

Agile Methods for Developing Safety Critical Airborne Software, Part 2

Wed, Oct 12, 2022 9:03PM • 54:43

SUMMARY KEYWORDS

software, cmmi, project, agile, requirements, plan, process, planning, release, documents, epics, planning process, compliance, product backlog, sprint, iteratively, milestone, dl, tools, maintain

00:00

Alright, we'll go ahead and get started, then maybe, if not, we can always post this recording, and the on LinkedIn and YouTube, of course. So we're gonna start our second round of webinar topics on this particular topic of agile methods for developing safety critical airborne software. So I have my esteemed guest here, Kritika Prasad, and she's from the excellence group. And both of us, she and I have both worked in this environment with developing aerospace software. So it's very familiar and close to our hearts, because we, we face the challenges of trying to build aircraft, airborne software, while trying to still maintain a good agile process along the way. So we're gonna have some good to be able to share with you. And this is good this particular webinar topic is going to be it's going to go on into the future as well. So we're going to cover one aspect of it today, kind of give you an outline, the areas that we're going to cover the Capability Maturity Model, do 178, and how it aligns with Agile methodologies. And if you do recall the last webinar that we gave on this topic, we talked about big picture, what are these particular areas of compliance, the Capability Maturity Model, and what it asks for in terms of a process that you must maintain in terms of being a mature organization to even get a contract to build aircraft software. So you need to maintain some compliance level around that, as well as do one. So gate is specifically designed for safety critical systems. And so from the entire software development lifecycle, DL 178 gives you guidance that you must maintain to have a robust way of producing aircraft software, especially safety critical systems, so that you have high quality assurance built in into it. And then the third area that we covered last time was how do these two areas of compliance aligned with the agile ways of working. And so as we know, agile, iterative way of working, that is very aligned with the traditional engineering process for maintaining good feedback loops to make sure you're course correcting the direction that you're going. So we know that Agile is a highly effective, high quality way of working. But we typically don't see it as aligning with these traditional compliance areas like dl 178, and CMMI, which require a lot of documentation. It seems like there's a lot of big upfront planning, a lot of waterfall, like ways of working, so how do you merge those two worlds, still trying to be agile, maintaining that iterative way of working, you know, maintaining the value of getting fast feedback, and continuous value being produced, still trying to live within these compliance driven areas like CMMI and DL 170. So with that said, we're gonna turn it over to critica. We're gonna really focus this particular webinar session, though, on the planning process, that both CMMI and DL 178 require for any high, highly critical safety critical software. So critical Oxford over to you.

03:26

Thanks, John. Hey, hello, everyone. We met before for the part one of the session. And you all know me, I'm critical from excellency group and I have 18 plus years of experience in this field, specifically in aerospace and DL, 170 and CMMI. And 10 years back, I started bringing agile into the picture of how it can be merged and still be compliant to CMMI. And following the guidelines of the 178 C. So this is my experience of how I make things happen. I'm sharing the best practices, so it could be an informative session for everybody. So this slide shows a brief overview of what CMMI is, it's a model, which provides gives the maturity of how we develop and refine an organization's software development process. And that maturity level goes from one to five. And organization can be one two, by default, they'll be one they can be 234 or five, depending on what process areas we meet. The 170 80 is the main pillar, which provides the guidance to produce software for airborne system and equipment that perform its intended function with the level of confidence and safety that complies with the airworthiness requirements. If you see the Capability Maturity Model and do 178 C gives you the process areas here and the criticality levels and the guidance here but would not tell us how it needs to be achieved, it can be achieved in the waterfall way it can be achieved in the iterative way agile way, it doesn't specify how one thing needs to be achieved. So it's up to us, the staff, managers and organization to define the methodology to achieve these things. So we can choose the agile methodology as well. Agile is as we all know, it's a collaborative decision making between requirements and solution teams, and it's an iterative approach. So, these two things can very well work with Agile methodology and we are going to take you from one process to the other and show you how it can be made work. Next slide. Okay, so, this slide, the left the left part is do 178 See the processes. The first is a software planning process. The next is the software development process. In the software development process, we have the requirements software requirements, process design, coding and integration process in the software verification process, we have the reviews and analysis, software testing. And then we have the configuration management quality assurance and certification lies in process, if you see the verification configuration management quality assurance certification license in the DL 178 C world is called as an integral process, because it needs to go along the moment you kick off the project, the configuration management, quality assurance everything should be in place and we have to follow those processes. And then the moment we start thinking about the requirements, the software verification process should also be thought of and the moment we kick off the project, the certification lies and all such need to be thought of. So those are called integral processes. And in the center is the Agile way. So let's go to the right side and look at the CMMI I just took the CMMI level two as an example as a basic one, I didn't want to go to level three level four into complicated one. So level two process areas or project planning, measurement and analysis, requirements management configuration management, project monitoring and control process and product quality assurance supplier agreement management, and if you look at the deal 178 C flow and if you look at the CMMI flow, there are a lot of things that we have to think about in the maturity of the organization when we work on this and the second one is the agile methodology. And here we have release planning epics features, user stories tasks, if required, then we have backlog refinement estimation sprint planning, and once we start the sprint, it will be designed build test and then we should come up with a potentially shippable increment and then sprint review retrospective and then we go about with our daily stand ups. So, this is just I want to show a pictorial representation of what are the processes in these and what are the methodologies in Agile we follow next lecture.

