Part1

Intro

Hi, this is Nancy me to Carnegie Mellon university. I’m delighted to join you. I’ll be doing some sessions on requirements and also on threat modeling. And then on malware analysis as it involves requirements. So you can see, I have a lot of interest in requirements and specifically in security requirements. I know that Jeff has already introduced the topic so let me tell you a little bit about me. I’m a fellow at the software engineering Institute. I spent about half of my career with IBM federal systems, working on large software systems and the other half of my career at Carnegie Mellon software engineering institute. I’ve been affiliated with the master of software engineering program. Since quite a long time for a while I was on the Executive Committee for the program some of the Faculty reported to me and then later on, I became an affiliated faculty Member myself. So I’m adjunct Professor software engineering at I asr, which of course is the tool that you're working with so it's great to be here with you.

1p

So on this slide you've got my user ID which I think is also on the canvas site, and you can feel free to contact me anytime. Before, during or after any of these sessions I’m glad to hear from you, if you later on want my opinion on something or my input on something after you're done with the course. That would be great I’m always delighted to hear from practitioners, who are actually trying to do some of these things in the field, and I know that applies to you.

2p

So here we're going to be talking about three small topics so we've got three segments in this very first one we're going to talk about requirements engineering in general. And SQUARE in particular, I know that Jeff has already started on some of these topics, so this will be, in part or refreshing important and look forward. And the second topic is specifically focusing on the steps of SQUARE. So that if you want to you could actually go out and try to use the process or even pick a few steps that you think are important and try to execute those. There's the lightweight version of SQUARE that is fewer steps and doesn't take as long, which is a good thing and it's a good thing that is if you're the right user and we'll talk about that as well, and then there's a version of SQUARE for acquisition. And we know that's really important, because these days there's a lot of software that's being acquired from a supply chain. So you may not be doing all of the development yourself, you may be relying on other vendors, in which case you want to know that that software supports your security requirements. And, at the end of all that will summarize, but for now we're going to be talking specifically about requirements engineering and a little bit about SQUARE.

3p - Requirement Engineering

So I’ll bet quite a few of you have experienced with requirements engineering and that may be a very good thing, because it will help you in thinking about security requirements.

4p

Are there any issues with requirements engineering you bet they are, and this is something that you've already discussed I’m sure. Why is security requirements engineering important. If you haven't already figured that out we're going to talk about that some more. And then I know Jeff introduced SQUARE briefly in one of the earlier sessions so we'll go back over that.

5p - Requirements Engineering Issues

So one of the issues associated with requirements engineering. There is a lot of data that shows that areas in requirements any kind of requirements left your security requirements can cost up to 200 times more to fix if they are not caught early on. In other words, if they're not caught until the system is operational. It can be pretty embarrassing if you've got a customer at the other end and you have to push fixes out to them after a while they start to wonder how good your products really are. And it's, as I said, really expensive. **Typically rework cost more than half of project effort. More than half of the defects are introduced during requirements engineering. And those kinds of areas costs a lot of money.** We’ve got a live example here.

6p - Requirements Engineering Issues: Example

This is based on real data it was collected by fortify working with a large software vendor. So the numbers are not real. But if you were to convert them to percentages, the percentages are real. So here if you find an error during the requirements activity. Here’s. An estimate what the cost might be to fix one buck and again. This isn't necessarily the correct dollar figure it's just representative of a percentage. If you find that same error in design, it becomes more expensive. And as you go further in your development process doesn't matter what process you're using you can change the names and the result will be the same. The longer it takes to find that problem, the more it will cost to fix and that assumes that you can actually fix it in a suitable way, which in some cases may not be that easy. So in this example you had 200 critical bugs you identified 50 of them during testing and the hundred and 150 during maintenance or operations, so it becomes really expensive. On the other hand, if you are able to identify some of those earlier let's suppose you identified 150 of them during coding. And 50 of them during testing so it's the same 200 bugs. But Look how much less expensive that would be just by being able to identify those errors earlier. And fix them sooner, not to mention how good you look, because you won't be finding all those bugs and when the system is operational you'll be finding them while it's still in House. So that would be a very good thing, so in this example. By fixing problems earlier you save almost $2 million. Again, the dollar amounts as numbers don't really matter but think about the percentages and keep in mind that this was a real study and a very large one. It’s not just made up data.

