SJ84999 Interview Transcript (redacted)

AUTHOR: Ok, so thinking about your role as Team Leader Software, could you start to describe the safety activities that you carry out.

SJ8499: Quite diverse I suppose. Um, I suppose constant throughout my whole career has been the, um, the needs who would do to, um, software development processes that made a particular standard, um, on XXX for the time we've been working on that for the last thirty (years). Is we, we, um, where it's a Defence Standard developing software to, um, predominantly XXX. Um, so obviously…our processes, you know, the, um, the level of testing etc. that we did, we specify things (that end up) producing the safety case. Um, when I moved into a world where we're trying to meet DO 178 instead, um, but again, still still a lot of processes involved, um, the thing that we did on XXX is we kind of started over the last 20 years or so, to, to, move more towards the failure analysis focus when it comes to software. So not only did we continue to follow the, um, the development process, and the needs to make the particular series. We also started to introduce, um, more of an active, um, move towards the, um, failure analysis on the software, um, and that sort of time on the projects, we, we, improved those processes to add more sort of recurring detail to what we were doing. Um, so I guess that's it in a nutshell.

AUTHOR: Ok that's reasonable, so from what you've said is it reasonable to state that the starting point would be your software team getting a set of safety requirements?

SJ84999: No.

AUTHOR: Ok…?

SJ84999: Um, yes and no, um. On the whole project it's very, it's not a sort of 'them and us' approach to systems and software. Um, and in fact, it's sort of like, was the software engineers who develop the software safety requirements, and it was pretty much for most of the time is was needed that way, or people working for me (needed it) that way. Um, so you saw some of this slight overlap between systems and software, so it didn't just sort of sit there and receive software safety requirements. We, um, were actively involved in developing them.

AUTHOR: Ok, thanks, could you talk about how they were developed?

SJ84999: Alright, um, well we changed the processes basically. I think I could give you some documentation on that.

AUTHOR: Yeah - forgot to mention, as we want this discussion to be independent, I'm deliberately not sharing the map of your company processes.

SJ849999: Ok got you. If you go back to the source - more than 20 years ago, we didn't really have software safety requirements, as the process was, um, tech stack functions with SIL level. We basically just developed all software to the same SIL. We were creating software that was, um, one monolithic build with no partitioning, no separate components, so everything had to be developed at the same SIL level. Um, and we just developed to the highest SIL, unfortunately…no real safety requirements, um.

AUTHOR: Sounds expensive.

SJ84999: Um. Not really no, not really, um, everything was developed to SIL 3, and, um that was it. No, no operating system on board, um, quite a relatively small system in comparison to what it is nowadays. Then XXX came along, so, um, a decision was made to introduce an operating system. So to allow us to, um, to host third party software, and, um, to try to improve the re-usability of software, that kind of thing. Um, so at that stage, they, um, they, there was all sorts of improvements that the staff needed to make, but possibly following on from, um, from where you are - at a University - that sort of introduced the idea of software safety requirements. I guess that now we would be creating software safety requirements, and checking that we've met them, and also the idea of checking our partners. Software, and developing it at different SIL levels, then meant you needed to identify things in a more rigorous way. Um, so, um, so the processes we sort of changed to introduce a software safety requirements process, and that was first developed by the people working at York. Um, and I don't know whether it's the correct term to use, at the time people referred to SHARD analysis.

AUTHOR: SHARD, Yes..

SJ84999: Yeah, using, you know, key guide words to, um, look at interfaces into things and see what sort of failures you could get. And what sort of protection mechanism we needed to protect against them, um, and at the time we didn't really use an operating system, you know, and it was known that this operating system to, um, ensure safe partitioning that kind of thing, if we wanted to make software of different, um SILs, um, so the first thing that was done with, uh, uh, SHARD analysis was done on the operating system/platform to ensure that it met safety requirements regarding, you know, correct scheduling, and safe partitioning etc. So that was done following a process, and that was based around one developed by York University, and that still holds. There's a, there's an analysis being performed on that. It gets maintained when changes are made to the software platform and, and it forms part of our Safety Case. At the time, um, the SHARD was done. Well, I want to wait, um, I was also sort of recognising that they, that as a whole, we need to recognise that the SHARD was applied to the, um Mission Computer, to, um identify software safety requirements that, that needed to be met by the application software. Um, and this, this changed, uh, a whole load of requirements that needed to be met, and this work was too late for people, I think. Um, they got people such as myself in trying to understand, you know, what the functionality, what the system needed for how that might, um, you know, for the information that we provided when SHE was to develop these safety requirements. And, um, so once underway it was pretty much the case that the safety requirements that were developed, were just sort of a re-expression of the normal functional requirements that the software already needed to meet. So they were a bit of a waste of time, right?

AUTHOR: Okay...

