Software Engineering - Agile Development and Scrum - 2025/07/08 01:55 SAST – Transcript

Attendees

Armand le Roux, Armand le Roux's Presentation, chantel magwaza, Chichi Masawi, Rogelyn Farley, Santelgo 01, yadav thapa

Transcript

00:05:00

Armand le Roux: Hi everybody and welcome to tonight's session. I think we're just going to give it about a minute or two just to see if some more people join up so no one misses. so yeah, we'll get the session started in just about a minute.

Armand le Roux: Cool beans. hi everybody and welcome to tonight's session. I hope you all had a wonderful Monday and you are ready to learn a little bit about methodologies. So, we'll take a look at methodologies today mainly with a focus on agile development and we'll take a look at scrum. We'll get a little bit of a more depth scrum. but we'll also take a look at some other agile methodologies as well and we'll talk about waterfall as well a little bit. give you guys a bit of background history of why all these things are there. yeah and if you don't know what a methodology is you'll find out today. So before we get started with everything I just need to do some housekeeping we need to go over first and then from there we'll get started with our session.

Armand le Roux: so you guys we'll start off the session where I ask you guys a few questions. I'll tell you guys how to answer it. So don't worry about that too much. then you guys can provide me with some answers first and then from there on we'll go and carry on with the rest of our session. So cool. some of the lecture rules is there's no use of disrespectful language in our questions. So you guys will see you can use the Q&A tab. So there should be a extra menu at the bottom right. you'll see is a bunch of nine dots in total. You can click on that and there should be an option for you to go to Q&A. So, it's your meeting It should say top meeting tools and then you can go hop into Q&A and from there you can basically just say ask a question and then you can submit a question through there. So, you don't necessarily just have to ask questions through this. If you want to say something or provide me an answer or anything like that, you can do that through the Q&A as well. So, that's just the best system we have at the moment.

Armand le Roux: So for now any and all messages that you want to put to direct towards me just do that through the Q&A. It doesn't necessarily have to be a question that on that note, if you are asking questions, remember that no question is dafted or silly and you can ask all the questions that you have. I'll try my best to provide you guys with an answer to the questions that you have. if I don't know the answer to a question or if I don't know, I'll just tell you about that. and probably go do some extra research, try and find an answer for you or either try and give you a resource that you can go look at to maybe find the answer that you are looking for. But we'll always at the end of the day try and get you to your solutions that you're looking for. so yeah, just ask all the questions that you guys have. So any questions that I can't answer for you, our support team will probably be able to answer those ones for you. So if you need any assistance on your boot camp at all, don't stress about it.

Armand le Roux: just Don't wonder about anything or worry about anything. Just ask the support team all the questions that you need to Get your answers and that will get you some relief because if you know what's going on, then there's no reason to stress, So, please do be sure to reach out to our support team about anything that you're wondering about or if there's anything you're unsure about or stressed about or anything like that. then if you're hearing impaired then it would be awesome for you to enable captions through the ibility settings of Google Chrome or either Microsoft Edge. so you can just hop over to settings accessibility and there should be a tick box where you can just enable captions. So this can be quite helpful but I must warn you that sometimes the captions just doesn't want to pick me up as it should.

00:10:00

Armand le Roux: So if something doesn't make sense to you then please just pop me a question and then I can clarify for you what was said or anything like that. yeah but it is a good tool to utilize. Then on to safeguarding. So if you guys are suffering from anything or experiencing anything such as abuse, neglect or perhaps homelessness or you're feeling upset or worried about something then you're more than welcome to reach out to us. As we do know, some of these scenarios can really cause a lot of stress for you guys and in turn cause a blocker where it's preventing you from proceeding with your boot camp and we don't want that for you guys. We want you guys to be able to work on your boot camp, progress with it as you need to so that you can at the end of the day complete the boot camp and become successful. So, please reach out to us if there's anything getting in your way and you need some assistance on that. cool.

Armand le Roux: will the 6 to8 session be the same as this one. for cyber security, there's going to be one next week that's about version control. Now, they do differ a little bit. You'll see that they run on different schedules. but it does cover pretty much the same stuff. GG. it's just at different times. if I may ask, what time zone are you in? Cuz I might be misleading you here. on to our stay the safe series we have for you guys, because we know we're a fully online boot camp, we want you guys to do a lot of research, right?

Armand le Roux: And in turn doing all of that research, you find yourself in many different corners of the internet, And this can lead to a situation where you can find yourself in areas where it is quite dangerous to be around. and we just want to teach you guys some best practices and make sure that we keep you guys safe as much as we can. so that's why we have the stay safe series. This is just going to be a few tips every week that we provide for you guys that you can follow just to be a little bit more safe online and that we can assure that you guys at least have some sort of knowledge on what to do and how to approach certain things online so that you don't become a security risk or kind of compromise yourself in any way. British time. Cool. Yeah.

Armand le Roux: so these sessions are more towards our U students but you're more than welcome to join them as well but I can certainly see how this is quite early for you to be here but yeah you're more than welcome to be here but yeah there's sessions on Thursdays you'll see that there's different sessions so the software engineering sessions will be every second Thursday there's going to be different sessions but they cover pretty much the same stuff as these US sessions Cool. For the US students, you have a lovely time at the moment. Yes, it is a good time. for me, it's 2:00 in the morning, which is perfect. so yeah, cool. B, let's go on to our stay safe series. So, cool.

Armand le Roux: This is all about guarding your personal info online. So, we want you guys to when you do have anything online any accounts and things like that, I preferably want you guys to use a strong unique password for each account. I know it's difficult to have very strong passwords for multiple accounts because it's just a lot to remember, And especially if you want them to be quite complex, Then there's going to be really difficult to remember and then you're going to have probably write them down in a book somewhere or stuff like that. So, this is where I'm going to tell you guys that using a password manager is not the end of the world and they work super They're very secure. and I do encourage you guys to try it out at least and see if a password manager does work for you cuz what password managers can do is they don't simply just store your passwords for you.

Armand le Roux: They can also generate very secure passwords for you. a lot of them integrate very well with either your cell phone or your computer or whatever system you're on and then they automatically fill out your passwords for and things like that when you do need to log in and so forth. So, they do have a lot of features and a lot of antivirus softares that you can get and VPNs as well do come with password managers. So, just see maybe you can grab a cool one somewhere, but there are some free password managers out there as well. that you can play around with too. the free ones are usually just not the best. They limit you in ways that suck kind of a lot. so yeah, have a look at what interest you the most there. So then also try and be cautious with your social media sharing. Don't share your entire life on social media and also be aware of how you can compromise yourself. I'm not here to scare you guys or anything like that.