7p - Requirements Problems

So what are the kinds of problems might occur. Well, you might not have all of the stakeholders that you need to have on board as an example on one project that I was on. We had included the customer who's paying for the software. But we didn't pay too much attention to the end users who were out in the field. Those end users showed up for reviews, but that was about it, they didn't understand anything about requirements or design for that matter. They didn't really understand what was missing until they looked at the user interface, and they realize that some of the data that they wanted to see was not there. Well, not only was it not there, it wasn't even being calculated so big goof we had to go back take a look at how to calculate the data include the hour and then get it onto the display. And you can imagine that took a bit of effort to get that done so not good. It’s even worse if you were to get it into the field and then find out that it's not usable because guess what if the end users don't like the product they won't use it, and it will have been wasted effort, nobody wants that. Sometimes the analysis is in performed so that's can be an in house problem and the actual specification can be pretty haphazard we've worked with client projects and asked for their requirements specs and in one case they kept saying oh 'yes we'll get that to you.' and finally toward the end they said well our requirements documents are so that they're not even useful or the only thing we can really count on is the code. You know, want to be in that situation.

8p - Effects of Requirements Problems

So what's the effect of all this. Well, none of them are very good. You could find if you have to go back and do rework that your schedule is going to be exceeded. Your budget might be exceeded. Alternatively, if you have a fixed budget and a hard deadline, you might have to reduce the capabilities that the system provides. Worse, yet if you rush you could end up with poor quality. As I mentioned, you might have products that aren't being used because the customers don't like them that wasn't what they wanted, which you would have found out if you have actually asked. Or, in the worst case, maybe. The way you've architected the system, you can actually easily fix the problem, in which case the project could end up being cancelled. I’ve been on projects that were cancelled and if you've been in the field for a while I’ll bet that you have also seen projects that were cancelled and, in some cases they're canceled because they can't meet their requirements in a reasonable amount of time. Or with a reasonable amount of budget. Now this doesn't just apply to security requirements. It applies to any requirements.

9p - Security Requirements

So let's add security into the mics. Typically they're looking at a particular application versus a family of applications where you might get some cost savings by sharing things course a family. Often they're ignored most users are interested in feature. Security is way down on their list of what they want to see in the system, unless they're in some kind of a critical infrastructure situation or in defense and obviously this summer we've seen plenty of cases where critical infrastructure has been impacted by security requirements from. It can be very expensive if you don't get to those issues until later on. And in addition to being expensive in terms of fixing the problems it's expensive in terms of embarrassment and loss of customer confidence and nobody wants that. **So we want security issues to be addressed early.** At the same time as any other kind of requirements.

10p - Security Requirements Methods

There are a lot of methods. Jeff already briefly introduce some of these **SQUAREs** a method for which I’m the principal investigator it's been used obviously here at Carnegie Mellon it's been used in many research settings and to some extent it's been used in industry. The other method that is closely related to SQUARE is a **SREP**. That was developed in Europe it's largely based on SQUARE, but it also incorporates elements of the common criteria, which is a very good thing, and they've had extensive funding to work on that also quite a bit of education, just as I’ve had with SQUARE. Some other popular methods that are out there, one is class which has been around for quite a long time. That’s more of a formal method. Also in the formal methods Category I would say **TROPOS**, which is a lifecycle approach. And, along with it is **Secure TROPOS**. It incorporates its own specifications elements. And then there are others like implementation of the common criteria. Number of others, some of which were worked on within various government agencies and some work done within private companies. I can give you references for all of these if you want to explore any of these further. But in this course, will be I’ll be talking about SQUARE. There are other methods that are more closely involved with threat modeling and I know that's going to be a heavy focus of yours, in the course. And again when we get to threat modeling there are quite a few additional methods that will touch on briefly as well.