SJ84999: Um, basic, basically, all that was really done with them, was that there was a sort of, of a trace...which tested them. So it was argued that these things have been tested by the tests, by and large. We'd already tested these things because they're already part of your normal requirements set. The safety requirements process just seemed to involve re-expressing what was already written down anyway. Anyway that was something on Mk 1, and these, these requirements just dealt with the software as a whole, you know. It didn't look at failures of the software itself. It looked at failures, failures external to the software - that the software might need to cope with, you know, and, and, the flow-down requirements regarding say what the software needed to do, if a particular input was lost. Yeah, um, it didn't really go any further in terms of analysing what the software itself could fail - the contribution to failures, um. So, anyway, that was that, it was on its way and that sort of got parked for a few years, um. When the next project came along. I think was the, um, one of the next projects that came along was the upgrade for the XXX aircraft, with LIFTCAP capability insertions, or something like that. It was, they, they basically won't set the XXX – but replacing it with XXX.

AUTHOR: Okay...

SJ84999: Um, so of course we had to pretty much replace the rotor craft mission system with one based on age. So this involved sort of copying much of the software across. It needed some changes because there were certain functions that were different, you know, certain equipment that's on different XXX aircraft, and so this meant that we had some sort of need to reopen these software safety requirements, and this was like when you started to like, recognise that the work for record rally. If you look at the documentation set that you've been given. I think I can quote this because it's the table there with software safety requirements that set things like, um, the XXX keeping no safety requirements ....Displays often improve the state to display a speed on the HUD such that there is not a loss of air speed information, sufficient period and frequency for the pilot to lose the ability to monitor the indications of Airspeed over the road and lose situational awareness. Note that its not a software requirement on...you know. I as a kind of software engineer come along and know what constitutes losing situational awareness, you know when it comes to the USPS drawings, you know, its just awful. So we sort of landed with these safety requirements that were updates for this LIFTCAP, um, project, and the first response was to say we'll just ditch - there's no point in keeping, they're not telling us anything, you know they're not saying anything more than what is already said in our normal requirements, and they're just causing a major overhead in terms of showing that we've tested them. Um, so we were all set to get rid of them, and this was not just people as myself. It was, it was a...so we we're planning to remove this from our process on our side, as it was no benefit. Um, I don't know if you're aware, but the XXX, when they buy, um, an aircraft, or an air product safety xxx, then they send lots of technical people in to see what you're doing, and at the time we had somebody involved who was actually doing an XXX at the XXX.

AUTHOR: Never a good thing!

SJ84999: No, and when XXX got wind of the idea that we we're ditching these things, it was 'enough'. It's like you're not doing that, you're contracted to follow the …process. You've got to do it. So well then that's a situation where I was. I was at the time it was my, my, are I was at the time I was I think, I think I was actually written in the system safety team, um, in conjunction with trying to define the software safety requirements. So I said, well we can't work with these that, that's just not usable. So, um. So basically what we did was change process and reworked the safety requirements. I don't think there was a specific process of process change associated with it. Um. So uh, what we ended up what we ended up doing was taking the original safety requirements and reworking them by looking through. Seeing the software requirements and rephrasing the existing requirements and adding to them. Um, what the, what the original requirements did was it kind of. Of just. Expressed what the um…should do, but they didn't express what they shouldn't do? You know, they'd say deal with the XXX correctly, display angle up a XXX, but didn't state shall not fail to display. Yeah, or if they did say that nobody verified that it would indeed go wrong. Not go wrong. Did you know?

AUTHOR: No.

SJ84999: Somebody had sort of tested the puzzle as written, and tested the positive aspects, but they hadn't written down most of the negative things that the XXX might contribute to, and even if they had nobody analysed that it could do that, you know, so that the whole thing where you would have got value from these things above and beyond what you were doing with the normal testing, but being limited. So this is basically in summary what we did with the revised process, we introduced more rigorous, robust requirements, including ones that, um, that said articulated what the XXX wasn't to do. I was saying it wasn't um inadvertently move the Pitch Ladder, you know, it, if, unless there's a reason to move the Pitch Ladder, it should always display it. Yeah, um, and introduced a mechanism process that would, um, allow us to, um, verify that it didn't. There didn't appear to be any reason why the XXX would accidentally fail to display the Pitch Ladder. Um, and what that did was it introduced this extra level of analysis, which we did across the application software. Um, I think on 1 to 8, nobody had recognised, but it partition, in, in splits. Nope the applications software originates within a monolithic um, it got split up and partitioned across an operating system nobody had recognised. I think that you could ensure, you were introducing, uh, an extra potential level of failure, you know, where things could go wrong across applications. Yes. Particular application code failed to output some data. The receiving application might not realise that the disc is frozen and you might get an unsafe effect. So basically what we did was we introduced an extra level of analysis that verifies that kind of thing. Um it was very long-winded, maybe too long-winded - lots of documentation, um. Part of the reason I needed all this extra documentation is it because it was kinda like a bolt on to our normal process. You couldn't really capture this information, and this results in requirements tests analysis in the normal process. It had to sort of sit alongside it and it's all ended up being doing turning spreadsheets and Word documents and things because the tool sets didn't accommodate it. Um, so so we did this and, and it, it helps satisfy it helps make things visible, you know, the, the, the software was indeed safe. Um, I would say that during all that, it was very long-winded, lots of documentation to maintain. So quite a lot of time. Um, did it find anything actually wrong in the software?