00:15:00

Armand le Roux: but it is good to just be aware of how you're sharing things and what you're sharing. things like if you think about Instagram that geotags things, you don't basically want to give someone a trail of everywhere you're going throughout the night by posting pictures on Instagram that's geotagged basically tracking your every move, so things like that, you want to be cautious about sharing your address very publicly and things like that as so it's just good to be cautious of what you are sharing online. and be sure that the people who can see it are the people you want to be able to see it. So, it's also good to set up privacy on your social media to make sure that only people that perhaps follow you or that you follow can see your things and not everybody. So then, keeping software updated. This is actually a really really big one.

Armand le Roux: so you'll see with most updates have a part that's about security. and that's why most updates are important really is because there are security patches and updates that come with these and they patch out vulnerabilities. So what can happen is a lot of times we have applications on our cell phones or our smartphones or our computers or whatever and then these apps we don't use anymore and then they get outdated and somewhere down the line there's a backd dooror found in these applications that can be used to exploit or compromise people's accounts, right? and now you haven't updated your app. So that exploit is very open on your system to be used.

Armand le Roux: So, it's always good to make sure that you do keep your software updated. even if you don't use software that often. and probably if you don't use software very often or at all, then it's just good to remove it from your device completely. So, try to stay safe from So, fishing scams are very very popular, very big, and they're getting super sophisticated. especially with AI nowadays as well, making things much much easier. you get this cool, it's probably not very cool, but you get this AI that assists fishing scammers to basically replicate websites and it replicates websites extremely well. then they can just attach these fake websites to their fishing emails to make it seem even more legit.

Armand le Roux: So, fishing scams are getting really, really popular and it is super difficult to spot some of them, but just be aware there are some small techniques that you can take a Things such as always have a look at the sender email that you've received it from. try and have a look at the contents of the email and so forth. And never click on attachments if you feel weird about an email or you're unsure about an email and so forth. There's all these sorts of things that you can take a look at to prevent fishing scams. and we'll probably touch on that in another week in one of these stay safe series. So just keep your eyes open for that one. but there is a session where we do chat about whoops where we do chat about cyber security as well. where we'll touch more on fishing scams, the types of fishing scams and all this stuff too. cool.

Armand le Roux: And it's also good to just monitor your account regularly. So you don't want to just have things happen on your account. So maybe turn on some notifications and check out if you have a bank account or so. Just make sure you check out your account every now and then that nothing weird is happening or set up some sort of notification system where you actually see what's happening on the account if anything changes or what happens and so forth so that you can at least be aware of anything So that is our stay safe series for this week. hopefully some of this can assist you guys. But the main thing I want you guys to probably take from this is to use strong and unique passwords for each account. That's very very important. So try a password manager, see how it works for you. and hopefully it does improve your life a little bit.

Armand le Roux: but yeah, it's always a good thing to just make sure that especially when it comes to accounts with sensitive details, very very important to use strong and unique passwords. So today we're chatting about agile and scrum. So you guys might not necessarily understand my picture now, but you will understand it a little bit later on. because we'll basically chat about scrum and scrum involves doing what's called standups. and you'll see they're actually quite useful.

00:20:00

Armand le Roux: they might not seem useful at first but once you get to the nitty-gritty then they're actually quite useful and they're pretty good things to have. okay so we'll see what they mean by standing for 15 minutes and how we can get more effective by doing with these sessions, what I like to do is I like to ask you guys, a few questions first, just to get your mind in the right direction of what we're trying to chat about and what we're trying to learn, so, okay, let's hover over let's do our learning outcomes first. I always forget about this one. So, our learning out for today is for you guys to define the core principles of agile development and its benefits in software projects.

Armand le Roux: define key scrum components including sprint backlog and increments and then identify and explain key roles in scrum. so we'll take a look at cool things. I'll show you different roles in scrum. We'll take a look at some scenarios and I'll have you guys match the role to the scenario and so forth. It's going to be a fun session. So here is the question that I have for you guys. Why is it often difficult to predict the future with complete accuracy? And how can we prepare for any unexpected changes?

Armand le Roux: So again, if you'd like to answer this for me, you can just go to that that nine dot menu at the bottom right, click on Q&A, and then ask a question, and you can just submit your answer through that. If you guys don't have to stress or type fast or anything, I'll give you guys a good a minute or two or three minutes maybe just to get your answer in. So, don't stress too much. Take your time and then you can give me an answer.

Armand le Roux: So, due to requirements changing, yes, that can definitely be a reason why it's difficult to predict the future. I think it's because things always change. Yes. and getting updated as often as possible would work to prepare for unexpected changes. Yes, staying informed and knowing what's going on around you is definitely going to help you be more prepared for something unexpected. it's also going to make things, less unexpected, cuz you probably expect most things because you're pretty aware of what's going on and what's happening. So, yeah. Does anyone like to add to those two answers that we've got

Armand le Roux: Okay.

00:25:00

Armand le Roux: So it is definitely difficult to predict the future with complete accuracy because life is pretty complex and pretty chaos as so there's just a lot going on and we can't take in all the factors into account. So it's really difficult for us to have a clear picture of exactly what the future holds for us. but we can always plan for unexpected changes in different ways.

Armand le Roux: like you guys mentioned, we can always be more aware of everything that's going on. and we can just be very informed, some other things we can do is we can do some scenario planning, So, we can think of different possible futures that we can lead into and then we can plan for different ones. we can also set up our systems to be a bit more flexible, right, and adaptable to change and not have a system that's rigid and can't really respond to change. and then we always just want to keep on learning, as long as we keep on learning and improving, we will be able to kind of see the patterns more clearly for next time and be better for the future. so unfortunately, it's difficult to be perfect from the get-go, right? we have the kind of trial and error. it's one thing we do good as humans is we trial and error and we get to a solution.

Armand le Roux: so yeah, unfortunately we're always going to have our mispredictions and our trouble that we run into in the future, but we can always try and plan our best and we can always make sure that the plans we do follow aren't so rigid that we can't really do anything to that type of move away from this structure strategy without breaking everything. So it's very important for us to make sure that when we plan and when we do do implement a plan that plan is at least a little bit flexible and adaptable. So my second question for you guys is what are the advantages of breaking a large task into more manageable pieces?

Armand le Roux: Faster performance and easy to track it. so yeah, I would definitely say we have improved time management. because we break things into smaller pieces it helps with our planning and scheduling to be more effective. So in turn we will have faster performance and then as well easy to track. Yes. because everything's broken up into smaller pieces. It's very easy to just tick things as they are done. and it's small things to do. So it kind of feels like you're progressing a bit better. rather than having a look at one large thing that needs to be done, we see it as 10 small things.

00:30:00

Armand le Roux: And as you tick one by one, you can actually see how you progress towards the end. I think it will help you to focus on things and make sure that every aspect is checked thoroughly. Sometimes when you look at a bigger picture, it makes you not see the little important things that could change the plan. That's a fantastic answer. Yes. so it does indeed improve our focus on and clarity on the task that we're trying to perform. and then breaking it up into smaller pieces allows us to thoroughly check everything as because we now think of everything involved instead of just the big thing. So that's a really really good answer. I like that one. changes are easy to adapt in small pieces compared to large pieces. Yes, for sure.