11p - SQUARE

So from here we're going to talk about **SQUARE** as Jeff described earlier. It's a nine step process. We additionally have a shorter process called **SQUARE-Light**. that's five steps and takes quite a bit less time, however, works best when you already have quite a bit of process in place you don't want to decide that you're going to do something and pick this up as a standalone. A couple of years ago I worked on SQUARE for privacy (**P-SQUARE**), with a graduate student who is interested in tools and he developed a very nice tool. The only drawback of it was that you kind of had to know what privacy standard, you were following and incorporate that into the tool. Once you had done that, then you could get a standardized set of requirements, out of it and then tailor them. So it was really nice effort and a nice tool, but it didn't depend on having standards. Re-constituted them incorporated into the tool. So obviously it's a bit out of date by now. **A-SQUARE** or SQUARE for acquisition I think can be very useful, because we all know that we're using a lot of products from third parties in our systems. And A-SQUARE gives you ways of incorporating security requirements when considering those products and addition it can give guidance to contracts that you're putting in place with vendors. So SQUAREs, not a new overall requirements engineering process. Pieces of could be incorporated into an existing process, and if you already have a method for developing security requirements, you could take a look at SQUARE and see if there's something you want to add.

12p - Reading Assignment: Khan/Zulkernine paper - On Selecting Appropriate Development Processes …

With that there's a reading that I’d like you to take a look at it's a survey paper that looks at a different number of different security requirements processes. And also comes up with features that they would like those processes to satisfy and you can see how SQUARE measures up against some of the other. Security requirements processes that they looked at, in addition, there are other references, there are reports I have a number. On my own files and so, if there's some area that you would like to explore further contact me, and I’ll send you some reference material. In addition to your reading assignment of course. So with that that's the end of this segment. In the next segment, we'll move on to talk about the nine SQUAREs steps, thank you.

Part2

Intro

Hi this is Nancy meet again. We’re now going to talk about the nine steps of SQUARE. It sounds like a lot, and in some regards that this lot. will be doing an overview here, if you think there are specific steps that you want to explore or all of them, I can give you a lot more reference material to go through how you would go about this in detail, we think about it in terms of roles. **Requirements engineers** a one role. **Security specialist** might be another role. And **stakeholders** are yet another role. So, as you go through a process like this. You need to keep in mind, who has what responsibilities and make sure that those responsibilities are carried out there, requires everyone's cooperation to do it successfully. So with that I’m going to bring up my slides again and go into slideshow.

13p - SQUARE Methodology

So here we go we're going to talk about SQUARE the nine steps, as I mentioned, and describe briefly what happens in each of those steps so that you can see how this process would work. It looks like a lot if you're going to do the full SQUARE you wanted to be on an important system, because it is time consuming. On systems that we worked on it, the elapsed time because of the meetings and reviews that were involved ended up being around three months, you might get some efficiencies if you are looking at a family of products.

14p - SQUARE

So SQUARE was developed by start with me as the principal investigator, we had support from SEI lab also at Carnegie Mellon University for the early research. And then we had some work that we did with clients quite a bit of education and training as well, just like this only to some extent more extensive we had. Three day workshops, where we taught faculty from other universities and we had one day or two day workshops that we did with industry and government organizations and some of those are still available for those who want to go deeper. So, as you know, it's a **stepwise methodology** for **eliciting**, **categorizing** and **prioritizing** **security requirements** and we treat **security requirements as** **quality attributes**, what I find with my students are sometimes they want to trade security off against something else. If you know that you're doing that and you've made that decision with this specific rationale that's fine. But it does need to be conscious decision and not just something that you fall into.