No, I don't think it did, you know, over the years engineers. Um, have done a good job. On the radar of building protection mechanisms in the software that would handle failures do the right thing. If a failure records, all those things went explicitly requires or written down, in the higher level documentation and really what this exercise was, it was, it did that it made what we already knew. So probably the case visible to the customer, um, but we didn't fundamentally improve the safety of the product. So, I think that if we'd followed 178C it would've helped because sometimes personally, you know, I've done and now having to follow 178C that wants you to. Um, wants you to clearly state your safety requirements and everything you do has to be assessed for safety, and you have to derive safety requirements, you know, it's all very, yeah, very explicitly written down. It sometimes seems like an enormous overhead really. To what we normally do, um, but at least if we do that, we wouldn't have this thing where you can't convince people that you've done it.

AUTHOR: Yes, it's the achievement of safety and…

SJ84999: Yeah, yeah but what it doesn't do, I would say, and this is something we're battling with at the moment myself on the safety team on the projects currently working on is it doesn't seem to. I suppose it's the process once it receives the pro- the, the, the process, the equivalent certainly. So there doesn't really seem to talk about anything that would cause you to do the analysis, the SHARD analysis for example that we did on XXX. Um, its probably implied somewhere or maybe its in the ARP standard, but it's not leaping out at me at the moment. I'm thinking where does that fit in?

AUTHOR: You'll be pleased to know that someone else shares your pain - having modelled the entire ARP suite of documents. The distinction being that they make an explicit assumption that safety is done 'over there' using ARP 4761. So I do feel your pain.

SJ84999: I do have a full picture of what sits above 178, and I think, I am, I think that the split between systems and software is not clear either. It's not obvious what, what's expected? It's like well, I don't know. It's just, it's, it's, it's really struggling uh, most of it. Um, and it's like not clear, you know, like for something like we've got on, on XXX where you've got a series of assets, a piece of software that is split up on an operating system. Applications, you know, it's not clear. It seems to be the case, the, in many regards, the way that 178 seems…and everything articulates things, is that the, the, the, the, the, the 178C only applies to individual bits of software on that and that the whole thing is a, a….

AUTHOR: System…It assumes a deductive lifecycle where flow is unidirectional…

SJ84999: Yeah, it's, it's not, it's really only very, it doesn't seem to deal very well with anything other than simple one block pieces of software.

(AD HOC DISCUSSIONS ON THE MERITS AND DOWNFALLS OF STANDARDS COMMITTEES - A DIGRESSION FOR A SHORT WHILE)

AUTHOR: So I've been relatively quiet because I'm writing loads of really,

really interesting stuff that you've said about the engineers that built in these protections anyway, because they're good engineers, and that's a point that's been brought up by Drew Rae, who works over in Australia. He makes the point really clearly that standards and processes will not give you a safe piece of equipment, it's the people because they care.

SJ84999: Yeah I think this is on, on XXX it's got real good pedigree and certainly some of the main bits of software in it. The Source Code has, has some IT stems back from 30 years ago nearly and there was there was a great deal of craftsmanship went into it in terms of, you know, that detailed understanding in protection and designing and thinking about what you would do in and a lot of that has been retained and it's got some really good stuff in it, but. Any, you know anybody unless you look at the Source Code you'd be oblivious to it, and it's not, it's not expressed anywhere. So this whole exercise logic quite a bit of it was about expressing that kind of thing, um, or expressing things that were hidden in the design. You know, um, something might be documented in the design, but it's a case of ten in there almost since we're requirements, um, it's a lot of different, slightly different focus to try and argue to people that the, the things that conversely, I have experienced. And I won't name anything. I have experienced software that is being written to equivalent safety standards, um, safety levels, same language, um, uh, safety cases etc., and if you actually look at how it's written it is not done as well as some of the things on XX I would say it's not as I would say, it's not as...see if its the wrong way. Things like I suppose if you know, either, um, do you know Ada?

AUTHOR: Very, very top level.

SJ84999: You know, Ada has been traditionally used on aircraft projects within ... because it offers the ability to create safe software. It's often argued that it's strongly typed so you can use strong typing, and it statically analysable. Now, when it comes to strong typing, it only offers you the ability to strongly type things. You don't need that. We strongly type things, you can use typing in just the same way that you come in the DC language, if you want to just Um, you know, and on XXX we've always put a lot of effort into the typing, you know, it's to protect against inadvertent...and erroneous calculations, um, to allow extra tracking, you know, in the w-we've made a really good effort and there's always lots of thought and design goes into it. Um, but I've seen all the projects and things written to equivalent standards that don't do you know. And there's not thinking anything that stops here. That means you have to do it. You know you won't get that from 178C or from Def Stans, um. And it's obviously someone might tweak that when you're trying to argue that something, you know, it's, it's not gonna do this, this safety case, then it naturally follows that it makes sense to do it in a good way but clearly it's a bit of a tenuous link, you know. Um.

AUTHOR: Yes, uh, you. You want to argue that standard ...178 leaves the argumentation of why it's good enough to the individual organisation, but they can only do that because they're relying on a regulator or certifying body with Crown or the XXX. If you are doing this willingly or voluntarily you're conforming with a standard, but your benchmark is good, is that your engineering judgement?