Armand le Roux: if something happens or needs to change with something small in the larger picture then it's much going to be much much easier to do. So some other things is we'll have a little bit easier time problem solving as well because we are focused on smaller aspects too. let's see Changes are easy to handle if separated into smaller tasks since perhaps the problem could become easier by doing one part instead of doing all of them at the same time. Yes, it can help you to avoid stress and procrastination making it less overwhelming. yes, for sure.

Armand le Roux: So that's something I will recommend for you guys too is when you do your tasks if something feels really big for you to do then always try and break it up into smaller parts and then do those parts one by one and take them off to see how you're progressing towards the end and see that the problem isn't actually as big as you think it is. usually if certain parts in coding takes us quite some time to do it's because there's something in there that we're not familiar with enough yet. So, there's something in that code that you're having a hard time with. Perhaps it's a loop structure or using strings. Maybe you aren't as comfortable with strings as you perhaps think you are. and so forth. so that can help you guys see that those things as so then when you break your task into smaller pieces, you can actually see what parts you're having difficulty with and you can go back and have a look at those things again.

Armand le Roux: so yeah it's really really beneficial to actually go through your tasks multiple times before you will be comfortable with any of those So yeah you guys touch on a lot of the things that I have here which is fantastic. so yeah let's hop into. So before I do that thank you guys all so much for answering the questions. I really appreciate the engagement. and the answering from you guys gave really good answers there. so keep it up. I am going to ask you guys for a few more answers here and there throughout the session. it shouldn't be as much to type as these. So we'll have polls at the end of the session and then throughout the session I just have some small things that I ask you guys as we go. Cool.

Armand le Roux: So that's my wonderful waterfall picture. But we are talking about agile and scrum a little bit more today. so to avoid any confusion, we'll just stay on this wonderful picture which is fine. So agile development basically started as a response to the limitations and frustrations with traditional software development methods particularly with the waterfall model. Right? So this is the model that you guys see in front of you here. And the problem with it it was rigid. It was heavily documented and it just really slow to adapt to change. So when things change it kind of messes up our whole process here because if you can take a look at this waterflow flow here we can see how things flow in one direction it's not really flowing back and that's kind of what the problem was.

Armand le Roux: So basically u in the 1980s and 1990s software development projects became increasingly more overbudget they were delivered late and they were just not meeting the user needs and of that makes sense right because if you think about it we're already delivering the software late right it's costing more than we tell people it's going to cost and now over time it took us time to develop that software so over time things change, we actually wanted different things. Now we want to change things up a little bit there. this waterfall process just doesn't allow for that. So what then happens is we're not meeting user needs, right? it was those changes that we're talking about. we can't predict the future with full certainty. And that's kind of what waterfall requires us to do. Cool.

00:35:00

Armand le Roux: So here we have our waterfall methodology and it is our traditional software development approach and it follows a linear and a sequential process like we can see on the screen here. so it basically consists of distinct phases that flow downward like a waterfall and that's where it gets its name. So you can see it kind of flows. a lot of waterfalls we know go just like down hardcore waterfall, but this one flows a little bit it's going downstairs, but nevertheless, it's still a waterfall. It's still flowing down. And we can see how it basically flows into the phases, The first thing first, we have our requirements gathering phase. So, this is where we get all the requirements. what is needed, what is this program going to be for? What should it do? And all those fun stuff.

Armand le Roux: we're basically planning it out right planning it out getting the overview of what's required for us. Then we go into our system design. This is where we're planning everything out. This is where we design the whole system and we try and create a system or design a system that meets all the requirements that We've gathered all these requirements, everything that this program should be able to do for our client, and now we're designing the system so that it can do all of those things. So, we're designing a system that we feel should be able to do all of those things requested by the user. Now, once we've done with our design aspect, we go to implementation. Right?

Armand le Roux: So now we implement the plan that was created what was designed by the team and this is basically where the development happens. So once we're done with our implementation right this is where we start testing right so then we vigorously test to see if everything is working as it should and if we're meeting all the requirements that we set out to meet right so is the system actually doing the job that it was set out or that was designed to do right so once we're done with testing and testing is perfect then we move over to deployment

Armand le Roux: ment. So deployment is when we finally then release the product to whoever wanted it, right? so that also is a process on its own where we go through a set things just to make sure that it deploys and it works and everything goes smoothly. And then finally we go down to maintenance. Right? This is where we maintain software. If there's any bugs or anything like that or any future updates or patches that's needed, this is where we would work on those. So this is not a bad system right so this is a good system to run with and you guys will see remnants of this once we go down to agile and scrum where it kind of follows a similar process but basically in smaller increments right but it's a good process to follow and this is kind of the process that you guys work with now right is you gather requirements right which is just kind

Armand le Roux: the task that we give you guys. Then you have to design the system. for now, if you're early in early stages, then you basically just write some pseudo code. That's kind of your designing pod. but later on, you do learn about all the diagrams. you learn about system design. you'll learn about making class diagrams, sequence diagrams, use case diagrams, all those sorts of things. and then this will allow you to actually start planning out designing a system that you want to build. U and then you go and implement that plan. And you'll be surprised how easy it is to start implementing code once you have this plan set out. because a lot of times what we do is especially with the tasks that we give you guys is you have the task, you read the task and you start doing the task bullet point by bullet point.

Armand le Roux: But it's a better idea to kind of gather all the requirements that you would need and try and design a system that can basically meet all of those requirements for you. Then you implement that, you code the system, you test it. What you guys would do now is you more manually test things rather than using automated tests, which is perfectly fine. and then deployment is basically when you submit it for review for us so we can check it out. so that's your basic process that you're following at the moment. it's a perfectly fine process. and it's okay for now but once systems get larger and larger this becomes more difficult to do and later on you'll see that you'll actually start changing your mind on certain things and how you want to do things and so forth. and that's why we want to be a little bit more open to change and that's where waterfall just makes things a little bit more complicated.

00:40:00

Armand le Roux: so with waterfall we have to complete each phase before moving to the next. So this leaves very little room for us to make any changes once we finished up with so when we implement waterfall it's usually for a project that has definfined requirements and has very low uncertainty. Right? So like we said the tasks that we give to you guys are set in stone. So they're not going to change while you're doing the task. so that's why you can follow a waterfall structure. Get your requirements, design your system, implement test it, and then you send it through for review. Cool. So you guys can see that you're also already implementing some of these things. and even if you haven't yet started with your course, I know some of you guys started with your boot camps three or four days ago, 3 days ago.

Armand le Roux: but you'll be using this type of system to build your So in February 2001 to be precise, there were 17 software developers that met at a ski resort in Snowbird, Utah, and they all came from different backgrounds and methodologies, but they all shared similar frustrations with basically the way we produce software.

