

Agile Methods for Developing Safety Critical Airborne Software Products

Wed, Oct 12, 2022 6:51PM • 55:31

SUMMARY KEYWORDS

project, cmmi, cmm, agile, requirements, aerospace, quality, software, kritika, questions, airborne, customer, critical, working, organization, configuration management, agile methodologies, process, review, development

00:02

Welcome, everyone, and thank you for joining today's webinar. My name is Charles Maddox with the eye for group. And our webinar topic today is discussing agile methods for developing safety critical airborne systems. A very interesting topic, especially those that are new to Agile methods and think that agile methods can apply in a highly regulated compliance driven industry. So I'm sure you'll get a lot out of today's discussion. I have a distinguished distinguished panelists with us today Kritika, and she's going to give a again a good talk about how to combine some of these safety critical aspects of primarily CMMI and d o 178. Criteria into your development process critical you want to give a a brief introduction to yourself and yeah, overview of your company.

00:57

Definitely Charles, thank you. Hello, everyone. Welcome to the webinar, and I'm critical Santana Krishnan I, the principle of excellency grew, I have more than 17 years of corporate experience specifically in aerospace domain, I have worked with a lot of safety critical applications. software's ranging from Level A to Level D. And my organization focuses on software consulting, product project management, consulting, ERP implementation. And in this webinar, we will be focusing, and this is an introductory webinar, it's going to be a series of sessions. Because it's a big topic to cover in just an hour, we are going to have a series of webinars, starting with an introduction of two main pillars. One is DL 178, which is the guidance for airborne software. And the other one is CMMI. So we are going to introduce you to concepts of do 178 And CMMI. And as Charles explained, we would like to take you through how we can implement a giant while still following the 178 and CMMI process areas. And we will also talk about what are a couple of pain points that we have in the current projects, and how we can address those pain points and challenges while implementing agile. So as I stated, it is going to be an introductory sessions and Charles and myself are open to any questions that you all my tab online offline, and you can contact us and we will consider subsequent sessions on this topic as well.

02:53

Alright, thanks, Kritika. Little bit of background on myself too. I'm Charles Maddox. I'm the principal and founder of Vi for group and we're a consulting company as well as a training company and agile methodologies. And actually, my background is and that's kind of where my start came from was an aerospace aerospace software systems. And that's how I got my start in agility. And, and so this this topic really resonates with me and, and critically, I have very familiarity with some common products and projects that we've worked on together. So very, you have a lot of knowledge to share with you guys and some best practices on you know, building aerospace software, and applying agile concepts. Alright, so without further ado, I'll turn it over to critica. And you can kind of give us a background on the Capability Maturity Model. And what that's all about.

03:47

Definitely, as I explained in the introduction, we are going to focus on CMM and also dl 178. This session is not going to elaborate what CMM and CMMI is and what are the process areas of it, it's just going to give an introduction of the expectation of CMM and expectation of do 178 C and how we can accomplish those expectation using Agile methodology. So, people who are familiar with CMM CMM is a capability maturity model. It is a methodology used to define and refine an organization's software development process. The model describes a high level evolutionary path of increasingly organized and systematically more mature process. Basically, what CMM does is it focuses on elements of essential practices and processes required for software development, process flow, it is a matter to evaluate and measure the maturity of your organization in following the software development methodology. So the maturity level ranges from why M through five that is the scale, we don't call CMM as an audit, it is called as an appraisal we have a lead appraiser who would come and appraise your organization maturity towards where we stand with our software development processes. And earlier CMM was focusing more on software development. Now, we have CMM practice in people management in systems engineering, that is where different models are being integrated together, we call it as CMMI and CMMI level one through five have different process areas. An example would be if your organization wants to be level two, there are different process areas for Level Two such as requirements management, project planning, project management and control and supplier agreement management, measurement and analysis and product quality. And we have configuration management as well this is just an example of CMM level two if your organization would like to achieve level two, similarly, we have process areas for level three, level four, level five and each of these process areas each of these levels would have different process areas like I explained as an example to accomplish and each of these process areas would have specific goal and generic goal to achieve. So, one of the benefits or some of the benefits of CMMI is improvement in scheduled productivity quality, because we are repeating consistently, these process areas and their objectives and showing evidence is how we are meeting those objectives. And when we are accomplishing quality, productivity and schedule obviously, we are going to get customer satisfaction and many regulatory industry requires the organization to be CMM level three, level four or level five. So, if your organization is at a particular level of CMM, then you have a competitive edge against other organizations which are not there yet. And also consistently repeating these goals and providing evidences will also generate bring you a lot of accuracy towards your goal. So, that is the key and the benefits of CMM in order to go in detail of the CMM each of the levels and one of the process areas and also to help your organization accomplish