15p - SQUARE

So who's involved? The stakeholders and by this I mean all of the stakeholders, not just a few of them requirements engineers who have security expertise. When we do the deeper workshop we actually do a role play with some students playing the role of requirements engineers and others playing the role of stakeholders we've developed some tools to support were. Mostly prototype tools and we found that the roles were needed in order to be able to use the tools successfully. It just worked a little bit differently than when you were only doing face to face meetings. When you were sharing information online you really had to think about who was doing the why. So **we developed with them at the same time as functional requirements**, not as an afterthought in many systems people think about functional requirements first and then they tack on security, later on, as an add on patch does it work, at least not satisfactorily, so we treat them at the same time. **In the early stages of the lifecycle** whatever model you're using so if it's user stories you're going to have either specific user stories for security or you're going to incorporate it into other user stories. If it's a more traditional approach, you've got to make sure that those requirements are documented just as well as all of your other requirements. And I like to tell people don't just pick things at random from a checklist. it's okay to use checklists, but you have to know why you're picking this specific things, otherwise you might be incorporating things that you don't lead and at the same time, overlooking something that you do need not good.

16p-17p - SQUARE Steps

So let's take a look at the nine steps, I know you've already seen this, but not at this level of detail that will be going into. We’ve got a little bit on each of these steps and again, this is a high level view, if you want to get a more detailed view we've got much more extensive materials available to you as well.

18p - 1. Agree on definitions.

So the first step is **agree on definitions** it doesn't much sounds like it should be easy. But guess what lots of times people don't think about defining terms they assume a definition in their head. Which may or may not be the same as the definition that the person next to them has assumed in their head. As one example I worked with one client. And some of them didn't think about information security, they were thinking about government security plus locations. So information security wasn't something that they had spent time on when we got down to terms like availability. They had different ideas of what it meant for the system to be available. When it came down to privileges they had different ideas about who should have what privileges, so **all those important terms, need to be defined**. You can do that in group sessions, perhaps through interviews, but before you leave this step, you need to have a documented set of definitions, you can find quite a few good ones. And IEEE documents, for example, and the documents such as sweet box software engineering body of knowledge, and these are just a few examples. Of the kind of things that you need to understand that **make sure that everybody has the same understanding on the project,** otherwise you could get further down the road. And find out that you weren't on the same page and again have to go back and do the work, so this is something that I emphasize, no matter what kind of requirements you're talking about.

19p - 2. Identify assets and security goals.

So step two, **identify the assets and security goals**, when I first developed the method, I made a serious oversight which was to not think about the assets. It's or what you're trying to protect. So what would be the point of having this security goals? If you didn't know what it was, you were trying to protect. Both of those are very important. The goal, so I needed to help you later on when you get to prioritization and when the security requirements fit. If you don't have a goal you're just going to pick requirements, willy-nilly and not really understand what's important than what is it. You've probably got a **business goal** your company, so you certainly have business goals for your systems, which means the **security goals have to support it**, we did some work with an industry and they had concerns that were a little bit different than we were expecting they were concerned about competitive advantage, and so they wanted to make sure that nobody could get in and steal their ideas. We worked with the government organization that did research. And they were concerned with keeping their researchers personal information private because they didn't want a bunch of protesters trying to target their researchers. At the same time, they had had visiting researchers from other organizations, so they wanted to give them some access, but not the same access as their in house folks. So all those can lead into your **security goals they need to be reviewed prioritized and documented**, because when you go to prioritize the security requirements you're going to find that you need to think about what the goals are chances are. Not every project will have the same priority that every goal will have the same priority and ultimately not every requirement will have the same priority. And our **exit criteria** here are **one business goal and several security goals**, maybe five to seven from modern size system.

20p - 3. Develop artifacts to support security requirements definition.

So step three, **identify or develop artifacts**. I mentioned in the previous part of this discussion that we had customers who didn't have requirements documents, they didn't have architecture documents. We had one client, we were about to deliver a report on our findings. And they said 'Oh, by the way we forgot to tell you that we have remote access to our systems'. Well, obviously that's really important if you're concerned with security. And why didn't, we know that, because the architecture document was a day. Sometimes those documents don't even exist, and you might have to create them or help somebody else to create them. So make sure that current complete consistent all of those good things, and here we have just some examples of the kind of artifacts that might exist. And some of them might have to be created chances are if you're thinking about security for the first time you won't necessarily have misuse cases. Or attack trees or other artifacts for security available.

