If it's a defect that needs to be fixed, the volume of work is not is not the customer's concern. It's ours, frankly, in terms of scheduling. So we still capture it, cause it helps us see what we're dealing with, but we don't use it to determine what we take to the customer. And the other thing we're doing with the customer BIF, there've been no submissions in the last three weeks to the forum, but, when we do, we'll be using a mini proposal form, like you would for a change work order, a CWO, because the customer is being presented with some information, and we're expecting a decision.

00:45:05.6

And we can't do that by just chatting to them about the scenario and then saying, so what do you want to do then? I think we need to present the information succinctly and invite a proper response. It's not a formal document in the sense that it's got a number and it's in dimensions, but it is held in Peak. So if we are taking a set of circumstances to the customer for a decision, we're going to capture them succinctly, it's on your page, and say, now based on that information only, what is your decision? At least then we know why they made the call or on what data they made the call. The other thing is because of that release management tab, now we can capture the dates of meetings and the outcome of discussions about Peaks within Peak itself, removing the need for separate minutes to be created. And because actually the data we care about is in the system. And the Horizon defects meeting, which is the critical one, which is customer visible for the significant faults, is also driven by the system because of the tags. That HDR-Tag that's added will allow us to pull the report off the system. Problems are also tagged with that. Incidents are also tagged with that. So we can pull all the information and say, right, what should we be sharing with the customer and what is the update, if we've already shared it, to reassure them that we are still progressing it to a conclusion? So then again, system driven, not a matter of remembering, or we should mention that one at the meeting. It's driven by what's in the system.

If it doesn't have the tag, it's not going to the meeting. If it does have the tag, it will probably go to the meeting. And I only say probably, because sometimes you read it and go that is perhaps not needed. We should change the classification. Because this is, there is a degree of human interaction still needed, because it isn't system derived, its system extracted. And that impact field is really important with business impacts, status update, and next action, because it really helps all of us understand where are we. And the great thing is having got to this point, particularly for this meeting, the Horizon defects review, our confidence and rigor around our way of working and the data in our system means we can say to post office, can you please make sure that reference links to this one? Why have you added another one on the list but not told us about it? Why are you saying that the update is this when, in actual fact, the one we just shared with you is different? So we can start to be more firm about our stance and the way things are working. And that's important, because we wish to disclose, but we're not having post office

File name: POA Improvements - Overview-20210805 - Sandie Bothick session 2
Client: Morrison & Foerster (UK) LLP
Job ID: UK0155456
Date Due: August 7, 2024
Transcript by TransPerfect

dictate our ways of working or tell us what the status of something is, because we should know it. I already showed you that one. So the place we're at, we're explaining the changes. We've already talked about it with some of the team leads, and they may have already cascaded some of this information, but this discussion is part of explaining the changes. We have been doing a lot of work to make alterations to existing TFSNow and Peak incident and defect entries to bring them in line with our new use of the data for reporting purposes. We're nearly finished.

A lot of people have helped get a lot of new collections and configuration items added. They've changed the value of state fields and so on to make them more accurate, so we can rely on the data. So we've been doing a lot of data cleaning onetime activity. What we now need to start to do is to embed this into standard operating procedures, local documents, work instructions, processes, procedures, whatever it is that determines how we work. We've got to build it in, so that it's not something that's recorded separately. And then we got to challenge our teams and people to work these new ways. It doesn't work if we carry on doing what we always did. We'll be doing data cleansing and cross checks all day long. So we need to move into slightly new ways of working. Most of it's about understanding process rather than the system. The system changes are not that's extensive. It's getting your head around what a defect Peak is, what an investigation Peak is, why this configuration items matter. But I think if we can get through that in the next week to two weeks maximum, then it's really about helping the management get a better view of what's going on, help them help you, as individuals who work on tickets, to say, hang on. Is that right? Are you sure that should be set? We need to get to that point, so that we're all helping each other keep the data clean and the processes consistent. And I think we will find all sorts of process blocks. We've found a handful in the work we've done already, but we are going to find some that will help us work out how we hand off from team to team to get better visibility of what's going on and make sure that that continuum, that left to right movement is as fast and efficient as possible.

00:50:05.9

And then more management reporting. Dan, Wendy want to know key facts, not everything, how we're getting on with our incidents, what's growing, what's declining, where are our problem areas, how many defects have we got, are we managing them to a conclusion, and so on and so on. How many have we declared to the customer in the Horizon defect review meeting, all of this stuff now needs to be built into standard management packs, and it will. And, of course, we need to start to encourage or use more post office reporting from our systems, not our people and emails, so that they equally see more of what's going on, or perhaps, hopefully initially, much more confident that we're doing the things that they thought we were doing, but, equally, they can engage with us and join in the decision making, because they're better informed. So that's where we're at in terms of the process. So the screen's just gone black. That was my whistle stop tour of things in TFS and Peak, things that are process related and how all of that stitches together for feeding meetings, management reporting internally, customer reporting externally. So I will invite--I know that went quick. There was a lot of content. I'll share this and the supporting document and the recording, but if there are questions at this point, then please ask.