those process areas. Obviously, Charles and I can help you out offline in the implementation goal. Here, we would like to cover the high level benefits because the more focus is going to be how we can bring in Agile here. Next and why we are implementing these goals and obtaining evidences, we are going to reduce the rework. And there's a lot of risk reduction because these going through these process areas and goals help us to think at each areas which we call as this best thinking towards the entire development lifecycle. And also we will be reducing costs and also reducing the defects and delivery errors because we are going to obtain evidences and also try to meet the objectives of each of these process areas as we are going through the entire development lifecycle.

08:44

Alright, so the first part, we've covered the CMM and one of the key things that I would like to talk about before we move into DL 178 c with respect to CMM model is the main goal what CMM model helps us to achieve is the processes need to be well defined, it needs to be repeatable, we should have a way of measuring these processes, analyzing the measurement and making sure we are there reaching the goal and improving it consistently. These are couple of key aspects of CMM which would help us to drive the project and the organization. So can have these things in mind when we are talking about agile and because we are going to explain to you how agile is going to help you in accomplishing these goals and with respect to following the CMM right would help you organization to mature at different levels. And we will talk about how an organization what are the benefits of having a matured organization mature processes and what the immature processes would lead the organization to then we talk about the agile. The next goal for today This discussion is deal 178 C, people who are familiar with airborne software would know what do 178 C would be? I will give an introduction of the 178 C and what do you want 78 C helps us to achieve while developing airborne software. And then later in the slides, let's talk about how we can implement agile into do a project that follows do 178 C. So do 178 C provides the guidance to produce software for airborne systems and equipment that performs its intended function. This is the typical definition of do 170 se with a level of confidence and safety that complies with airworthiness requirements. So basically, any regulatory agency that would certify your air worthiness software would use the o 178. C as guidelines for certification. And what do 178 C determines us to do? It's a poll, it provides you a set of objectives for each phase of the software development lifecycle, and what activities we need to follow to meet those objectives, and what evidence is required to fulfill those activities and objectives. And also it talks about independence, software lifecycle data and control category of software level. This is not a dedicated do 178 C session. But anybody who has any question with respect to do 178 C and implementation, you can definitely contact Charles and I. And we can take separate sessions on do 178 C. So basically, when we talk about independence and software lifecycle data, the huge deal 178 follows expects a lot of documentation to be done and also to ensure the objectives are being met. And certain levels of software here, the levels of software we're talking about is design assurance level, that's what we call with you 178 terminology, it depends on the criticality level of your software. For example, if you are working on in flight entertainment system, that could be Level II or a level D. And whereas if you're working on an actuator or flight control system, the criticality of that software will be Level A, which will be determined by system safety assurance team or assurance group to determine the criticality level and each of these level requires independence, for example, the development developer cannot be the same as tester for certain levels, certain levels, we need to have for all the levels, we need to have a separate software quality assurance group. That's what is the independence we are talking here about.

And software lifecycle data is actually for each of these phases. What are the software lifecycle data we need to produce and control category here is related to your configuration management. And we call it as control category one, control category two, and what are the guidelines for each control category, which is like for example, baselining, change control, problem reporting, and watch data needs to fall under what control category and what guidelines need to be followed will be determined. So this is a very, very short explanation of the Oh, 178. It's an ocean, it's a topic of itself. So definitely anybody needs any support can reach out to us.