Armand le Roux: and the problems that we've been running into. So, what they ended up doing is they created the agile manifesto, which is basically a short document that laid out four key values as well as 12 principles to guide us in software development. And these are actually quite useful. So it's basically just these principles and values kind of push us towards proucing software quickly efficiently and producing high quality software as well whilst doing it quickly and efficiently. So it is quite a process.

Armand le Roux: It is quite a thing but it somewhat works I mean running software has been rebuilt this way for quite some time now and it works. but yeah it definitely does have its setbacks as well. So it's definitely not like the end all be all, right? But we got to take a look at different types of approaches, different methodologies. Who knows, maybe you're the one who thinks of a different methodology one day that is just better, maybe you see a different pattern, you see a different picture and you take a different approach to things and it works out because that's what all these people did and it worked out for them. So you should always be open to those sorts of things too. okay.

Armand le Roux: So the agile methodology is a project management approach and it involves breaking the project into phases and it emphasizes continuous collaboration and improvement. So trying to take a project and we break it into phases. Right? So, Waterfall kind of does the same thing, but it's different with agile because we break it up in a way that we're continuously building and delivering software while still being open to additions and changes in the software. So, let's take a look at the agile manifesto. So, here is the manifesto. So these are the four core values that we've spoke about very briefly in the previous slide.

Armand le Roux: so yeah let's take a look at so it says here through this work we have come to value individuals and interactions over processes and tools right so interacting with our clients and customers over the processes and tools that we use right working software over comprehensive documentation so we rather have things working or rather have the software running as opposed to having software that's heavily documented, then we'll have good customer collaboration over contract negotiation. So having good collaboration is more important than the contract and the money involved, Because we're still trying to build high quality software while still making money on the side, but we'll chat about that in a second. And then responding to change over following a plan, right? So being always being open to change and not being stuck to a rigid plan.

00:45:00

Armand le Roux: So then once we're done with these four values, Then comes the most important sentence that is while there is value in the items on the right, we value the items on the left more. So do not think that the items on the right don't matter. It's just that we value the things on the left more. Right? So we still want good processes and tools, right?

Armand le Roux: but not at the cost of individuals and interactions. Right? We still want working software, a comprehensive documentation, but not at the cost of not having working software, right?

So, if someone's writing documentation where there's a bug to be solved nowhere, let's solve the bug or get rid of or get that feature out before we worry about documenting those things basically that way. But we still want to do those things, right? We still want to document. We still want to have good process and tools. we need to do contract negotiation. we cannot not make money, We need to make money. and then we always have a plan to follow, So, we're still following a plan, but our plan is open to changes, and that's what's important. So, I feel like the 12 principles that are involved kind of lay things out a little bit better or kind of explain these core values a little bit more.

Armand le Roux: So we'll take a look at those 12 principles and then this should kind of clarify the agile methodology a little bit more for So you can see we have the 12 principles of our agile software development. so yeah let's go through them one by one and then we'll just chat about them a little bit. So the first one we have here is that our highest priority is to satisfy the customer through the early and continuous delivery of valuable software. So it's always important for our customers to be happy. and we basically want to continuously deliver software to them. Right?

Armand le Roux: So you'll see later on when we take a look at how these different agile methodologies work, they're basically constantly producing a new version for the customer to take a look at with more added features and more basically just a next version of their application and you're constantly doing that, So you're doing these cycles and then every cycle you give them a better program to work with. then our next one here is changing So we welcome changing requirements even late in development. agile processes harness change for the customer's competitive advantage. Right?

Armand le Roux: So even if it's very late in development and we're basically almost done with the application as a whole or the project as a whole, we're still welcome change and we're still willing to make changes and adjust things. and the reason for this is like we've mentioned the customer's competitive advantage. So you guys have probably seen how crazy the software industry is and how competitive it is, So, whenever you see one piece of software make its way to an app store or so, then probably within about a week or two weeks, you'll see a bunch of copies of that software. People trying to do pretty much the same thing. and one of those apps do end up taking a little bit of the user base. and it does make things very very competitive in the software field.

Armand le Roux: That's why it's very very important that we create the best software we can and we're always willing to change for the competitive advantage right so if we can change something in the project to make it more unique and make it stand out then that's something we would actually opt to do if it's going to provide us that competitive advantage right cool then want frequent delivery right so we want to deliver working software

Armand le Roux: frequently from a couple of weeks to a couple of months with a preference on the shorter time scale. So we basically want to produce working software constantly. So like I said in these sorts of cycles that we have, we make changes, we make a better version of the software and we provide that software to our clients. And we basically want to do that constantly until we've built the larger project as a whole and we're eventually done with the project. so we always want to aim for a shorter time scale. So the shorter the cycles the better it would be because we kind of rely on these cycles to also get feedback. and that feedback is very very valuable as well because getting feedback from our client will allow us to know if we are heading in the right direction with this project and if we are meeting the requirements that they would like us to meet. Right. cool.

00:50:00

Armand le Roux: Then we want to communicate regularly. So business people and developers must work together daily throughout the project because we want to make sure that communication is strong. Want to make sure no one's misunderstanding what should happen. and everybody's on the same page. is what features are in this cycle? can we get all of that done? Is anyone stuck anywhere? Is anyone blocked? How can we prevent that from happening? How can we unblock this person? All these sorts of things are things that we need to take into account. And if there's regular communication, if there's good communication, then we can do that very very easily. So then we want to support our team members. So we want to build projects around motivated individuals. We want to give them the environment and the support that they need. And we also want to trust them to get the job done. So this is quite important is that we don't want to micromanage our developers, right?

Armand le Roux: because this is something that can happen quite often and it can really kind of hinder the process, so we just give people what they need and we trust them to get the job done because we kind of put this team together. We're trusting in this team and we're expecting them to actually do what is face tof face communication. So this is our sixth principle. the most effective and efficient method of conveying information and within a development team is a face-toface conversation. So the reason for this is it's important to see someone's face when you're talking to them.

Armand le Roux: because it's difficult for someone to lie or hide emotion or things like that. so not necessarily that people will lie, but a lot of times people are stressed and they're hiding it because they're stressed because they don't want to look bad and things like that and you can see that in someone's so it's really good to have face to face communication. It's not necessarily have to be in person. it can be a video call as well but it's important that you can see the person's face. you can see their emotions.

Armand le Roux: you can see how they react to certain things. because that's just going to help things be more transparent. and if there's good communication like we spoke about a little bit earlier, then we can get the job done much better, so it's always good to be able to have face to face conversations, be able to talk to each other one-on-one, be transparent, and kind of be able to talk about what needs to get measure work progress. So how do we do that? our primary measurement of progress is going to be working software, So as long as we constantly produce working software, we can see that as progress because each piece of working software that we produce is more features added and closer to the end of our project that we're trying to produce. Cool.