SJ84999: Yeah.

AUTHOR: Can I take you back…you made a really good point about the SHARD analysis that was carried out. I'm paraphrasing here so correct me if I'm not saying it correctly. The requirements that came out as a result of the original SHARD where no different to the functional requirements for the components.

SJ84999: The functional requirements, um, on XXX, the direct expression, they predominantly expressed using, um, used to be, oh, is it, uh. I've forgotten the name for it now. Um, is that basically data flow. Um, so the express, the whole, the whole of the whole of the system, including the Mission Computers and the software running the Mission Computers modelled using data flow. Uh, no, no, it's a functional data flow, so it's um, I can't remember the domain name for the, the, the old technique - Teamwork or Moscow or something like that. Y-you just break down the functionality of the system in the final, final detail with a set of processes and you, you should…you use data flow to show the relationships between the processes. Um, and ultimately you end up with very quite low level things that are, um, describing quite a lot of detail what the functional requirements of the, of the software are on the RFP. Um, so those things, um. Tell you everything you need to to know, you know, like, there's a data calculations that kind of thing, you know, you need to calculate… using the following equation, blag, blah, blah, blah, blah. Um, and that's for the majority of the requirements on day with P-take, um, the it's process generated text requirements, but the majority of them were just re-expressing what was written as a functional requirement. The functional requirements were sort of descriptive, the process requirements were kind of functional descriptions, probably with a company in…algorithms etc. Um, the, the resulting software safety requirements really just extracting a text-based version of that was in the data-flow model by and large.

AUTHOR: Any thoughts on why that happened?

SJ84999: It's, it's because. It's it, um. It's, it's because, um…

AUTHOR: I don't mean to put you on the spot. I just wondered whether you have any thoughts as to, as to why?

SJ84999: It's because they, the majority of the safety requirements were obviously already, you know, the system analysis, the system safety analysis, and the system design already, you know, over the years, it, it, it, already analysed the hazards and flowed down, um, the whole of the sort of functional design of the system was aimed at mitigating those hazards you know, so, um. You know, losing situational awareness and this sort of this, this need to, um. I think it was the, um. Um. And the, the pilot, I think, on the, on the HUD that the sort of like the, the two were kind of compared against each other and if the pilot can get some kind of combined effects of what the, um. What the, um XXX based on those things, I think that the whole sort of functional design and what the Mission Computer naturally does is aimed at mitigating hazards. Um, so anything that it needs to do to mitigate hazards, it's already done. You know, in terms of the system level hazards, um, caused by the very nature, so they've already been captured as requirements.

AUTHOR: Um, would it be correct to say that it was sort of an inductive flow-down, but no deductive considerations as to how the, how, the hazard could be realised?

SJ84999: No, they, they review the Fault Trees that show how the Mission Computer can contribute to the hazards.

AUTHOR: So, I'm thinking more the specific failure modes. The reason I asked you that…is (the process) still designed to look at how a failure can manifest, not just that it will manifest so prevent it?

SJ84999: Yeah.

AUTHOR: It's not designed to consider deductive flows?

SJ84999: I think this was part of the problem, the term SHARD was banded around quote a lot when, um, we we're doing…And when it came to the…as a whole and the software safety requirements, it was stated that SHARD analysis was being performed to drive out the software safety requirement, but in effect all it did was look at the interfaces to the…the external interfaces were involved in the safety related functions and it looked at what failures could occur in those interfaces. And what the RFP should to protect against them, um, or to react to the. The SHARD analysis did not look didn't do a sort of an outward looking, um, analysis. it didn't look at how it could deal with if something goes wrong. Nobody analysed that, nobody said, right. We've got an application layer that's partitioned what can go wrong that could, cause, you know, they, the display to go wrong and I think, I remember at the time when the XXX were involved. They seemed to have a. Was this XXX from XXX that was doing a XXX and this is probably tied in with what you're saying here. It seems to offer a…with me sold what we were doing at the time. I think that XXX said I seem to remember XXX saying you've told us XXX was the only analysis to do to derive the software safety requirements and you didn't, this isn't SHARD analysis and I never understood what he was saying, but, um. I think maybe ties in with what you're saying, and at the same time I'd also had a slight concern myself. What do we want to wait the, we don't all of this proper SHARD analysis on the software platform to see how it could fail and induce a problem, but we hadn't done the same on the application software and as it as an incident. I remember how the ones that when were in the infancy of developing the XXX aircraft where, um, where we do need part of the application software and we will. Sort of driving the imports with with a similar, a simulation of the inputs on the 1553 BUS and I remember looking at the pitch ladder and it was turning, you know, it was, it seemed to be rolling, um, and I remember one of my team members saying, oh look, it was great. We've got it. I was like, well, it shouldn't be. The rules of fixed value. Yeah, nobody was stopping it. It won't change it lie. Oh w-what's happening I said, well what's happening is the software's assigned to extrapolate stuff to account for delays, um, in between the risk. Of information, and it's displayed on the HUD and you've got a situation where its extrapolating a fixed value, so you get in a row, um, and it's um. It's, it's, it's giving you the wrong effects and that's the sort of thing that would be dangerous you know. I think that sort of thing, that would be dangerous. Um, so I'd thought about this being important that we haven't done this level of analysis. I mean, in effect in the full system is fine, you know, I mean, it's, it's, it was not good. It was going to be okay, but it did always make me think so. At the same time, this XXX, kind of banging on about you having to reach out and also shout at me that you need to do more. I sort of recognise that we often don't have as much on the application. Software is what we've done on the platform. So, um, so I was quite kind of happy to go ahead and do extra per project. You know, I mean, that's the thing the safety process is not blocking, y- you know, improve things over time. I think one of the problems we've always has is the expectations change over time, you know, if you look back over thirty years, it'd be no concept of doing all this, at the end of the day.