13

File name: POA Improvements - Overview-20210805 - Sandie Bothick session 2
Client: Morrison & Foerster (UK) LLP
Job ID: UK0155456
Date Due: August 7, 2024
Transcript by TransPerfect

MALE 2: I've got a quick one, I've got a call at 12. But I'm just trying to think scenarios, and I think one of the first slides kind of mentioned, you know, an impact to branch, potentially. So, for example, we've got some routers in a third party, the DVLA's data center, for example. If we lose connection to that router, and, you know, with hindsight, let's say it's the WAN link to that router, is that a live defect, or is that just an incident? You know, I think, you know, we would handle it the same and all that kind of stuff. I'm just thinking do we classify that as a live defect? Because, you know, it is affecting live. You know, branches wouldn't be able to do anything relating to DVLA in the branch. But it's essentially a, you know, a brake fix. It's not a deviation from a design or anything like that.

STEVE BROWELL: So it's a great question, and I'm thinking on my feet to the answer. I think in that scenario, we probably have a major incident on our hands. But you've picked an extreme one, I think so.

MALE 2: Well, yeah.

STEVE BROWELL: And there are probably others that are slightly lower down, which are similar but not quite there. You know, there's no harm done if you want to mark it as a live defect, because it needs a fix. Now, the fix might just be that the link's down and it'll come back up eventually. And because, I don't know, it's third party or it's ET or something, so it'll probably arrive and disappear very, very quickly. But if possible that that would be a live defect, I guess it depends if it's in our scope of obligation, cause if it's BT's lines down, that's not a fault in anything we're doing, but it's something we're going to manage. It is just an incident, then. But it's where the defect for me is when it's something where it shouldn't be in that state from us. Andy Hemingway asked one. He said what happens if a primary system fails but the secondary kicks in? Technically, that's not right. But no one's affected. In that scenario, no, it wouldn't be a live defect. That's just the technology misbehaved, and we need to work out why and sort it out. If the cause was something we've done wrong, then that is a defect, and we need to fix it. But if that's just, you know, stuff sometimes flaps, stops working, then that's just an incident for me. We're looking at things that need a fix to something the way we've done it, it's not just a, I don't know, the machine's hot, so we just turn the air con on a bit harder, you know, that kind of.

MALE 2: So in another scenario, then, for instance, we are upgrading or, you know, we're upgrading the firmware, software or whatever on a device within Belfast under a change control, something within that firm, where either the firmware goes wrong or, you know, we've made an error in that upgrade, and the following day we come in and whatever, you know, something is not working. That's branch impacting. So I think that would probably be classed as a defect, then, wouldn't it?

STEVE BROWELL: If it's branch affecting, I'd just encourage--yeah, true. I think, for me, in the event of doubt, assume the worst. I'd rather talk it down than talk it up and say we've missed it. So for the avoidance of doubt, if you're not sure, assume it is, because at least then we can, you know, we can disclose and manage our way down.

File name: POA Improvements - Overview-20210805 - Sandie Bothick session 2
Client: Morrison & Foerster (UK) LLP
Job ID: UK0155456
Date Due: August 7, 2024
Transcript by TransPerfect

00:55:12.4

MALE 2: And we can add that LiveAffectingDefect CI after an incident's been raised, and, you know, if we kind of were--

STEVE BROWELL: You can, and we have.

MALE 2: --20 minutes to progressing that incident and suddenly decide, actually, this probably does fit that model, then we can just add that in, can't we?

STEVE BROWELL: You can. Absolutely, yes.

MALE 3: Steve, using Chris's example there, I think, obviously, if, say, doing like, an upgrade to a device and basically the plan has followed and maybe, you know, something was missed, something like that, it has caused an issue, I wouldn't classify it as a defect. But I agree with you, if they put the tag on there, there's no harm, at least then we know it basically has some impact in live. If, for instance, say the supplier's have given us an IOS update and for some reason there's actually an issue, a bug basically that has been unforeseen by the vendor when we were applied, it has caused a live affecting incident, then I would call that as a defect.

STEVE BROWELL: Yes. Yeah. I think if it happens during a change window and it's just spotted as part of the routine monitoring and you then fix the thing, that's not particularly a defect. That's a planned situation, isn't it? That you are managing yourself through. It's only if you think that, hang on a minute, why did that happen? That's wrong. That's where we're trying to get to, just to be honest with ourselves. That needs a fix, and that fix needs to be properly scheduled and dealt with as fast as we possibly can, so the system is back into normal state, as we promised it would be, you know, not the wider obligations, which might involve Verizon and BT and DVLA and whoever else you may well be mucking around. That's not ours to fix, but we would want to track it.

MALE 3: Sorry, I've got to go, I've got a customer.

STEVE BROWELL: Me too. That's fine. I understand. I'll stop the recording. And yeah, anyone thinks of anything afterwards, you think I don't get that, that doesn't make sense, that's impractical, please share it. There may well be some logic failures here that we need to think through as we start to make the changes.

MALE 3: Thanks.