08:03

So, the deal and 7080 planning process first and foremost step for any deal 170 80 planning process is of course, this is not a deal 170 at training session, but is to understand analyze the objectives of the planning processes. And example, we need to know what are the development processes, what are the integral processes, which we spoke about, we need to know what is the relationship between these processes and also feedback mechanism about these processes. If you see Deion Sanders, it says talks about feedback mechanism into these processes and what is the software lifecycle environment for this what are the standards we would like to follow? And what would be the software plans and how we like to comply with these plans? And how are we going to revision these and develop the software plans? The next is in any do 178 See process area is the activities and specifically for any effective planning activity is a determined determination of how we are going to perform it right. The first is we have to identify the software plan provide direction to the people working in the software project and then we have to know what our standards are what are the methods and tools that we are choosing for that particular project and then we have to identify how we are going to coordinate between the software development and integral processes and and ensure the transition criteria from one phase to the other and what are the external dependencies we have what are the if we have going to have a user modifiable software. So, these are some of the examples of things that we need to consider in any dl 170 HC project when we are working on so first and foremost we have to determine the analyze the objectives of the project, then planning process then we need to understand what are the activities need to perform so So, once we know the objectives, once we perform the activities to meet the objectives, then the next step is to the output you're generating from those objectives. And in case of planning process, it will be your software plans. And in the 178 C word, it will be PSAC, SDP software PSAC is planned for software aspects of certification, SDP software development plan, SVP software verification plan SCMP software configuration management plan and SQ AP Software Quality Assurance Plan, so those would be the outputs. Lectures after chance.

10:39

Alright, thanks. Good. So yeah, I just wanted to, you know, before we dig a little deeper into the actual requirements of how you would meet CMMI, or do 178, I just wanted to have a level setting slide here about what agile planning looks like at the various levels. So, at the highest level needs, sometimes they call it the onion, the planning onion, you know, kind of can peel back the layers of the different levels of planning. So at the highest level, you have the vision. And so if you think about a safety critical software project, you typically get an ask from, you know, say a big airline company is asking you to build a software, you know, your company has contracted to build this, this particular software modules or system, and you get a big list of requirements, you get a requirements document of some sort, from the big aircraft company, all right, that that document that ask or that project get started with some overall vision, you know, some overall system as being identified major capabilities of the system that are required. And even sometimes, you know, you might even get a requirements document specifically from you know, from the the, the actual company about what they want, what they need for this particular system to do. So still yet we we identify, you know, this, this initial deliverable that we get from, from our big aircraft company as a vision and kind of a roadmap of when they need the roadmap at defining the milestones and delivery dates, whatever, we contractually have signed up with that aircraft company to deliver to them, we've gotten that kind of contract and the overall project scope identified upfront at that point. But in order for us to actually get into the details of how we're going to