AUTHOR: No, no.

SJ84999: No, uh, uh, non- XXX w- w- so it was like, you know, like…every time we redo what we do, uh, a new variant, it means. Sure, we have to do more of this and it's good in a way for, in another way, does it ever fundamentally change anything?

AUTHOR: A billion dollar question.

SJ84999: And, and it relies on people. Savvy people doing it in order to have any benefit coming out of it anyway, you know. You look at the risk requirements we ended up four months away. The people writing those didn't have much of a concept of what was in XXX, you know, or the software or anything, and that was the result that you ended up with something that really wasn't much value, um. A much better job was done on XXX because I was doing it and understood, you know, line by line, what was in software and what had happened over the years. Um, so I had a much better appreciation, um, and could offer more value, I suppose, um. It all relies ultimately on the people I think. 'Cos I think if you gave it in depth knowledge of what's in you really appreciate the analysis, you do it anyway. So. I don't know.

AUTHOR: That really does resonate with my research area. Modelling of standards has raised a very interesting point about every new application, new project, new technology, one of the difficulties that standards committees have, of course, is it takes them about 5 years to agree on a paragraph change. I'm exaggerating, of course…but it takes at least three! So, for the standards we're discussing now, for 178, uh, that asserted best practice or recognised good practice was written eleven years ago, but let's take a view three years from you…fourteen years ago, recognised good practice and the average lifespan of technology is less than 18 months. Uh, now, um, that's the problem, that they must stay relevant…and they're written by a group of international experts…and there is no empirical evidence that anything in any standard makes anything safe or safer.

SJ84999: There isn't no, you're right. I mean, it's like, I looked at 178C and I sometimes say, it's p- this is my opinion. I have only. I've not actually done it fully in practice…but one thing I have is a little bit of a bee in my bonnet about what they sometimes think it makes things might be stupid. But I, I sometimes think it looks safe because, um, it expects, expects, um, very vigorous definition of requirements and express low-level requirements for everything. And expects the safety team, to like, look at these low level requirements and assess them now. My, my concern is sometimes that, that things come out because things are not just flown down, if somebody gives you a requirement and you have to, um entirely protect I suppose. It results in a sort of vetting of the requirements that might not happen if it's more, you know, if it's not quite so, so really, however. I do feel that it's almost like if you run things down in too much detail, we get people just manually test, but you met them and you're on con, but they won't really question whether the thing's correct if you give people too much on their plate, they don't put the thought process in, and, you know, this was another problem we had like, low level requirements writing everything down, you know, the senior software, um. It's a massive overhead, and we actually used to kind of do it - not low level requirements as such, but we used to write a lot more in our designs in code on what we do. Now it is, we actually removed it because it was just a maintenance overhead and it didn't add any value, and this was another thing that the XXX said it seems like you've got rid of all this stuff, so it's unsafe, and really what XXX was saying is we want you to do 178 level of requirements, and you're not doing it and we're not happy.

AUTHOR: What's your definition of a low level requirement?

SJ84999: Oh, that's another thing.

AUTHOR: Sorry, that might be unfair of me…

SJ84999: No, no, this is a like a non-written expectation of 178, I think. If you say, well, if you take 178, just at face value and it's not dictating they take any particular form, is it, um. Yeah if you, if you talk to, various senior people doing more that 178C, let's say all these low level requirements we've got to be like, really, really explicit, you can't have you kind of almost everything has to be written down as if it was a low level requirements. High level requirements can't have any information - all that goes just goes in the software. Where, you know, it, if you're, if you draw something in red, you've got to define the exact, you know, green red blue values that it needs to be drawn in, and that needs to go in as a high-level requirement or a low level requirement, and this kind of thing, and it shall, you know, explicitly test it, and, um, you think, well, where did it say that in 178C. It doesn't say that.

AUTHOR: No, it just tells you the three levels, low, medium, and high.