13:41

Excellent. All right. Well, thanks for that. Yeah. Yeah, thanks for thanks for that. Kritika. So and agile methodologies and, and how they relate to those requirements. So Kritika just gave some typical, but very standard industry requirements that are placed on airborne software development projects. And now we're going to talk about the agile methodologies, that and how they relate to those requirements. So not specifically that demographics, but the agile mindset approach in the Agile delivery delivery methods that are being used, how do we reconcile with which are sometimes seemingly, you know, long term sequential, and some, some even float? They think they're waterfall ish. It's just a waterfall ish way of working by nature, you know, big aircraft projects, that's the only way we can work them because you got big upfront requirements and you know, the whole lifecycle is just so long. How do you fit agile into that equation? So what we have here, what I'm showing here are some of the, the, a little bit of the pair of phrase agile manifesto principles. And in this think about how these principles then fit into your ways of working within an aerospace project. Alright, so, first principle, right or this first what we got here, this first bullet point is satisfied customers through early and continuous delivery, of value of work of feedback of, of information. So oftentimes having what we might call iterative cycles that don't, you know, we say, well, the plane is not flying it. So we don't get the full value. But there's a lot of small increments of valuable discussion information, documentation, which is often required in in aerospace, in highly regulated, highly compliant environment that is that is necessary. So we can iterate we can satisfy customers through these earliest, early continuous delivery of these items. Right. One of the things about welcomed changing requirements, even late in development, and critic could could speak to this, we talked about it before that the change impact may be so significant, and an aerospace project that it needs to go through a lot of change control. I mean, you don't want planes dropping out of the sky, because of, you know, changes that were missed, appropriated. So change management is big in the aerospace environment. So the way to de risk that is to have frequent review, on the opportunities to look at change. And again, that's where the iterative cycle comes in handy. And then actually, the risks the entire project altogether, it makes it a morkie, it makes it a safer environment, you know, safety critical software, that's the whole purpose of do 178, we're trying to prevent it, we're trying to ensure safety, it's better to iterate and simplify and have shorter review cycles to make that happen. Deliver frequently, I just talked about that. There's no contesting that. But breaking silos, breaking down silos, that is something that's newer, I would say it's a it's a newer concept to the aerospace industry, because traditionally, it's so functionally siloed, because of the nature of the big projects. And now you need to, you know, kind of optimize on, you know, jet engines, for example, electronics, you know, landing gear, you know, communications, you know, you need to have specialists in the silos. But if you're, you know, anybody that's been on a project long enough or been on aerospace project, you know, that the integration piece is the most significant, it is the biggest factor on whether you can have that flight test and actually have a successful into a project, the sooner you integrate, usually the, the

shorter the timelines, the better you meet the end dates. So you need to break silos, in order to make that happen, you need to have collaboration with these groups working together, why funded, you know, fundamentally integration,

17:51

and then testing is so significant, building projects around motivated individuals. So there's a, you know, like, in any, this is kind of a across the board, whether you're working in Agile, or air, you know, agile project, non aerospace, you know, working on building websites, for example, versus building safety, critical products, that sustainable pace is a big factor, right? You don't want to contribute to burnout, you don't want to contribute to a, you know, a bad environment that's not supportive of, you know, the culture generating the culture of the the the agile teams and the product teams, you want to make sure that we're creating a good culture and motivation is very key. Communication. And obviously, face to face is the best way of working. But communication is significant. And oftentimes, in your traditional waterfall is type project. The communication is what we call asynchronous or happening when needed or with with, you know, longer times between events where people get together and discuss the program and the project's technical dependencies or work that's happening, right, the more frequent the more predictable cadences that we can have around communication, the better. And we set the standard to what we're trying to get out of these different events and objectives of how we need to get feedback into the development teams aisle, making sure we're moving in the right direction. And just like anything else, any other product or project that we're working on, not just an aerospace project, actually seeing working system demos is appropriate. The aerospace industry and highly regulated industries have a they have a lot they have a history of wanting to just have a checklist to check off meet, we've met the boxes of the critical design review, for example, that critical, it's not that you don't have to throw that critical design review out the window, but a part of that critical design review should be that we maybe have seen some working working code. We've actually seen a system demo. So these design reviews and these checkpoints. step might be a standard output for some of your state of involvement reviews, or have such a part of the deal 178 method to make sure that we're meeting the right requirements, having a working system demo, it should be a core requirement, because how can you actually verify progress if you're not actually seeing with your own eyes, the work that's getting done. So make maintaining a sustainable pace, I already talked about that. Moving on to continuous excellent, continuous excellence, and in terms of technical practices, and that's one area where agility, and the methods of CMMI really come to head because they both promote do we have standards in which we're building our product to are we building and, and acting and just in our processes as well? Do our processes have a standard that are repeatable? Do our product requirements have have a have requirements that are we can produce them in a repeatable way. So that's a part of the technical excellence that we're talking about, that we're agile, and techniques, like CMMI, are completely in support of one another. Simplicity, I gave a little bit about that, before that, you know, when you're building airborne safety critical systems, the environment can get very complex. I mean, especially when you're trying to evaluate some type of safety critical nature about the product and you need to you need to be able to prove that you have safety considerations built into the product, whether it be through a failure, failure modes and effects analysis report, or, or an impact report of some sort that you can actually prove that but doing things in smaller batches, smaller evaluation of smaller components is always a way of simplifying your analysis. And again, getting that feedback and being more particular about the smaller bite sized pieces to make sure that you are getting the level of granularity and, you