actually deliver on this, we need to go one level down. And that's kind of where we kick off, which is called the release planning phase of your traditional agile planning onion. And that's kind of where we start to meet the requirements to compliance requirements of meetings. The vo 178 requirements and the CMMI requirements is typically at this level, is that this release planning level. And then you know, further down into the details like code, deliverables, release deliverables, all that kind of, you know, and change requests, we're gonna get to that in a second, or more handled at the iteration and daily planning levels. But but in terms of the upfront, it's typically called a quote, air quotes, big upfront planning that we typically associate with highly compliant software programs. And agile, you can handle this big upfront where it's not as big upfront, it's still iterative, because there's multiple release planning intervals that you go through. But we handle this, traditionally what's called big upfront planning at your release planning level. All right. And then, as you can see, just kind of giving you a pictorial of what that looks like. And critical is going to explain more the details on how you actually meet some of these requirements with release planning. But the overall project vision and scope into contractual delivery dates, if you will, are captured in its initial vision. And then we produce a high level backlog, product backlog as we call it an agile, the product backlog is, you know, all the capabilities essentially that we need to deliver on the project. And then the release planning process is going to incrementally show how we get this done over time. Okay, and that contains the details of the actual deliverables over time and the various milestones that we're going to reach over time in terms of the development process, is how that evolves. So it's not all big upfront, it's still iterative, and but we use the release planning process to break down that big chunk that big rock of the project scope upfront and so we can interleave iteratively create our compliance document and update them over time. Alright, so critic ology tarot readings, kind of more detailed understanding of how release planning meets these requirements.

15:01

Just go back to the previous slide, I just add how I did in my previous projects. So ideally, I'm just thinking from the do 178 And CMMI perspective you know what, like Charles said, Right we receive system requirements from a customer saying hey, this is what needs to be developed, we have a system team, we have a software team, and we are collaboratively working with each other and the other team might be agile might not be agile. But let's say I'm just giving an example let's say the software team is agile. So once we get the system requirements, we recall it as system aspects of software certification come up with okay what from the system requirements come up with the requirements verification matrix at high level I'm talking about which the system engineer would be doing and then assigning okay out of these system requirements okay, what all falls under mechanical, electrical software, some requirements could be both you know, so, that kind of alignment will be obtained and then we will have discussions with the customer which very much agile would focus also on customer interaction. So, we will discuss with customer and understanding okay, what is your expectation we received the system requirements. And then, which we in the Agile world can call that scrum of scrums with the system team and other teams to understand what is needed for the software team. So once that is obtained, what I do from let's say my software product backlog is take those inputs that I received and put it into my vision for the software perspective. And this can be done at the system perspective also. But since we are focusing more on the software, I would like to take software as an example. So I will put it into my software product backlog and think from the perspective okay, what is needed from the software side, okay, these are the requirements that we received that we need to fulfill Okay, from the day 178 perspective, we have to now we got the system requirements, we got

the system aspects of software requirements, and then we determined the criticality level of the software. And we need to kick off the software planning process and then we kick off the project and then we kick off the release planning. And in the release planning, we add epics, very high level epics of what is needed from my immediate milestone, okay, I need to do the software planning process for agile and as a whole also, in any organization, we need to have an overall plan of how things need to be done. In some organization, it could be single plan comprised of all software project, but for the 170 project you need to have, depending on the level of the software, you need to have a set of plans for your particular software project. So then my epic would be okay, I create a PSAC for this system. And these are the inputs, this is my acceptance criteria, your acceptance criteria for that epic could be my preset reviewed by a peer review or software manager and moves to the next level of review or submitting a draft to the customer. I'm just giving us an example. It could be anything that we want and then go into the Agile estimation process. And then in the Agile estimation process, we go add okay, this is the high level all these plans I need to generate this is my epic, and how do I go into my features? Okay, these are the features are and when I'm working on this, the engineers I myself I'm talking about let's say the manager is working on the plan or some leaders working on the plan and some engineers can start thinking about okay, we already received the software requirements, what is the design, what is my development environment, I need for this what is my verification environment I need for this because all these details have to be documented in the plans. So, so, all of these perspectives and all of these start classes can go into your epics can go into your feature can go into your user stories, and then my product backlog can be built for my first milestone and a high level release backlog can be built for example, if I have to deliver a particular URL level of software, critical critical flight critical software or in some terms, we call different labels right different colors. So, if I have to generate that will be my end release for that and my intermediate release will be or each of these milestones. So I determined a complete release planning and I determined my immediate milestone planning and then from my milestone planning, I come into my sprint planning. So pretty much that is what I tried to say with this slide. And I just now gave an example of how the thought process can be thought and when you're thinking in this process, obviously automatically or CMMI things a lot of things will be achieved. I will I have shown that is my example. Next, we'll go over that next