SJ84999: The other thing that confuses the hell out of me then, is well, of course, a lot of the sort of operations that are going in new designs tend to be traditional, right pseudo code. So yeah, you write pseudo code because you can't express it without writing pseudo code, cause it's toward the right in English or whatever. So you think - what is that - is that pseudo code or a low level requirement? I think I'm not sure really, I mean, I think it’s more like code than pseudo code because it’s not really, um, you know, if, if, I always use the example of a sort in alphabetical order. If you ask the right pseudo code to explain how to sort of list…or it might be quite involved and you wouldn’t really be able to tell what it was trying to do, whereas if you wrote a low level requirement that said, so the listing, so alphabetically that would be a much more explicit requirement, so you know we’ve had investigations into that and it came to the conclusion. Yeah, well, yeah there’s definitely something in the questions and answers section in 178…it says that you, should you avoid, you know, you could just use pseudo code as low level requirements, just because of these sort of problems that I’m saying with the, like, sort of explorer but again, if you read the main body of the document and you didn’t tell you about that. So at the moment on our project, the people have got different ideas and you can come to blows almost over it, and, and we’ve got some really struck this because we have definitely got different teams. We’re trying to define processes and we’ve sort of got a, as an example we have a team who’s sort of almost developing a modelling process who, who were kind of writing stuff that they say, and it’s a low level requirement or can constitute a level of requirement and I personally don’t think it comes out. And it’s almost left to a matter of opinion almost you know, and it’s gonna become a real problem because it’s not, it’s not clear, and I’m not one hundred per cent confident that my thoughts are right? I’ve never done it before I’ve never tried to get through certification or thought I’d stay with it, and until you do, how would you know what’s right? And wrong?

AUTHOR: No, I mean for what it’s worth, um, it’s very similar to, when people talk about complexity or size and use metrics like medium/large – the question I always put back to them is okay, you’ve described this as a low level requirement, but what does that mean? And if I’m describing something as a medium requirement, what does that mean? What changes in reality? And the answer is nothing - apart from where your trace ends.
SJ84999: Yeah.

AUTHOR: So the question is it low level? Is it medium? Is it high? The question is does it matter 178? Unfortunately it only majors on a trace, not the validity of the trace - so what you were hinting at recently. I do feel your pain, because as a consultant this is mentioned frequently. Also frequently asked is when does a it stop being a requirement and start being a specification – and there’s no clear answer, other than when there is no value decomposing it further - and that is judgement based, because it is technology and application usage dependent.

SJ84999: That is it, yeah. I mean it’s it’s like that 178, w-what’s the statement? It makes the low level requirements requirements that you come up with basically saying you can develop the requirements you can go ahead and develop source code without further information or something and it’s like, oh, it’s like that’s really ambiguous because some people read that as being well, it, most of it, most actually be the equivalent of pseudo code consumers, right in there. All the decisions are going in yet, you know, and I’m like, well, no, I just see that as an abstract version of source code. Because you could say that like in the old days before the ADS high level languages, and you have to write everything in assembler, you could say, well, the Ada is the low-level requirement, if, if you follow what you're saying you know, and I don't believe it is, so you know, write in actual language or something. I don't see that it's a low level requirement. I see that as being an abstract representation of the actual source code. Um, but. Sorry, it's like since, you know, you go round in circles on this kind of thing and it wastes an awful lot of time, you know, discussing it.

AUTHOR: Yeah I think part of the answer is, is that your company sets a clear definition of when the requirements stop.

SJ84999: Well our company doesn't know, you know.

AUTHOR: No but I think that's the only reasonable solution and you get the, uh, then my advice - sorry, digressing a little bit - would be to get the certifying body to agree with those metrics.

SJ84999: I mean, the ironic thing with it all is as well from a safety perspective. Does it really matter?

AUTHOR: No, no, oh yes, it might.

SJ84999: It might you know, a lot, a lot of it seems to be in in builds. I think this is another thing, maybe this isn't the other thing to come out of this, but quite a lot of these things seem to. Most work on the basis I think that there is this hierarchy of superiority or intellect or ability, um, you know, the top the system people define the, um, the system requirements and flow down software requirements onto the software team and, you know, that sort of the people sitting at the top, you know, and those software requirements, you know, they, they can be relied upon, but turnkey software engineers that I've got to make sure that they design the software to meet those requirements and implement it. Um, then you sort of get the design, but in the middle of coming along, and it's like, well, these people can be relied upon though they're far superior to those who code at they bottom, you know, they, they'll, they'll write the design and they'll flow rigorous low level requirements on to the software people and those because it better meet these low level requirements or else, um. And those low level requirements, you know, can be relied upon to be okay, you know, and as long as it can trace them, so the high level requirements then define, you know, and then you get the poor little monkeys down at the bottom, whose job is just to meet low level requirements, you know, as long as they…implement these low level requirements then then publish it all. And it, it kinda seems to follow that principle, you know.

AUTHOR: So they can just 'give it to a process monkey'?

SJ84999: Yeah, and it's like, well, that's not really what makes safe software. The, the, the, the thing that's makes it safe is those people at the bottom without question. Um, you don't want support so much information, but they just end up switching off…implementing something without questioning. That's, that's one thing that I do worry about. And to try to be more supportive and encouraging, if that's low functionality, it's that there is a reasonable argument that it says each layer of design abstraction requirements fully typically, specifically, explicitly or it can't be, and we need to derive it further. Um,

AUTHOR: I think the trick with that is the strength of the requirements, and more importantly the measures of performance.

SJ84999: Hmmm.