know, focus on the safety critical aspects as needed. So the Agile process, again, kind of showing where Agile is in support of the evaluation of safety critical items, being able to simplify the safety critical items into smaller batches is always a better way to go, less riskier way to go and produces ultimately safer products in the end. All right, self organizing teams that generate value that generate the most value. So self organization, that principle around, you know, agile ways of working is ultimately this is something in practice, which we've seen on in some aerospace projects, where the ability of teams to self organize themselves, always kind of Windover versus having a very rigid structure and your traditional project team structure, and then having these teams gotta force collaborate, work together, just based on the project managers, or program managers, you know, you know, their directive, whereas if you have a group that is meant to be collaborative, and you're telling them that, hey, you need to reach out, you need to work with other teams, you need to form agile teams, maybe cross functional teams, if need be, where appropriate, this enhances collaboration. And either this, obviously, you know, you work faster, and you get things done in a better way. And then like, like any good agile project, you know, in any good

23:35

project, even in CMMI, we recommend, you know, using even data driven improvement from a CMMI perspective, but we should be reflecting on regular intervals to enhance the continuous improvement of the program that we're on. And yeah, why wait, if you can take action on some good practices that you've identified during a sprint, and you or a program increment, for example, that you want to implement going forward. So these are a couple of just ways to reconcile how agile manifesto principles in, you know, some of these aren't principles, but some of these are just ways of working are also there in supportive methods like CMMI and do 178. And before we go, and I'm just going to turn it over you Kritika here in a second, but one of the biggest misconceptions and highly regulated organizations in general is that agile and highly regulated, don't go together, like highly regulated must be done in a waterfall way. Whereas Agile is more meant for building a website, you know, a product that you know, product development that happens in short, you know, short cycles where you can get something to your customer in two weeks, that you know, unless you kind of fit that, that that kind of concept. You can deliver your customer functioning feature and two weeks and they can use it. If it's not Not that he can't do Agile No, that's not that is not true 100% Not true. And airborne software, you know, airborne software can definitely take advantage of this cyclical iterative approach where we're de risking overall development, where we're acting on the feedback that we're getting every iteration on the direction that the teams are going, we're integrating more, we're integrating sooner than later, because of the, again, agile principles that we know we're working under. Overall, we get a lot of benefits and working in an agile way. So before I turn it over to critica, just you know, be on the lookout, we're going to actually go more in depth into some actual practices and processes that we've used on these projects that we worked on together to actually show you how agile can be a benefit to your airborne systems projects are critical to use some of the common current project challenges that you see facing airborne systems projects.

26:03

So thank you, Charles. This list is from our personal project experience that we've had using different other methodologies. requirements definition at the beginning of the project, for most of the methodologies, you know, that is what is the anticipation to have the requirements defined at the