Examples> system architecture diagrams, use/misuse case scenarios/diagrams, attack trees, templates and forms

21p - Example of Artifacts - Misuse Case

Here's an example of a misuse case. I know that Jeff has already shown you a different diagram of misuse case. There are lots of ways of representing this use cases, you can do with a diagram, you can do with structured text, in this case, we have a system, this is one that we actually worked on and we showed how. A hacker was able to login using a spoof identity and do quite a bit of damage now this didn't actually happen because we were working with them on developing requirements, but it was a possible scenario, considering the architecture that they had so misuse cases can be very useful.

22p - Example of Artifacts - Attack Tree

Also, we’ve seen usage of attack trees. **Attack trees are interesting and they can also be weighted when you put yourself in the position of the attacker. You can see what's hard for the attacker to do and what's easy.** And then speculate well at least you want to get rid of the intrusions that are easy to prevent. And then look at the ones that are harder, so if it's easy for the hacker to get in you know. You don't want those things to happen. On another note, you could say the same applies to policy issues such as preventing phishing attacks, where somebody clicks on a link by mistake. Those can be pretty easy to prevent if you have good training, good policies in place, and maybe you save the user from themselves by not allowing those links to happen lots of ways to prevent that. So attack trees are another artifact that you might find useful. Again, if this is not familiar to you and you want to learn more about it, I can give you some references.

23p - 4. Assess risks.

Step four now is to perform **risk assessment**. If you haven't been doing much with security, there’s a good chance that you do some kind of a technical risk assessment on your project or a business risk assessment. If you've been thinking about security perhaps you've already identified the risk assessment method. We found several that we considered and we liked the method that miss National Institute of standards here in the US, a documented all most all of their standards are available for free download. So if you find that they're doing something that you'd like. By all means take it and use it. You might need someone with expertise in a risk analysis to go ahead with this. And the **exit criteria** for security risk is **documentation of all the threats** **their likelihood and their classifications** again there's a lot. In the literature on all of these much more than one when we started working in this area and that's a good thing. So, at the time we looked at a number of risky techniques and we really liked the one that we could get for free, which was the one that was available from this than and they've kept those documents up to date so they're still very useful.

24p - 5. Select elicitation technique(s).

Five is the law decide on the elicitation technique. You want to have something that's appropriate for the project. The number of stakeholders that you need and the expertise of the stakeholder. So you need to consider all of the stakeholders, when you do this. I think it's important to have representation from stakeholder groups involved in security requirements, you might find that your executives are met that available. Maybe you can drag them into one session and you might have to rely on others in later sessions, but one way or another, the illicit station technique is very important, because that is what's going to determine whether you get a good set of security requirements. And again, there are quite a few of these around we looked at structured interviews. A method that I really liked called **ARM (Accelerated Requirements Methods)** so it's a form of structured interview. But it's very well documented and also had industrial use **QFD (Quality Function Deployment)** is it one that's been around for a long time and is very popular. We found that **IBIS (Issue Based Information systems)** which is also been around for a while wasn't actually that useful to us, but this is something that you can consider there's. A lot available in this area, so you need to pick the right one, we actually did a tradeoff analysis of what we thought were the important features. To pick one. And that's how we ended up with ARM.

25p - 6. Elicit security requirements.

Once you've done that part of the process is actually **elicit the requirements**. Or otherwise identify them you want to execute that technique. **Try to avoid vague requirements** so security field, unfortunately this film was vague requirements, but you really have to be able to verify them in some fashion. And make sure they're not ambiguous. So it's tricky but it's really important. Again we're at **the requirements level so think about what it is you want to** **do**, not necessarily a mechanism, sometimes you have to specify a mechanism. But it's important to think about exactly why you're doing this. And not so much about what the mechanism is you might have to think about mechanism and then back up to the requirement that's Okay, as long as what you're documenting is a requirement and not the implementation. So the **exit criteria** here are an **initial set of requirements**.