AUTHOR: You hinted at the different colours which may be required – you should be able to reference a human factors schema as your measure of performance that says, how any HUD or digital display warnings are expressed. This advises one aspect of it, and here's the list of colours that are acceptable and testable. Here's my design, but then, yeah, the minutiae. At some point there is no value in deriving further.

SJ84999: No.

AUTHOR: Alright, I have lost track of how much stuff I've written down. I will very likely have to follow this up with some emails to you. This has been so useful, um, not only the stuff you described, but also the feeling of a kindred spirit!

DISCUSSION ON WORKING GROUPS AS A DISTRACTION.

SJ84999: … That, that just reminds me a little bit of a corollary too. You said, oh, when it talks about the monolithic design of previous variants and that, that must be very expensive what we've actually ended up with on Mk 1 was to wait, and subsequent development when it comes to the software that's being developed by the main team it is all done to the highest SIL anyways. All to SIL 3, which was the highest filter, um, but, um the system has flown down onto it. Um, the reason for that is that the system analysis, um, is XXX - it's so subjective to change that you can't rely on the functions that are identified. And they associated SILs, um, sufficiently early in the process to be reliably do the software to different SILs. So the risk of change is so great that you went into having to develop everything to the same high level process anyway, just so they can accommodate changed, um, stream when they suddenly decide the different ways software fails. So in terms of that, um, we didn't get any benefits from having an operating system in terms of reducing cost on the software that we developed, it just increased it all, it gave us all it gave us was if you get third party software in the, it's clearly, um, I can moved in math, but it's clearly only ones who were lower level and that you can see its functionality contained such that you can, you can make it only simple to say they're not seeing any sort of mix of skills that we actually have on call. Everything we developed was always done to the same SIL level because you can't get the system process, not quick enough before you have to start developing the software. So, to do it to different levels. That's what I've found in the past.

AUTHOR: A difficulty also, of course, is the different philosophies that you are experiencing, where your contract is still on 55 Issue 2, it was talking about integrity as safety integrity of components, and you then have 61508 which talks about the integrity of safety functions, As such, the difficulty you have transferring to 178C is you're now talking about arguing equivalency of DALs which aim to lessen or moderate the severity of outcome.

SJ84999: Yeah.

AUTHOR: So you've got two different philosophies. So you're already talking apples with pears, but I do agree at the point that if requirements that come down as safety requirements at a system level aren't um good enough, well actually they should be rejected.

SJ8499: But you don't get, you get, this all seems to like imply that there's like this Waterfall (lifecycle) and we all just sit there waiting for systems to do this massive hazard analysis to flow things down onto us, but it doesn't work like that. I mean it's interesting as well in a certain respects since I ran into an account for, um, analysing the contribution of…in terms of what we do. Yeah, and for that you need sort of the system design to develop in the software, um, requirements to develop and sort of hazard analysis overlaps, so you don't just sit there waiting for these things to be given to you, you can't, you know, you have to start to specify…integrated with the software before you finished it. I mean, one of the problems you have with the asset analysis is it's a massive, massive job that's dictated by the aircraft level three. So you have to produce all these figures and stuff. The default apportionment like probabilities of failure down and blah, blah. No, no, no, no, well ultimately that kind of governs and they, they, the safety levels of software needs to meet. So theoretically depending on your projects and they, um, and the failure rates allowable failure rates on a number of aircraft in the fleet and things you could end up with this change of SIL, you know, um. If you think about maintaining that's crackers. Lately, um, it really is. Something like allowable crash rate of warning x whatever, you know, so depending on your fleet size and things that can influence what you're doing, and what is a massive mathematical job to chop all this stuff up. So you never get a result so well down the line.

AUTHOR: Mathematical voodoo. But it gives people a warm comfort blanket that says if it's one in ten or one to minus 7 it will never happen. You try to point out that it still could - on the very next (sortie).

SJ84999: Yeah it's crazy really. So this is one of the problems we do, you know, you can do developing for the rest so I think that the only time that you, you won't do that is if somebody identifies, you know, a delay or whatever or so forth. Um, and on XXX we've got, we mitigate it becoming so far, by, um, putting in, you know, detection in…in the system design to prevent the Mission Computer from straying too far. Um, yeah.

AUTHOR: I mean, although your specific software failures…obviously drive an integrity level, if you do have an early understanding of the hazards that you contribute to, it is possible to start the inductive analysis by considering as you rightly said way back when SHARD or something is done correctly. It's both deductive and thinking about how can I contribute to this hazard? Not just how do I mitigate it? How do I realise this hazard? Yeah that's what I was trying to ask what your opinion was on why they didn't reveal what was intended, which I think the XXX colleague, um, that was, that was poor timing that XXX was part of the procurement team and also doing the XXX at the same time.