19:57

slide shows Okay, so whatever I explained in my previous slide, pretty much I just gave the yellow boxes here are just examples of how we need to think about the project. Because we have to come out of the mindset of Oh, agile wouldn't work for any space project. Now, anything can work for any business that you want just you need some tailoring in and out of it. And some add ons to what is needed to accomplish. So first, as Charles explained, evaluate your product vision your product vision can be if an organization is working on that complete system, it is entire system for an aircraft, and your part would be a software for that product. So it could be that that is your product vision, then I'll elaborate your product vision into backlog items like what I explained, okay, I need to any big aircraft customer, right, we can have PDR, CDR PPR, and we can also have sol one sy two, so a tree like that. So data, mind your complete vision of your product backlog and determine your milestones that you need to achieve. And then once you determine that, then go add all the epics, and all the user stories. And if you have a group of people working on it, let's say five to six people, so determine the activities, what the developers need to start thinking, what's the verification team needs to start thinking about the lead need to start thinking, and then coordinate all this and prioritize all of this in your backlog. And then

that's my third point, prioritize, create feature user story and based on the deliveries, and then set your release goal, my first release goal is to start working on the plan planning process, and then deliver a draft plan to the customer to obtain that I need to do internal reviews of that. And then at the same time, you have to start thinking about received requirement verification matrix. So what will fall under your software category? What would be the verification methods of that? So all those goals need to be set? And then you break down into your sprint goals? Okay, what would be sprint one goal to goal three goals like that. And depending on the feedback, you achieve, you tailor those goals to what the next sprint would be. And also understand from your retrospective of what it should be, it can be made better.

22:12

Excellent. Okay, Charles, do you want to say something or I can?

22:21

Sure, sure. I'll chime in here. So yeah, this, this slide is just, you know, wanting to really emphasize the complexity from the Scaled Agile website. And being that even though you're in a compliance driven organization or industry, there's still an opportunity to use agile methods, and how this continuous or iterative approach to meeting compliance requirements. So, you know, there's you guys, again, we're kind of stuck in a mindset, oftentimes, that you have to do, you have to meet these compliance requirements in large batches. Right, or, at the end, you do the entire thing at the end. And, you know, again, the waterfall mindset is kind of the is the way that we think about these things. And we don't typically think that we can do these things iteratively, CMMI, nor do 178, even though they're asked for a lot of documents, ask for a lot of compliance, areas to be fulfilled, do not dictate how you go about creating those artifacts to show that you've met those compliance requirements. So that's one key point, there's no, they're not telling you how you need to go about doing it. And actually, I want to give a kind of a testament to the reason how I even became an agilus. In the first place is in this particular industry, I was very, I was very fed up with the fact that the documents that were being asked to produce, and these, you know, do whatever nuclear example that you know, it was there completely artificial documents that were being asked to produce at certain stages of the waterfall process where they were meaningless. I'm like, Well, why am I why am I going through this effort, this time and overhead effort to try and get this document together, and try to get into this final state if it's never going to actually reach that, you know, point until way, way later in the development lifecycle. So let's just have a process that aligns around this idea of how we incrementally and iteratively produce these requirements, these these compliance requirements. And so because it's the natural way that we go about evolving a project in the first place, so there's a system here called Lean QMS. And if you're interested, anybody can go to the Scaled Agile website and actually see there's kind of a there's a method and a structure around how certain certain challenges like this had been solved in the past and some case studies out there with other companies, large companies that have used that iterative process to meet requirements, compliance, regulatory security. requirements, you name it, on how to do it in an iterative way. So just wanting to really level set this idea about continuous compliance is the best approach that you just don't do it all in one batch at the end. And as you know, that could get you in trouble. You know, you, if you try to do it all, at the very end, it's probably something that you didn't capture early on. So that's a, it's want to make sure that we're understanding this process of continuous compliance, that it can be iteratively done versus being done all the time, because our focus here is

planning, right, we're trying to meet the planning required, how do we meet the planning requirements of deal 178 And CMMI, traditionally, you'll say big upfront planning, create a, you know, a big project plan that captures everything offered detailed, you know, details and degree of detail that your traditional project plans might have. But that's not the reality in an agile environment, right? This iteratively evolves over time. And so that's kind of the way that we work naturally. So it's, you know, it's kind of base our projections are planning based on reality. And that way, we can also produce our documents in the same way based on the actual reality that's happening, and still meet our compliance or regulatory requirements. With minimal waste. That's the thing that you're trying to avoid. Because if you do things prematurely, creating all this big upfront planning prematurely, a lot of that's probably going to go to waste. Because that's not the actual evolution of what the project is going to see down the road, we need to be more iterative and just make the real time update does what's needed to kind of follow the plan, the short term plan, the relief plans, the iteration, the iteration plans, that we know. Alright, great. I'll turn it over to you to get more details around this, this idea in this concept,