Armand le Roux: so then the agile processes promote sustainable development. So the sponsors, developers and users should be able to maintain a constant pace indefinitely. Right? So basically the way we set up our process it should be done in a way where we can constantly keep that same pace of development. Right? So we constantly want to produce the same amount of working software essentially through each cycle. and we want those cycles to be the same amount of length, right? We don't want to have a vary in lengths based on how we're doing things. because that kind of makes things less consistent and that makes things unstable, So, we want more consistency. If we have consistency, we have a little bit more stability within the team.

Armand le Roux: So that's why we want things and that's why we want to set up the process in a way where our teams can basically keep a constant pace throughout the development process. And the main important part about this is that our team doesn't burn out, so whenever this we don't want things like you guys have probably heard of crunch before and things like that. We don't want things like that to necessarily happen. we want our processes to be set up in a way that we never want to have employees burn out or feel like they're ever overworked. Cool.

00:55:00

Armand le Roux: Then we want continuous attention to technical excellence and good design enhances our agility. That's going to be principle nine. and it's basically we want clean code. We want a solid architecture and that matters. We don't really want to cut any corners. so we still want to produce proper code even though we necessarily don't want to want a hard focus on documentation and things like that. We still want things to be clear and we want a good structure and a nice architecture for our software. Then simp the art of maximizing the amount of work not done is essential.

Armand le Roux: So what this is is we don't want to build any unnecessary features and we want to keep our software lean. So we only add the features into our software that we need. We don't add anything that's not necessary. and this is a good way of going about things because a lot of times these features and extra things that we like to add in, they just kind of form extra work and then we kind of increase our chances of feeling something like scope creep where we just have too much to do and the project just gets too large and we don't want to do it anymore. Cool. Then we have the best architectures, requirements and design emerge from self-organizing teams, So we should let our teams make their own decisions because they know what is best to do.

Armand le Roux: So again like I said providing our teams with the environment and the support they need and this kind of attaches on to that right we trusted these people that's why we hired them right we made them a part of the team because we basically ed in their expertise and their knowledge so now we kind of have to trust in that and have them make their own decisions and have them decide what they want to do so right we want to avoid micro managing the team. We just want to support them and provide them with the environment that they need and then at regular intervals the team reflects on how to become more efficient then tunes and adjusts its behavior accordingly. So because we're working in these we can call them cycles right it gives us a learning opportunity after each cycle.

Armand le Roux: So we go through a process, we develop some features, we produce that software to the user or to the customer and then we reflect on how well Is the process working efficiently? can we improve in some ways? Can we adapt? how can we improve things, How can we make things better? how can we work better as a team? How can we have more efficient communication throughout the process? All these sorts of things can be under review and can be chatted about to be improved. And you'll be surprised how much things can improve as each cycle goes because as you work on different sets of features, right, you run into different blockers and different scenarios.

Armand le Roux: And this is kind of then where you can see some different perspectives and then we can work and adapt our plan a little bit to adjust and make sure that it actually takes these sorts of things into account for our next cycle that we didn't really take into account for our previous cycle. So I see it's 5 minutes too. So what I think we'll do is we'll take about a 5 minute break. So we'll take a quick 5 minute break where you guys can get up, you can stretch your legs a little bit, maybe go grab something to drink or snack on.

Armand le Roux: and then once we come back we'll chat about some agile frameworks and then we'll take a little bit more of a deeper dive into scrum itself and then I'm going to show you guys some roles in scrum and then we'll match some scenarios with the roles as so a little bit more fun in this session but yeah let's do a quick 5 minute break just so you guys can stretch your legs a little bit and refresh yourself. so let me see if there's any tools here. No, there's no such thing or it's like a timer or anything like that. So what I'll do Yeah, that's fine.

01:00:00

Armand le Roux: For you guys that are here, we are going on a 5m minute break. So just know that we'll be back in 5 minutes. So that's going to be like let's say 2 minutes past. We'll come back. So it's about 6 minutes and then we'll carry on with the rest of the session. Cool. Sika is in about 5

01:05:00

Armand le Roux: And we are back. So, I have three agile frameworks here for you guys. and two of them we'll just touch on a little bit more, just like a brief chat over. And then c we'll chat about Scrum. We'll go through the process of Scrum, how it works, what it is. I'll have you guys do a little bit of a standup for me. We'll take a look at the roles in Scrum and then we'll also match some scenarios to the roles. So, it's not that much to do. It feels like a lot, but it's not that much. and you guys will probably enjoy this. Let's take a look at Cananban. So, there's three agile frameworks that I have for you guys. Canban, XP, that stands for extreme programming, and then Scrum, of course.

Armand le Roux: projects that are creating cananban are basically overseen by a canban board. So the cananban board is divided into multiple columns to basically depict the process flow for the development of our software. So this process helps in enhancing the visibility among the team members because each team member can see what is happening and what isn't happening so they can prepare themselves for anything that they have upcoming that they're going to need to do for the final delivery of the project. So there's a lot of everybody involved knows what's going on with the project at any given time because they can just take a look at the canban board, so what's nice about this is it requires thorough transparency as well as good interaction between our team members for the proper development of a product.

Armand le Roux: and the thing with this is, right, when people are on the Canman board, when you're taking a look at what's being worked on and what's not, you have to be transparent about what you're working on, So, you can't start working on a feature and then not show that on the board and then suddenly someone else picks up that feature and you're like, " damn it. I've been working on that now and now I have to kind of awkwardly say, listen, I've already started doing that." and so forth. So you need good communication, good transparency. everybody needs to be willing to show what they're doing and what they're busy with at any given time so that everybody can be aware of what should happen and what needs to happen, right? yeah, let me quickly take a look here.

Armand le Roux: Let's do ops. Basically like this, right? So, we can see the backlog. We can see what someone's doing. We can see what's under review. And then we can see what's done, So, basically a board like this. And this is basically the project. And it shows everybody who's busy with what and what's happening. Cool. Then we get extreme programming as so extreme programming is a agile methodology but it focuses on delivering high quality software through frequent and continuous feedback, collaboration and adaptation.

01:10:00

Armand le Roux: So this requires a close working relationship between the development team, the customer and the stakeholders of course. with any agile system and it emphasizes a rapid iterative development and deployment process. So what happens is our project starts with user stories which are basically short descriptions of different scenarios that the customer and users would like the system to support. and then those stories are then written onto separate cards and then they can be easily grouped together for implementation. Then we'll decide which stories we'd like to group together and implement the features to cover those stories. so yeah, it's more about having these stories and then we're collecting them together and then we're solving them. But with extreme programming it's more

Armand le Roux: about taking good practices to the extreme. So if we think about it in pair programming, we see things such in extreme programming we see things such as pair programming as a good idea, programming in pairs to review each other's work and things like that. because that's good, let's do it all the time rather than just sometimes, since it's good to test a program early on, let's just test everything before we even start writing code, right? So it basically takes whatever good practices we have and it kind of flips it around and takes it to the extreme to see how we can push the boundaries of that good practice.