SJ84999: It was because I was really cursed. I mean, they, we've always found that the XXX are very savvy you know. They always send teams of people to, um, that your engineering or be involved in engineering and also send pilots and things to assess the products and get heavily involved in trying to influence the direction it's going, and I'm as I say we were, we were. This XXX, it was a very, um, who is very, um. I don't know a very difficult person... Yeah he was already in the UK and so, yeah....So he was obviously well acquainted with all the latest possible techniques as well...I think from a software safety perspective, there's nothing that, it, it, it, it, it, it, it's not really a case that somebody's dictating a set of processes. So we'll just stop following a meeting with the pointless. The only things we have dictated, they don't know, um. Are they, um, you know, the Def Stan and 178...everything else is, um, left to the department to develop I suppose, um. Clearly as I said, the only thing, this thing would be XXX we always send it without...contractual yeah, always good to have. On track too, well, I mean, we on the opposite side, we will not contract. We were contracted to do the processes that we do not do - and that's why we ended getting hold of the code that is C Sharp. Cause XXX didn't believe that we did what we said we'd done, but on the contract side, yep, so we got to dig our heels in the ground a great deal because it was clearly the XXX particularly XXX were used to working to 178, you know, they have to get through things through the respective regulatory body...and they're used to dealing with 178 and we do not develop to 178 for XXX and then so that is key things we're missing, missing information in the designs. No low level requirements, that kind of thing. I really didn't like it and tried to get us to do it, basically try to force us to do it when we took our XXX and said. Oh, no it's out o contract and it will not, if it could also be happy to do it if it would have added any benefit.

AUTHOR: And if it was funded.

SJ84999: And if it was funded. And they were basically saying without it they didn't believe it without it....any benefit because it was the equivalent of what we previously had it removed because it didn't add any benefit.

AUTHOR: No, but applying it in the way you did subsequently added some benefit, or at least I think you said it identified the bits that mattered - that you needed to demonstrate.

SJ84999: No, no, the low level requirements thing I'm talking about here.

AUTHOR: Sorry.

SJ84999: Um, you know. This, this is where I'm a bit dubious with the low level requirements myself, you know, the way that it works on, um, XXX, it's always where is it's on XXX. It's software, the same people who designed the software, implement the software and vice versa? You know, it's not this some superior beings do a design and then handover to somebody to implement it, it's all integral, and, you know, years ago, we would probably be following the processes we had to years ago, we would have a design that was very comprehensive detailing lots of code. You'd code, and then you would implement…what you found was the design got out of sync with the code and it ends up being a massive comeback job to stick all these extra stuff back in the design. And nobody looked at it anyway because if they wanted to know the things where they look in the code and we got rid of it, cause it was just a maintenance now….didn't offer anything for cost, and it's like really upset XXX came along. Oh, if I remove this stuff it used to be non-Roll stuff when we originally designed the aircraft. Yeah, and it was pointless, and a maintenance problem, and so we removed it, XXX didn't like that cause I think that …being a low level requirement, but, um, this is where I do question things. We want 178 and the benefits of these low level requirements. I see them as being good in flagging up things to the safety team that might otherwise be hidden, but sometimes seems like a bit of a slight drama to crack a nut.

AUTHOR: Yeah, so it does nothing differently other than would you say associate them as safety requirements?

SJ84999: Yeah I do feel they are a bit of a XXX. Maybe I thought I was being you know, I feel it's dying or something, can you see?

AUTHOR: I know, I thought kind of exactly the same thing when you described the maths, but I think it was a polite way of saying pointless.

SJ84999: Yeah.

AUTHOR: And adds no value.

SJ84999: That's good then yeah.

AUTHOR: I hadn't considered the extra costs and that's, that's really interesting.

SJ84999: Yeah believe me they will, you know, it's gonna. You know, to, to write them and maintain them because really, what happens is you, you code the things that leads, once you get down to that level, you know, people are doing things in the code, without looking, and might go back and tweak, you know, you'll have requirements, you know, it's, it's just like a massive overhead, yes, So, so traditionally, we have, on XXX, we ended up rolling them out so it was designed really well, it's just a, it's a useful thing to help people who understand all the software is built in the key design decisions and the key concepts in it, rather than just being full of all this junk. It's already in the code. You know whatI mean? It's like…yeah, yeah. Um, yeah I'm very skeptical of the standard.

AUTHOR: Um, and luckily, you're doing this on a conformance basis, you're doing it willingly, okay.

SJ84999: Sorry, I'm not sure what you mean.

AUTHOR: You don’t have to abide by it, you know, not contracted, to do, you can take the intent and the objectives of the standard and create your own process.

SJ84999: On XXX, no - but on the thing I’m working on now, we’ll have to do it.

AUTHOR: Oh, okay, in that case you don’t have the luxury.

SJ84999: Oh whoa don’t mind this at all, you know.

AUTHOR: But I guess that would be a point with the certifying body, um, that says ’at this point this is when the requirements stop and the specification starts’. Or at least provide the metrics for a tool that can read through…at the same time which is…always a flawed plan.

SJ84999: Yeah it’s tricky for this project, the, it’s not clear who the certification body will be.

AUTHOR: That’s even worse.

SJ84999: Yeah, um, I mean, we have all the info in the past from XXX with what we’re doing, no, no so they’re just kind of making it up.

DISCUSSION ON CURRENT RESEARCH PROJECTS (OMITTED FOR CONFIDENTIALITY).

END OF TRANSCRIPT.