26:43

yeah, sure. So what I did was, I took a real example, that I have worked on, I didn't mention the project names, and I just gave an example of what artifacts we could give as an example, you know, take any project for that matter, these artifacts could be a potential evidence or example for the CMMI practice definition. And this practice definition might be from the previous version of the CMMI. But it gives you an idea of all the specific and general goals that is expected out of how did I try to map is it the software planning process up to 178, I took the project planning process of CMMI and I took release planning process of agile and try to integrate all of those things and that's pretty much it. So, the first one, let me go over on the one establish a toplevel I explained before we have the product backlog, where you can add all your work breakdown structure and then for an overall project, you can input the effort from your product backlog in Agile tool, I come up with a project schedule, which will have of course have an overhead time and also other interdependencies established in estimates of the attributes of the work product and tasks and the software estimation process in the Agile world and our story points that should provide given a high level estimate define the lifecycle phases upon which to us the all the phases of the project even at PISA could give us all the phases of the and you can define your transition criteria, the transition criteria need not be this phase should be my requirements phase should be completely done before I move to the design phase it can be start having a baseline of the requirements and go and draft providing the draft and then going into the next design. So it can be thought of as a ballot process. So that can be defined estimate the project effort and costs for the work product and as based on the estimate rational that's again our software estimation process your product by project backlog and then any additional cost that the project might have to stick no GS and all that. establish and maintain the project's budgets and schedule the current backlog and what it would take for us to account each the release and other overhead cost and then your schedule you can come up with a release schedule and your sprint schedule and obviously determined the change. Every time you get a change request every time you have an impact then what what would the impact on your cost schedule critical for any project planning process? was a project management process. So we can maintain a pm risk register and the other day to de risk of the project, what we do we, we maintain a kind of a sprint planning checklist where we use to make sure the user stories have acceptance criteria, and all stakeholders present here in the meeting. And are there any risks identified at the project level or at the sprint level? And also apart from that, what we do is if there is an impediment, not availability

of the tool, and there is a dependency of the external stakeholders, or like the scrum of Scrum between the system and all the other departments, if there is a dependency that's not being answered, that is raised as Oh, I'm sorry. Am I audible? Just

30:51

Yes, you are, just continue on, just talk to me.

30:54

Okay. And then any impediments that we raise? If the impediment is not resolved, within a day or two, it will determine the days and it goes into the project risk. For example, if you're waiting for a tool, if you're waiting for something from the Systems Department, you don't get the answer. Within a couple of days, it automatically becomes project risk. So it's like playing around with the Agile tool and

31:21

what is there playing around with? These, how do these things be being so heavily a part of sprint, you're part of your daily standup?

31:40

I have a wrist. I didn't get this tool. I didn't this. See that gets into the sprint wrist and it goes into the project risks. So things like that. Then plan for management of project data, we identify all of the project data and the tools used for those, like your requirements. Gotta go? Yeah,

32:16

good. Yeah, I think it's pretty. Yeah, you're really freezing up pretty bad. I could actually think, okay, maybe check your network really quick. And then I could continue to offer I could continue on here, because you're kind of like freezing up a little bit. But I could pick up from where you left off at plan for management of project data. And so project lifecycle lifecycle data, as documented this document in your software development plan, and your software configuration plan. As well as on the last bullet point, there is the CMMI practice that's requiring you to plan for necessary resources to perform on a project that'd be captured in your pee sack. So as you can see, here, we actually, you know, did a lot of work for you here, if you're gonna bring this back to your company, you actually have kind of a traceability here on how you actually could meet some of the CMMI requirements and the evidence is that you are producing the documents that align to these various practices. So one thing too and like it's, you know, in a software configuration plan a software development plan in your deal 178 It points to the certain categories that you'd need to have within those plans. So So again, how to create those plans are dictated by the Oh 178 But just showing you the mapping that satisfy these areas, and CMMI actually meet those requirements as well. Pretty good you want to try it try it again. And maybe pick up on what was it No Yeah, yep, it's pretty clear Yep. All right.