beginning of the project and majority of the aircraft project would have that, but it is tough to get 100 person definition, we need a system requirements or an ACD. So, that is the input for a software development lifecycle to define the high level software requirements, but we should add to an extent anticipate changes as well define obtaining the changes at the later end when we are following a strict waterfall approach would be very cost effective and there will be increased rework times. But in the agile methodologies as we are developing sprints, we expect the changes as we go so that we do not have a huge change impact analysis or the end of the project or at the when we are done three fourths of the project, we can expect changes in every sprint, and we can determine the impact analysis and we can fulfill the change request at the sprint level itself. So that is one of the challenges majority of the project will face then with respect to design options. When we are we can have a design based thinking as we go over the sprints and in the Agile way, instead of just having that is one of the big advantage of using Agile we can have for do 178 projects, we have different levels of testing, which is high level testing, low level testing, system integration testing, and we can begin writing the test cases and when we start having the requirements and we can collaborate more with the requirements team with the development team work with the testers. So this can happen during the sprint based agile based development. And what would happen is this was lower the risk of having heavy task of lower level low level testing at when that milestone is hit and when the low level test cases or statement coverage, decision coverage, have lot of bugs or coverage issues, then it comes to the development team and they will have a report churn have rework or the later of the project. So this can be avoided when we have this agile based development approach. So as I'm explaining each of the challenges that I have faced, I'm also giving a perspective of a small solution for each of this which we can we will be I saw as explained we will be taking some project examples and going through in subsequent sessions with practical example. And also this way of having multiple design options improve the efficiency, because we are having constant design based thinking and architectural based thinking as we are going through our multiple sprints. And like I explained the first point our scope changes. Generally in any aerospace project, if we get a scope change, and let's say when the project is three, four done, we will be really going back and doing this heavy change impact analysis and then going to the first step and identifying from the system to the high level requirements to the design. So these four changes would be most welcome in the agile methodologies as we go through the sprint and we can do it the sprint level or a particular miniature level of scope changes would be easier to handle rather than taking doing scope change or change. Ctrl at the rear end of the project, and then obviously improvements in all of these would help us to improve the schedule. So, we would not have a huge schedule impact.

30:14

I will I will explain somebody has a question on the documentation, I will explain schedule impact and obviously, it will save a lot of cost when we are taking care of all these rework and it will also reduce the release interval. So, the customer need not we can have a frequent even in the in this we can have prototype delivery to the customer within a couple of Sprint's so the customer need not wait for delivery at a particular milestone and then come back with our feedback because a lot of these feedback delivery and feedback takes a lot of time with respect to aerospace project. And that really, really is integral and that time can be saved. And that would have an reduction in the delivery delays as well. So these are different approaches that personally I have followed in all in my aerospace project, and that had reaped good benefits for me, and as simple as a daily standups. So when we have our daily standards with the requirements, developers and the testers, we have the interaction between the team

collaboratively so that has results lot of questions that we have with the team, so that we don't need to wait for the weekly team meeting or the milestone review, beat CDR PDR, we can still have those milestone reviews with the customer. But if you follow these methodologies, the open issues we get from all these reviews would be tremendously reduced. And it would bring a lot of customer satisfaction when we are doing this way. And even people who are familiar with the stages of involvement audits, we can have even sprint based on these audits, and have prepared ourselves in multiple sprints for this audit and not have one audit with the customer or the airworthiness authority at the end. So these are the personal experience and personal implementation that I have done in all of my aerospace projects, and seen a good benefits out of it. So that is one of the main reason for today's session sharing those experiences with all of you. And other things about following Agile is visibility. Generally, with any waterfall approach, or any aerospace approach, the visibility to the management are two, I have played the role of engineering manager in many of my projects, I've been an engineering manager, the visibility would be very less when the people are working in silo and working towards a particular milestone. But with respect of introducing this agile in that approach, the development process itself would be transparent, and management and people can be involved in the development process, and the visibility will be higher. So there would not be any surprises when there is a customer delivery, when they and they will not be surprises when there is a management meeting. So that brings a lot of transparency in the processes when we are working. And of course, as I explained before, it reduces a lot of design risks and implementation risk, and improves testing efficiency as well. And if you think about all of this, this obviously will improve the quality of the product delivered in improvement in scheduled improvement in cost and overall customer satisfaction as well.