Armand le Roux: So now I can take a deeper look into Scrum. So you've guessed it, Scrum is in fact a agile development framework. and it's commonly used in software development as well as other industries. So scrum basically prescribes for teams to break work into goals to complete it within a timeboxed iteration right and we call these time boxed iteration sprints.

Armand le Roux: So the whole idea of scrum is basically to have these cycles that we call sprints and then we have goals that we're trying to reach for each sprint. and that's kind of the main process involved with scrum that if you break it down really into its simplest form that's what it's about. And then we can see the process in a nice neat little image for us. you can see the entire process as it unfolds and how it actually works very well. so let's go through these and see what it's about.

Armand le Roux: So at first a clear vision of the project is going to be provided to the entire team with a set of features that we would like and that those features would basically be in an order of importance right so we have a bunch of features that we want to implement and then each feature would have what's the word I'm looking here like a priority right

Armand le Roux: So, Is it low priority? and then all of those features kind of get thrown together and all of those features that we want to add into the program ultimately form what's called a product backlog. So, it's a backlog of everything we want in the product. Cool. then from there, so here we can see this is our product backlog.

Armand le Roux: This is where things start, So then from there we go into our sprint planning meeting, So this is where we need to now plan what our sprint is going to look like that's ahead, right? So how we plan the sprint is we go into our product backlog and we take a look at So then we take a look at the features with the highest priority because obviously they're the ones that need to go into the sprint because they need to get done. we need to finish them off, right? So, what we then do is we take features from the product backlog and we determine which of those features we can complete within the sprint time frame. So, remember a sprint is only going to be a set time. and it's usually between 1 to 4 weeks. We can see that here. So, our sprint is usually between 1 to 4 weeks and we want to keep that time frame consistent, right?

01:15:00

Armand le Roux: So that means if we have two week sprints, we always have two week sprints. If we have four week sprints, we'll always have four week sprints and so forth. We're not going to have a two week sprint this time and then a 3 week sprint and then we're only going to do a oneweek sprint now. We don't want to do that. We want to have consistency, Because remember part of the agile manifesto is having that consistency and preventing our team from burning out. So we want to have a process where our developers can continuously work work without feeling like they're basically killing themselves, So it's important for them to be able to stay fresh, feel like they get enough breaks, feel like they get enough time for themselves, as well as feel like they're being as productive as they possibly can in the workforce.

Armand le Roux: So what then happens right is once all the features are decided on. So what happens with the product backlog right is even though we have that are high priority features that are low priority. What can happen is a lot of times the low priority things wouldn't take that much time. So some of the low priority things can get thrown into a sprint along with higher priority things. just so that we can make sure that we're still clearing those things from the backlog as well. but the main important thing is once we've set out some features, once we've basically confirmed the features that we want to add into this current sprint and we've confirmed with the team has to commit to that sprint. They has to say yes, we will be able to do this in this time.

Armand le Roux: and that's very important because once the team commits to it, then they feel like they will be able to do it and then they kind of have to do it, and if you're not able to meet all of those requirements in that time, then that's going to be up for review, But we'll chat about that in a second. But our team has to commit to the sprint and they have to confirm that they will be able to complete this set of features in this set of So when we're busy with a sprint, right? So now we've basically gone we have our product backlog. We planned what we're going to add into the sprint. We spoke to the team.

Armand le Roux: The team said yes to this backlog and now we can hop into our sprint. So now the sprint's going, right? So when we're busy with the sprint, the team should not be interrupted, right? They should only be allowed to focus on reaching the goals that were set out for the sprint. No new work, no other things come into play that needs to be done. We're busy with The sprint's what's happening now and this needs to get done. So this is kind of where the team needs to be left to do their thing. and another important thing is the sprint backlog should never change cuz remember we spoke to the team and the team committed to a set of features within a certain amount of time, and if we start chopping and changing, we're now messing around with their planning that they thought, we'll be able to do this in this time.

Armand le Roux: But now things change and the team usually knows how long certain things would take to accomplish not the product owner and this kind of creates a situation where we can't just have anyone add things to the backlog. but it's just a small extra feature that we need now. if it was that important it would have been a part of the product backlog already and it would have been a part of this sprint. But if anything is that important then what needs to happen is it needs to go into the product backlog and it needs to be a high priority for the next sprint. we can't just add things to a sprint and it's really important that we do follow these rules because if we don't follow them then this is where we can lead to things such as burnout having disgruntled employees where things change often and stuff like that.

01:20:00

Armand le Roux: We're open to change, but once we've committed to a certain set set of features, we want to finish those before we go and do extra changes and new things, because it's so important to follow a plan, we still value the items on the right, Even though the items on the left are more important, we still value the things on the right. So, we still want to work with somewhat of a plan. We don't just want to chop and change everything at a during our sprint, so we have our sprint that's going on here. So during this time, we have what's called, you can see these daily scrums. So every 24 hours, we have what's called a daily scrum. So this is where everybody gets together and then the scrum master goes and he says crouch, bind, set for everybody and then they just hit and they start scrumming.

Armand le Roux: so if rugby, you would know that. But, funnily enough, the word scrum here does come from the term in rugby, which is where they scrum. It's basically a set piece in rugby where you contest for the ball. but yeah, that's kind of where the name comes from if you guys did wonder. but yeah, the daily scrum is basically a 15minute meeting every day. and in this meeting the whole team comes together along with the scrum master and they basically chat about what each one is doing. So each team member gives what's called a standup and that's what I'm going to have you guys do a little bit later for me in this session.

Armand le Roux: so here's where each team member will share what they've done the previous day, what they will be doing today, and if anything is blocking them from achieving their tasks or their goals. and this is important because that's where our scrum master will see later on where our scrum master comes into play as they will actually assist in getting rid of things like that. So if something's blocking you, we need to get rid of those blockers, And that's what makes these daily scrum meetings so important is because you can hear from the team how they are progressing with the features in the scrum because remember you've committed to these things to be completed within this sprint.

Armand le Roux: so it's important for us to basically meet daily and make sure that we are making progress on this sprint backlog to ensure that we are completing the backlog by the end of the sprint. We can't have anything left by the end of the sprint. and if there's anything left that means there was probably some sort of blocker that was never addressed, So we always need to make sure that there's transparency right and honesty within the team. Cool.

Armand le Roux: So then after a sprint the team will then demonstrate the work that they have done to the stakeholders to get feedback right and then after that they'll also be reviewing their process right so we can see here they finish the work we do a sprint review and we also look back right we also take a look and see what could we have done better right could we have made changes here the way that we addressed this blocker probably wasn't the best way Maybe we should address it this way next time. maybe we overcommitted to features and we weren't able to get all features done. So for the next sprint, we'll basically commit We'll commit to less features to make sure that we rather get everything done instead of having to please explain to stakeholders why we weren't able to get to all the features and so forth.