34:07

This is the kind

34:10

of a big delay in your in your audio maybe get one more try it doesn't work the time No, no worries. I can pick it up from here but try one more time.

34:24

Okay, now

34:28

I can hear you but I think there's still a big delays possibly that you might three free once you give it a try. I was Yep, we can hear you.

34:38

Okay, um, so basically is whatever example

34:45

Hey, create critical maybe turn off your video and then I think that will do it. Turn off your video, maybe some of your bandwidth. All right. Yep. Yep. Go ahead. Okay.

35:02

So, basically what the matrix shown here is, whatever the methodology we have whatever tool we have, how do we utilize it implemented to obtain or comply what is required for the project bd 178 are CMMI. So, some of the techniques that I have used is coming up with the sprint planning checklist coming up with a sprint review checklist, so that we don't miss thinking about the race the impediments and also obtaining the commitment from the stakeholders. So, all perspectives of the sprint and a backlog is thought about and also the first one plan for knowledge and skill needed to perform the project what we ask us at every sprint meeting, do we have enough resources does the resource have enough skill set to perform the task. So that is added part of the sprint checklist plan the involvement of identify stakeholders, this is documented in the piece of who the stakeholders are, who the team members would be. And also we ensure in a sprint meeting and in our scrum of Scrum all stakeholders are available so that we can obtain periodic feedback and not obtained peer feedback after we delivered something or after we have done with a particular milestone so that it makes our iterative process much better. And the next one is establish and maintain your overall project plan content and of course, our agile tools or the Agile process that a lot of scheduling burndown release details and software plans for DL 178 P sock STP does the work, review all the plans and for that to CMMI requirement. And of course, we have planned review process for DL 178. And that can be incorporated in GA, reconsider the project plan to reflect available and estimated resource. So as explained to you before, and every sprint we ask this question about the resources and who's available, what is their capacity for this particular sprint. And we will revisit that often commitment from stakeholders stakeholders commitment I As explained as obtained during our release planning and sprint planning, and establish and maintain an organizational policy, like Charles explained, in addition to the sprint, and in addition to the plans generated for D 178 are the processes, we need to have an organizational QMS and the processes associated to ensure following the processes and making sure we are generating what is required for the organization. And then establish and maintain the plan for performing the project planning process. And that's exactly what we've been discussing. So far. It's a software planning

process. Next slide shows Yeah, yeah. Um, do you want to go over configuration management or pause between for q&a?

38:00

Yeah, one more slide, actually. So we're gonna say configuration management for our next webinar in this series. So it's, you know, goes in pretty much in depth into a lot of the those topics. But But one thing that we do want to end on before we open it up for q&a is just to kind of revisit, what are the key Agile artifacts that capture all of those areas that are critical just went through with CMMI and DL one, seven, yay. So a lot of the planning requirements are captured in the release identification through the release planning process. So the team increments and what are those iteration goals and those goals for each iteration that are going to be accomplished throughout the course of a release plan. So that is a key artifact. The definition of done is a highly valuable artifact, you can see I even have that highlighted in green, because a lot of the details, quality requirements to down to the code level, quality requirements, like code reviews and things of that nature, are already captured. Those are not your definition of Done artifacts at the story level at the release level. In a minute, the epic level, there's certain definition of Done requirements were like Kritika mentioned that there's certain epics, you know, having a piece sack, you know, what does that mean, when it's done? What do we need to have in place for that? So definition of done at the various levels are essentially ultimately become your quality checklist that can be audited along the way as artifacts that you've met these requirements throughout. Then your standard agile artifacts, that everyone should be familiar with epics features, user stories, tasks, the test cases that aligned with the acceptance criteria, or test results themselves as well as the bugs and the defects that get captured throughout the testing process. So all of these artifacts need to be managed that you know, and I got tracking stuff Some of your liking to JIRA, as your DevOps rally, whatever tool that you use are flexible tools to use to capture all the compliance requirements in these artifacts, accommodating change. So a big part of a big part of deal 178. And you know, critical software development is managing change, how are we tracking change that is handled through your sprint cycles. It's fortunate that sprint cycles are only two weeks, so we have a feedback loop there. But even at our daily standup, every single day, there's an opportunity to identify and capture any type of change that might be needed. And then be officially needed documents that like in a change management system, change control system, there's a process around that, which could ultimately end up being a bug or a new story on your backlog. Whatever the case, may be it actually that change can turn into an actual artifact, a traditional agile artifacts in any of those areas could be a task, again, the user story, a bug, or the whatever the case may be, however you choose to use those artifacts in your change management system. And then lastly, the software demos that happen at various levels and stages within agile development, either at the team level, or maybe even at the system, level system verification systems demo level, that these are ways to actually cross check your project management plan, your progress that you're making, on hitting certain key milestones, of software actually being, you know, completed and ready for integration testing, or maybe ready for either release. But the software demos are your true checkpoints to making sure that you're meeting these project milestones, and then obviously, opportunities to actually collect feedback along the way. All right. So with that said, we're gonna move into our q&a portion of the call, we're right at about 15 minutes, too. So any q&a opportunities here from our audience, around what we just covered today? So again, kind of topic that we covered today is the DL 178 in the CMMI planning requirements, and how do you match that with