33:51

All right, thanks for that Kritika. So now that that essentially ends our discussion for today, so we can open it up for questions about what we just talked about. So feel free to chime in. Again, this is an overview of, you know, the overall constructs of building aerospace software using agile methodologies. And this last slide here kind of contains all the problem areas that we typically see on an aerospace project. And then as we as Kritika mentioned, we're gonna have another couple of more webinars after this that go into how you address these particular challenges using Agile techniques. And so that's what we're going to but we could still talk at a high level about these and kind of give you a little taste of what we might cover in these next webinar sessions. But yeah, feel free to ask any questions right now that you have about the content that we just provided. So critic, I see one question that just came through it says, How is the documentation taking care of when adopting the Agile way of project development? And how is it different from adopting different different life cycles? And I am sure airborne oops, give me one second here. I'm sure airborne systems need a good level of documentation. So how do we address documentation from your experience critica

35:19

in my projects, if you see Agile methodology, right people think of course the big difference between Agile methodology and aerospace project is of course documentation because Agile is more of delivering the working software than being heavy on documentation. But that part we need not be very strict. How I have followed in my project is, is let's say Sprint Zero or the first my planning sprints I would have when the when I get the system requirements and when I have the requirements team beginning to analyze and look at the requirements my development team will begin working on the

planning of the project. And also if you take CMMI as well as in DL 178 CMMI also requires project planning and control that is a key part of CMMI. So I have user stories and definition of Done and acceptance criteria define for my plans as well. And then I have my verification team who will be having the review checklist as part of their definition of Done and user story of acceptance criteria for the planning review. And my first couple of Sprint's, while we are analyzing the requirements while we are getting ready with a high level design would be focusing on my project planning phase. And that would be my delivery. For example, we deliver potentially shippable increment to the customer per agile, my potentially shippable increment for my first couple of Sprint's would be my planning documents, because that is necessary for any Agile software development. And also, what we follow is we call something called as requirements verification matrix. And where we would show the bi directional traceability starting from system requirements to high level requirements to low level requirements to code and to the test cases and backwards. So all these can be achieved with Agile tools. Collaborating with requirements management tools, I don't want to mention any specific tools here. But agile tools and collaborating with requirements management tool can accomplish that. And in those cases, what your requirements document or design document, when you have clearly defined user stories, clearly defined acceptance criteria, you can these documents would be an export from those tools and review of those document, because one thing we have to be very clear about aerospace project is if you're using a particular tool for any of the methods, the tool needs to be qualified, so but we can do a manual review of what the export from the tool and it can be delivered. So your first couple of Sprint's would be focusing on that. One thing what when we are falling any methodology is we have to make that methodology work for our business needs. You know, that is the most important part of implementing with Agile or any other methodologies. Make sure what our business goals are, and how we can better or best use this methodology to fit that goal and obtain results. So that's that have been my philosophy while using a giant for this.

38:40

Thanks for that. Yeah, just to kind of piggyback on what you just said that, it sounds like that, you know, you're using the existing methodologies agile, the tools that we commonly use, and the artifacts actually talked about having backlog items for the documents themselves, which then these doc, this documentation is a deliverable, you might consider that it's a non functional requirement to deliver aerospace software. So it's, it's not functional, but it is it is a requirement is a deliverable that you need to have and you manage the work of that just like you manage the work of anything else. And it sounds like you're using the tools that you would normally use in your agile development. You're using the same artifact types and you're managing the work that the very same way. So very smart idea. I can see that. Definitely. So that's a great way of looking at it. I have another question that came in. It says, Is there a customer representative and your daily stand ups? Do the teams have access to the product owner to clarify requirements, discuss trade offs and etc? They didn't hear that a product owner was participating in your daily stand ups.

39:45

Okay. Oh for aerospace projects, we have systems engineer who would be our own. We don't have customer representative in our daily standards because customers would Be by a weekly collaboration with the customer, but we have systems engineer. And we have our internal product team, who would be the representative from the customer side. And of course, as an engineering manager, I would also

be the voice of the customer and wise of the team on both sides for getting the input and in case of any questions, taking it back to the customer. But having a daily standup with the systems engineer with the product team and the development team definitely brings in lots of questions. And as I said, one of the things that I missed probably wanted to explain is, this also helped us to do a scrum of scrums.

Because my hardware team we are product based company, my hardware team and my Systems team. Were even though they don't follow 100 person agile, but they we would have do a scrum of Scrum with them when we have any questions with respect to the software that we need to take it to the Systems team or take it to the hardware team. And that happens in a lot of collaboration. And majority of these questions will be sorted out with these with these methods. And apart from that when there are if there are any questions, we would definitely have customer involved. And we would have weekly meetings with the customer as well we used to have technical interchange meeting and also project status meeting which the team would participate with as the lead and the managers would participate. So having, for example, there is a requirement that has hardware dependency that that has an hardware dependency, then having the scrum of Scrum would help. I saw there's one more question is engineering manager performing the role of a scrum master. Now I do have Scrum Masters reporting to me, but I am heavily involved as well. If there is for example, my tool is automated, if there is an impediment, if there is an issue, I get an email and I do participate in my daily standards. I'm not playing the role of the scrum master. But I don't mind when when the scrum master is not available. But I do have scrum master playing the role. Since I'm presenting these ideologies to you. It looks like as I'm playing this Grandmaster, but I do have some monsters.