Armand le Roux: So this is quite an important part of the process is making sure that we reflect and we improve upon the process with every iteration that we go through. So now you guys can kind of see how this is kind of just like a smaller waterfall process, right? we get our requirements and then inside the scrum we basically plan out our requirements or we design our system we create the features we test to see if it's working and then we finish off and we add it right so it's basically like these mini waterfall sprints that we have it's not a completely accurate description but it's basically the process that we're following we're getting our features we're developing our features we're pushing them out and then we're grabbing some more features

Armand le Roux: is from the backlog. So let's chat about some roles that are involved here. That's a scenario, not a role. Okay, let's chat about the roles involved. So up top here we can see here is our scrum master. So the scrum master is basically the keeper of the process, So they're the advocate for the team and they're the protector of the team. so their job here is to remove obstacles facilitate team communication mediate discussions within the team and negotiate with those external to the team. But above all they basically exist to service the team in any way.

01:25:00

Armand le Roux: So the scrum master's goal is to make sure that the sprint completes and all the features are in the sprint. so that's going to be the main goal for the scrum master is they're making sure that everything gets done. and they do that by doing things such as removing obstacles facilitating communication between the team members negotiating with external members to ensure that their team members are kept safe or not too much is expected of the team to be able to finish in a sprint and things like that. Can you please explain again what a sprint is?

Armand le Roux: So a sprint is essentially just a one to four week interval that we have where we're completing the sprint backlog that So the product backlog has a bunch of features for the entire product that we're trying to develop and we're developing the product in cycles called sprints. So then for each cycle we'll grab some features from the backlog add it to the sprint backlog and then we work on those features So in this cycle we'll work on those So sprint is essentially just a cycle in which we work on features from the product backlog.

Armand le Roux: So, just to hop back into our scrum master here. So, our scrum master basically makes sure that we can get through one of these cycles that we're referring to. so yeah like I said if there's too many features expected from the team then they kind of protect the team from the product owner and say listen we can't do all of this features and stuff like that. if perhaps some features are taking long or the team is slow moving slowly they make sure to remove any obstacles that's basically slowing down the process or blocking our team from achieving their goals that they need to achieve.

Armand le Roux: So that's the job of our scrum master. Then we have our product owner. So the product owner represents the voice of the customer and they have the authority to make decisions about the product. So this person basically owns the product backlog and they're responsible for communicating the vision to the team. So what to be needs to be communicated through the product and they also define and prioritize the backlog items. Right? So the priority level of each item in the backlog gets determined by the product owner. and then they also define what is in the backlog, So they know what features are needed. So they have the backlog ready. They have the priorities of each item within the backlog and then that's where the team and the product owner kind of comes together.

Armand le Roux: we have a sprint planning meeting and we come together to decide which of those features we can take from the backlog and work on and complete within a sprint. So the product owner works with the team on a daily basis to answer questions and provide any product guidance. so the product owner is there to make sure that the team is clear on the vision and they know what is expected of them. It's not like you don't want to be working on anything and then you're not really sure what's the end goal of it, so it always helps to when you're working on a feature to know what's the end goal of that feature as it might change the way you think about how you're trying to develop this certain aspect of the project.

Armand le Roux: And then finally we have the team right. So the team consists out of seven plus or minus two people who are jointly responsible for the delivery of the product. So they own the estimates, they make task commitments, they report daily status to each other in a daily scrum. they are self-organizing. So they structure themselves. No one from the outside intervenes and tells them how they work together and how they should approach the sprint. The team comes together and they decide who works on what feature and they self-organize themselves to make sure that the sprint can be completed. so yeah, the team basically owns how it chooses to build the product features.

01:30:00

Armand le Roux: So the team where the product owner basically owns what the product owner gets to decide what the program should do or what the project project should do and it gets to decide how this project does those things that it's set out to do. So, I want to present you guys with four scenarios and then I'll give you guys about 2 minutes for each scenario. where you guys can decide which role fits the scenario the best. So, it's only going to be between three roles. So, we spoke about the scrum master, the product owner, and the team. Right? So, I'll take a look at four roles and then you can tell me is that a role for a scenario. Sorry.

Armand le Roux: then you can tell me is that scenario for someone who's a scrum master, a team member or a product owner. So here's the first one. So during the daily standup, a team member mentions that they're stuck on integrating a third party payment gateway. You listen carefully, offer suggestions, and follow up after the meeting to remove roadblocks. You also make sure the team has everything they need to stay on track with the sprint goals. Who am I? No, I'm just kidding. But what role fits this scenario the best for you guys. Take your time. Think about it carefully and then pop me an answer.

Armand le Roux: the answers I'm seeing is looking good.

Armand le Roux: So, I see the overall majority here says that it's going to be a scrum master. And I would say that you guys are in fact correct. Yes, it's a scrum master.

Armand le Roux: So the scrum master some of the parts here that gives it away is you're listening to a team member you offer suggestions and you follow up after the meeting to remove roadblocks right so that's kind of the job of a scrum master is to remove any blockers that you may experience and also making sure that the team has everything that they need to stay on track with their sprint goals cuz remember as a scrum master your goal is to ensure that the sprint actually is successful.

Armand le Roux: that everything in the sprint is cool. Let's hop over to our next scenario. So, you lead the efforts to break down the product vision into smaller actionable tasks. You organize a backlog grooming session to estimate the effort needed for each item and ensure they are ready for the next sprint. You also collaborate with stakeholders to clarify requirements. So, what role do you guys think this is?

01:35:00

Armand le Roux: And I see everybody is saying product owner which is also correct. You guys are doing very very well with this one. So yeah, you lead the efforts to break down the product into vision into smaller actionable tasks. you organize a backlog grooming session to estimate the efforts needed for each item and ensure they are ready for the next sprint. to ensure that the features are ready and you check for how much effort is needed for each item. and you also collaborate with stakeholders to clarify requirements. So yeah, that basically describes a product owner for us. So our next scenario here is the team struggles with estimating user stories. You introduce them to techniques like planning poker and help them practice in a collaborative environment. You also encourage open discussion to build confidence in their decision making.

Armand le Roux: So, what rule do you guys think fits with this one?

Armand le Roux: So I feel like it's scrum master was the right one. So only one person says scrum master this time. So this is in fact also a scrum master. the reason for this is the team is struggling with estimating user stories. So you introduce them to different techniques. You're trying to help them get rid of a blocker. You're helping the team get rid of a blocker. and you're also basically encouraging discussion between them and so forth. So those are the sorts of things that a scrum master does do.

Armand le Roux: And then your final one here is you create test cases for the new shopping cart feature and execute them to ensure it functions as expected. You identify a few bugs and collaborate with the developers to resolve them. Once fixed, you verify the changes and give your approval.

01:40:00

Armand le Roux: Okay, so this last one is a team member. So the first part here kind of gives it away. You create test cases for the new shopping cart feature. So you're actively busy with development of the product. so that's kind of where this leads to being a team member. but it's okay if you didn't get these scenarios correct. if you only got some of them correct, it's fine.