an agile planning process? And then again, we're going to kind of go on to future topics and see configuration management is probably our next topic in this series. But in terms of planning,

42:30

I see a question for Sandeep. First of all, excellent session. Thank you Sandy. Question is what type of roles you have or recommend in this process? Do you retain the Agile roles such as product owner scrum master developer or do you train them to keep a balance between traditional and agile approach actually, we can maintain the same roles and the you know, because that clears for example, product owner can be your point of contact when you have a system aspects of certain safety requirements, he can work with a system engineer and understand the product and obtain the product backlog that can be done by the product owner just in a different project. Do 178 Or is aerospace software project, it can be done by him and Scrum Master can still facilitate all the ceremonies, just that to aid the scrum master to aid the product owner, just few methods and few templates and few trainings you know, would help other than that, we can definitely maintain the same rules.

43:47

Well, one thing I like to add to that two credit cards that I think we got maybe a similar experience and this is that your scrum master actually becomes a critical plays a critical role and the software quality assurance aspects of this and making sure that the team and the program meets these you know, the compliance requirements. So you know that that those traditional agile I think somebody's go on mute. Quickly. Okay. So your traditional doc software projects, you know, your traditional agile process that your scrum master usually make sure that we're following in meeting the guidelines of a good agile process. Now this process needs to be aligned with making sure that we meet your compliance and regulatory requirements. Alright, so so your scrum master kind of plays a role in in making sure that that happens and so it Do you have a quality assurance organization that's on top of this that the scrum master can work with? Otherwise, you can just cut off the middleman and have your scrum master the, you know, be up to speed on what these requirements are? And so that they're really driving the project and the program to meet these requirements as well. All right. Tailor Yep, exactly. Another question critical, how do you convince a more traditionally driven management or leadership team to accept or adopt the Agile approaches for Medtronic systems development? Can you please share your learnings? So, you got to

45:49

my company also was the company that I worked for earlier, if you see all the companies who've been doing software or product for years together, right, they have been used to a particular way of approach and you know, and anything change would people any company might not like because they are more more having more conservative approach. So it's like introducing the change without making an impact, huge impact to the organization is what I started following. You know, for example, it's it is called Scrum of scrums. But it is pretty much a meeting with the other department to understand the their priorities and your Trapeze and making sure it is aligned in the traditional way, they might call it project status meeting in your in the Agile way. It could be scrum of scrums. You know, in fact, if you see when you start introducing the small changes, and you bring in stakeholder involvement and organization, leaders involvement, they will seriously appreciate because you're sure you are having this transparency, of telling what issues you have, what impediments you have, what risks you see, and also giving them a

visibility of burndown of what can be accomplished, what cannot be accomplished. In fact, customer would also like it because you're in frequent interactions with customer on any criticality question, believe it or any deliveries, be it or even when you're doing PPR sol I, in my agile way, I even do a pre review a pre audit of things. And I have shown to the customer, my pre review findings and how I have they appreciate all this actually, when you say hey, I'm going to change this entire thing, and I'm going to take you to a new approach they would not be but when you slowly introduce these things, and slowly get them used to it, and then slowly train it. And they that's my approach my work with them, and make them understand what you're doing transparency, and that in fact work even with the customer, I will say