42:30

One of the one of the patterns just to comment on that question, too, that I found when you know working in the, in the software development environment for airborne systems was that the quality someone from the quality organization was a good representative was a good person to use as a scrum master it because of that, you know they're being they're providing the guidelines for meeting a lot of these requirements for development, making sure we're able to meet, you know, FAA requirements, so you have an accountable person that's really driving the team towards meeting some of these requirements, not just the functional requirements, but the non functional. And so the scrum master, making sure we're driving to again following our practices that we should be following from a CMMI approach from idea 178 approach. And so having a scrum master kind of really being held to being accountable to making sure we meet not only the deliverables of the functionality, but overall the non functional areas as well getting being able to get the software certified. It's a good it's a good you know, part of the organization to pull a scrum master from from being that having that done being accountable for making sure that the whole program meets that criteria. So,

43:54

question why because even part of our organization we have quality representative, because the entire concept of do 178 also right quality configuration management and certification if you see we call it as an integral process, which means it goes hand in hand with your development process. They don't go do your development, but you have to keep the quality perspective configuration management and certification political perspective in your mind when you're going through your complete development and verification process that is why they are called as integral process and do 178. So the quality quality will not be an afterthought, if the quality personnel is involved and participating and

understanding project planning perspective and for example, the 178 requires Software Quality Assurance Plan software configuration management plan. The quality personnel can also have a user story and configuration management person can have a user story defined and can start producing the deliveries and generally quality person will look But the evidence of our reviews and all these would be defined in your plan, a percentage of your revenue checklists, your test cases will be reviewed by the quality personnel and signed off. So having them being part actively being part of your project and they, the quality issues would become less and improvement and quality can be seen. And one other thing, of course, we can take, take this all in the next sessions. Having configuration management change control is a very, very big part of the Oh 178. Configuration management is a very, very big part of beer, 178, and CMMI. So all this having an iterative approach and a change control tool. And implementing this in Agile would help us in the long run of change control process and configuration management process. So those issues, because the most costly mistake anybody could do is delivering a wrong configuration in downloading alarm configuration and working all those can be sorted out if the configuration management person is also involved, and change control can be heavily involved part of the development lifecycle as well.

46:20

All right. Any more questions from anyone in the audience here wants to ask a question.

46:33

You weren't here. I have one question critica. That around independence and testing. We typically say in an agile way of working that. We want it to be collaborative, that if you do have separation of duties, between development and test, and you're trying to keep them more or less like in a silo. And that's, that's really an agile way of working, where Agile is very collaborative. How do you reconcile the deal when 78 requirements of independence and testing and having the necessary collapse and unnecessary collaboration with those two groups to produce a quality outcome? You have any thoughts on that?

47:28

Yes, definitely? That's a good question. Because the when you have the independent testing involved, right, they, you have to begin thinking about your verification environment, and you have to develop awareness. We call it a software verification and validation plan. Actually, at the beginning stage of the project, it can change as we go. But having the test team involved with the interaction about the platform about this test setup, and having a test based thinking at the beginning of the project definitely helps. And that's how I have led my project as well. And also when the developers start, when the requirements team starts writing the high level requirements, and the test team can begin thinking about test cases, and begin raising questions. When the developers are writing their requirements and acceptance criteria, the test team need to understand what the acceptance criteria is, and this should be an independent. In my organization. It's an independent team working on this, and collaborating this with sort of lot of questions and lot of issues. And especially when we begin to work on low level testing, giving constant feedback, raising constant bugs and implementing the bugs would help in the long run. And somebody asked a question,