Armand le Roux: And I don't expect you guys to exactly know what a team member, a scrum master, and a project product owner does right after you've heard it just once, for the first time. but yeah, it's good to keep in mind what their main thing is, So here we can see we're touching on the product backlog. So that refers to the product owner. here the team is being helped. So you're helping out the team. removing struggles from them. That's what a scrum master is doing. And here creating test cases for a feature and executing them to it functions as expected. You're identifying bugs, those sorts of things. That's what you do as a team member. You're actually actively busy with thank you guys so much for participating in those scenarios. There's one more thing that I'd like you guys to type out for me and then I promise you won't have to type anymore unless you have a question of your own.

Armand le Roux: but I want you guys to do a daily standup for me. So, this doesn't have to be anything personal. This can even be just about the boot camp itself. so I'd like to know from you guys, what have you accomplished yesterday? What will you do today? And what is blocking you?

Armand le Roux: I see you guys are a little shy with the stand up. We just Okay. what's

01:45:00

Armand le Roux: So, while some of you guys are still typing out your standups, we'll go ahead and we'll chat a little bit about the disadvantages of agile as this is the last part of our session. so agile seems pretty good, right? It seems pretty nice, but it does still come with its disadvantages, So, one of them here is the lack of predictability. So, the issues here if you have cost, time, and resources that can be hard to estimate up front as we're always open to change. So, it means that features can always become more. So, the pro product the project can take longer to complete.

Armand le Roux: it might require more resources than we thought at first and so forth. so it isn't really easy to predict and it's because agile is so flexible, it makes it difficult because we don't really know how long it will take us to get to our final product. so then scope creep as well. You guys heard me talk about that. So features can keep on expanding endlessly because agile encourages incorporating feedback and changes and changing requirements based on user feedback if we don't do that without discipline it can lead to uncontrollable growth in the scope and then we get scope creep right where we have too much things to do and we end up not wanting to finish the project because it's just too much to get done. then it also requires a high team commitment.

Armand le Roux: So agile needs a team that's engaged, skilled and available as well. So we have daily standups, we have regular reviews, we have quick decisions and that depend on active collaboration and full-time dedication. And this isn't always possible with team members. especially we all have families, we have personal lives and so forth. So it does sometimes make it difficult to commit to an agile methodology that basically needs constant involvement. they aren't ideal for large complex projects that have fixed requirements. So it isn't really the best fit for everything and it must be defined up front. So agile basically thrives on uncertainty and it's a problem if stakeholders demand fixed deliverables from the get-go because then you can kind of opt for more of a waterfall approach where everything is already as it should be,

Armand le Roux: Yeah, documentation can be neglected. So, agile kind of allows us to sacrifice documentation in favor of having software that works. But we should be careful not to have too little documentation as it can make maintenance a little bit difficult later in the product lifespan.

Armand le Roux: and you'll see as well, if you go back to code that you've written, let's say you're now a few, let's say you've coded for about two years, and you go back to code you've written about a year and a half ago and you haven't commented or documented that code, it's going to be very difficult to necessarily understand what you were thinking and the mindset that you were in. And trying to understand what the code is doing can become very very difficult. so documentation is really really important especially when it comes to maintenance and agile kind of does allow for us to forget about and neglect documentation a little bit more. and that can bite us a little later on when you need to start maintaining software. and then finally client involvement is constantly needed. So some clients don't always have the time or interest to be deeply involved throughout the whole process of the development.

Armand le Roux: and agile works best if clients can give regular feedback. because without it, decisions may stall and the product may drift off course because we aren't really sure exactly what the client wants. so we of We rely on constant client inment. And that is the session guys. So I have two poll questions for you guys. let me just get to them. Here's the first one. What is the main goal of agile development? So to get to the polls, you can go to that nine dot menu again and instead of Q&A, you can just click on polls and then you can vote on a poll for me there.

01:50:00

Armand le Roux: So, let's have a look here. What is the main goal of agile development? To eliminate the need for documentation. Nope. to deliver value to the customer quickly and iteratively. That sounds pretty good. to ensure all planning is completed before starting work. No, that sounds like waterfall. And then to avoid collaboration. So no, it's going to be to deliver value to the customer quickly and iteratively I see you guys voted Fantastic. So here's your second and last poll for the session. what is the role of a scrum master?

Armand le Roux: Let's take a look here. Prioritize the product backlog. nope. That sounds like a product manager. Manage the team and assigned tasks. No, not really. Facilitate the scrum process and remove obstacles. Yes, that sounds like a scrum master. approve the final product that's a product owner rather so cool I think the right answer here which you guys chose is facilitate the scrum process and remove obstacles fantastic guys thank you so much for joining me for this session I hope you guys had fun and you learned something new and you find the session at least interesting

Armand le Roux: so yeah, this is A part of the session now. So if you guys do have any questions that you're more than welcome to ask me now. and then next week we're going to be taking a look at logging is quite cool. So when you have applications, when you build programs, you'll see that it's a good idea kind of as programs run to log what's happening inside of the program so we can see what's happening. And so when we run into any bugs or when unexpected behaviors occur, we can just take a look at our logs to see what's going on, what happened, and this can help us debug our problems. But yeah, that's something we'll touch on next week and we'll learn more about next week. so yeah, even if you guys do need to hop off, I'd like to thank you guys for joining me for this session. It was an awesome session. Thank you guys for being so engaging and being so cool. yeah, I'd like to remind you guys to take regular breaks. So that's a 5 minute break every hour or at least every two hours.

01:55:00

Armand le Roux: Make sure you step away from your computer computer, take a nice break. stretch your legs and so also keep some liquids nearby. So, some water preferably, but you can also drink some coffee and tea. just make sure you do get some liquids in. Then, also don't strain your eyes. Look away from the screen every once in a while. look out the window or so into the distance. And then last but not least, remember to have fun. It's important for you guys to have fun. It will get you through the tough times, the easy times, and everything in between. Remember to use all your resources to your advantage that you have. and let's get you through this boot camp and make you the best developer you can be. Awesome guys. Thank you so much and enjoy the rest of your evening. If you do have a question, I'll still stick around for you guys for a few more minutes and then I'll hop off. Cool. But you guys are more than welcome to hop off if you need to go.

Armand le Roux: I know the Google meetings are recorded. but I wonder where I can find these sessions so I can rewatch. so this is something we're trying to work on for you guys. from what I'm aware, I don't think there is a rewatch system. I would have to go clarify that, which I will do. I'll ask tomorrow about that and then I'll just tell you guys in the next session. Awesome.

Armand le Roux: So I don't see any more questions coming in which leads me to believe it's safe to end this session. So again this last thank you to you guys for joining. I hope you guys had a wonderful time and then I'll see you guys next week for another session where we'll chat about logging. Awesome. Bye everyone.

Meeting ended after 01:58:30 👋

This editable transcript was computer generated and might contain errors. People can also change the text after it was created.