47:53

one thing I can I can answer that to Kritika is what I found is that the approach that I use I you know, just actual example, in aerospace world, the approach that I used was that the Agile way is a lot lighter weight than your traditional waterfall approach. Because your traditional compliance driven organization that's required, like you're kicked out of medical devices, same thing, a lot of documents, if you're doing big upfront planning and waterfall is planning, you have to produce, you're probably producing all these documents in a way that do not align with the reality, like I clear the reality of how the work is actually evolving. And so there's a lot of waste. That's why it's painful to in these environments to go through this compliance exercise of creating all these documents, just for the sake of creating all these documents, when you know you don't, you're not going to really kind of solidify these documents until way, way later on down the road. So let's do it in the more rational sensible way is this update the documents when we have the relevant information to update the documents, let's let's do it iteratively, which means that there's a lot of stuff that we don't have to do right now. Because we don't have the information to do that. And so when you communicate it to your leadership that way, they're like, Oh, get out and get out of doing a lot of work upfront. We know we can just actually get to do an actual work. And so I kind of, you know, lift the load, the only thing that they might actually push back on to is the the amount of meetings as our is tip, you know, traditionally more meetings, more feedback opportunities, so they don't, they see that as a pain, right? We already got too many meetings, and you're adding us to have a whole bunch of more meetings to talk about things. So that's the maybe the only pushback but we got to just give and take right? So we're not going to allow you to do a lot of wasteful administrative overhead work in documents, just you know, for the sake of going through exercise, but we are going to ask for your feedback. We're going to keep these meetings short and sweet to the point and we're not going to extra time, but we were going to try to make them valuable so we can synchronize on things. And to get feedback, because we're not, we're not asking for a lot of overhead stuff like we used to before. So that's, that's the kind of give and take, but I think they typically get it because that's, that's how you normally would go about doing things anyway, it's in an agile way, it's just that they've never traditional was a process that they were familiar with, that was structured enough to kind of lead them through this iterative process. But if you got a good structure to it, and you could show them the value of the lightweight that I think they see the benefit,

50:33

and some dps to your point, we need lots of patience, and to make any change, but it is definitely doable. And take one step at a time and take a life project and take the project towards the change. And people see the changes and successes when it comes along. And if you want any help, in

particular for your project, or anything, in particular for mechatronics or something, our contacts are available in the slide, you can reach us offline, and we can help you particularly for what your customized request for sure. Yeah, definitely.

51:18

Any other comments, questions? About what we talked about? Yep, fine deck, and recording usually goes out, you'll see that in a couple of days. But yeah, um, if nothing else, we'll go ahead and end it. Sure. Audience, anybody has anything that you guys want to ask us. And again, these these sessions will continue. You see, we just covered one part of the compliance process, which is the planning process, but we soon have, you know, configuration management, development, you know, verification and validation. And, you know, all of these areas that are going to be coming, they're going to show you some, you know, basically what was critical just went over with you, is essentially how you can manage your entire safety critical software systems process. I mean, you have it all mapped out for you. It's, it's awesome. So a lot of things that you see there is coming from actual projects. So, definitely gonna share some more valuable facts. Go ahead. Go ahead.

52:39

Do you recommend any tools JIRA workflow, how do you recommend doing roadmapping? Actually, every tool can be made to work for your needs. Because in my experience, right, I have used several tools I've used target process, that is one of the reason I removed all the tool names when I wanted to show you, you know, any tools can be made to work. I have worked with target process, I have worked with JIRA, I have worked with Azure DevOps. So any tools can do the work for you. And yeah, we can. There are tools that I wanted to cover that. In the next session, probably the thing is how to map different tools together. Because sometimes at a high level, there will be organizational roadmap tools. And then how do you take that organizational roadmap tools into, let's say, the system team uses a system, roadmap tool, and then into software? And then how to take it forward to your configuration management tool, test tool and the result? So yeah, definitely, many tools right now, right now are widely integrated, or there are plugins available, which can be integrated. There will be some additional works that that need to be done for the integration. But yes, it can be done. And we can take some examples and work on it in the next session.

54:09

All right. Well, thank you guys so much for attending. Again, let's look forward for you attending our next session. So kind of keep an eye on our our webinar calendar is I don't think it will be next month, but it's definitely in the coming webinars sessions. But yeah, reach out to Kritika or I and we will have our contact information available to you in infinite queue directly for those that attended this webinar. All right. Thank you. Thank you guys so much. Have a great day, rest of your day and see you next time. Thank you