48:53

there's a question here. When we when we say quality personnel, are we referring to a separate group responsible for creating bugs, regression testing? Automation? Let me look, can I take a stab at that? That's kind of what I'm passionate about that question. The reason why is because in the aerospace industry, and actually in highly regulated industries, I think the word quality means something different than in a lot of other industry in a lot of other industries and those using agile and word testing. So testing, you know, testing and Bug, bug fixing and things of that nature. That's an that's an engineering activity. That's not a quality activity. Quality when we're speaking of quality is do we have a, a, do we have a testing process? Do we have a development process? Are we are we building to that process? Is it is it a quality process? So there's checkpoints can we measure are we being effective and developing in the same way that the same time do we have the review is the change requests, processes in place the configuration management approach? That's when we're talking about those kinds of aspects in terms of how do we have a quality process and way of managing the work that we have to do in general, that is quality, that there's that we have some practices in place that we can identify that that's how we do something. And then we can show that yes, we do it that way. And actually, we can measure how we do it, how well we do it until we can improve upon it. There's a quote from Guru and quality, the total dogs kind of the author's Total Quality Management System, and we're Deming and kind of the godfather of lean, he actually he made a statement that always sticks with me is that quality, must quality, the word quality means that it must be specified means that we need to define and create like a standard, essentially, this is how we do things. So then we can actually measure ourselves, how good are we at doing it? And that goes, you know, so the typical quality probably that most people think about the testing aspect is, how many bugs do we have in our code? I mean, that's one aspect of it, you know, how good are we at creating, you know, code without bugs, that's one aspect of it. And you know, but that's not the entirety of what quality means. And so in aerospace industry and other highly regulated industries, quality means something way more, it's how we, how we do things, and how we measure ourselves that we can actually improve upon that standard in which we do things. So want to point that out. So yeah, creating bugs and testing that is a part of quality, but it is not as just a small aspect of it. That's what that's called quality control on the product, not quality assurance of our overall, you know, program that we have, we have a quality program, because yeah, you can't get you can't work with an aerospace company, unless you can assure them that your programs will be operating in a quality fashion. That means then, of course, possibly you're going to deliver quality products to them. But you need to have a quality assurance program. diacritic. I'll turn the next question over to you.

52:10

Quality Assurance actually comes begins from even when we begin to build the project, it starts from planning, it goes up to not only delivering the project, but goes to maintenance, you know, as you explain it, testing bugs, regression automation is part of it. But it is not completely it is like are we up to our standards? Are we complying to the processes? Do we have a process? Are we complying to the process? Are we measuring, if we are complying? And do we have metrics to track? And are we repeatable? Are we coming back? And we can do the same thing over and over without compromising on the products? Quality or deliverables? Yeah, in the aerospace and highly regulated world, the good quality is more than regression bugs and testing. All right. Any other questions?

53:07

That's about it. We got about five minutes. Look, I know there was a there was a, you know, in the effort of starting to wrap up. I'm just going to kind of jump to our last slide here. But in the end, there was a question that came in earlier about this session being recorded. So yes, it will be recorded. Those that were in attendance to the webinar, you will get a link sent out to you for the recording, and you can share it with your colleagues. And as mentioned at the beginning, that this is going to be a serious, so those problem areas, we're going to go in depth and critical. You know, we have experienced actually working on these projects. And we could give you some advice on how we solve some of these challenges using Agile methods. And still maintaining the requirements of CMMI and do 178. And getting your software approved by the FAA ultimately is what the goal is and having it be released. And so we are going to give you some firsthand examples of how we did that in some of the upcoming sessions. So with that said, Get tickets screenshot of this. And you can, you know, follow up with critical UI with any questions or feedback. And you know, we're definitely open to helping you on your project and giving you some insight. But it's critical you can

54:29

I have not mentioned my my website yet, because it's going live this month. And so I will then give my website and my professional contact, but this is my cell phone number, and I'm in LinkedIn. So any questions that anybody might have, even after the training, feel free to reach us out. To Charles or to me, that's my email and my phone number and you can reach me out at LinkedIn. And if you want us to AOL Have an offline session with you are any implementation questions or you would like to collaborate with us on any implementation please, you can reach us out. And we are available.

55:12

All right, thank you. So and thank you guys again for joining our session. We look forward to seeing you at some upcoming sessions. Everyone, you guys have a have a great day.

55:25

Thank you, everyone. Thank you.

55:27

Bye bye. Thank you. Bye bye