26p - 7. Categorize requirements.

Next, we found was useful to **categorize** them because very often requirements appear in separate area so might be a requirement that the system level might be at the software level or could in fact be a constraint on the architecture. Certainly, you want to minimize the number of architectural constraints, but if it has to be that way, then it needs to be documented as such once you've done that then you can go to step eight.

27p - 7. Categorize requirements: example

We have in step seven some examples of how you categorize requirements. And this is just one illustration.

28p - 8. Prioritize requirements.

So step eight **prioritization** we looked at a number of methods and decided to use the **analytical hierarchy process** for HP. For a while there were tools to support it, they still may be out there, there are a lot of other parts ization techniques as well, so if you do decide to do this, you might revisit the topic and see which technique will work best for you. I recommend doing a cost benefit analysis to help the stakeholders, because, after all, once you do that prioritization you will probably find that you have more requirements, then you have budget for. We worked with one client and they said we really like all these requirements, but we're going to have to implement them according to a priority scheme. We can't afford to do all them once, even if we had the money we don't have the staff to do it. So that's a consideration, I always recommend prioritizing requirements. Even if it's a security requirement if you think an event is unlikely, the risk is low, maybe there's a requirement, but it may not matter that much.

29p - 9. Inspect requirements. (Reference of Fagan inspection: <https://blog.naver.com/ryution/42889826>)

And then the last step. **Inspection**. We tried some experiments and we found that formal inspections, such as a **Fagan inspection**. Worked a lot better than **peer reviews,** in the sense. That more problems for discovered tracked and fixed. Peer reviews can be effective, but if you're pressed for time, sometimes they get short changed. I frequently find with our students that when their schedules get into trouble, the first thing they want to do is skip inspections. Bad mistake. And of course we're talking about **inspections at all stages of development** not just one talk. The **exit criteria**, all the **requirements have now been selected and documented**. What we did with our tools was we actually developed a tool that supported the nine steps. And that got us into this situation where we had to consider the roles and exactly what kind of support, we would include for each step and we did that with several teams of students and ended up with, I thought, a pretty good prototype tool.

30p - Approach

So it took us about three months of calendar time to complete development of security requirements for mid-sized project but keep in mind that's not all we were doing. They were developing their functional requirements at the same time working on existing versions of their system, we had to schedule meetings when we could get hold of stakeholders all of those kinds of things, so it did take a while. We’ve implemented the method in a number of case studies and then industry and then some government organizations. Lots of people use it, based on the documents that we have on our website, and so I don't necessarily know about that unless somebody asked me a question which is always a nice bit of feedback.

31p - Reading Assignment: SQUARE Technical Report

So with that here's a reading assignment. I suggest that you skim this because it's pretty long and definitely take a look at the case study that's in there. It’s a technical report that you can download from the SDI website, the link is right there so by all means go ahead and do that skim the report look at the details of the case study and think about what aspects of this, you might want to use. That’s it for now that's the end of this part of our discussion or security requirements and I’ll be back with you for another little briefing on the remaining elements of SQUARE.

Part3

Intro

hi this is Nancy meet again we're now on our third and final part of the discussion about security requirements and SQUARE in this session will be talking about SQUARE-Light which are you referred to earlier and A-SQUARE, which is where for acquisition. Regardless of what method you use it's important to have a process for defining security requirements, whether or not you use SQUARE another method. or create your own is less important than having one and I’d like you to consider all of the activities that we've discussed if the nine steps are not for you, you might just pick a few that seem relevant to your process and add them. Or you might do something different altogether, my main message is that you need to be thinking about an identifying security requirements and have a process in place to do that so with that we're going to take a look at our PowerPoint slides, and I will be.

32p-33p

As promised we'll be talking about SQUARE-Light and A-SQUARE which I think is certainly relevant to today's environment. A-SQUARE-Light was an attempt to come up with a version of SQUARE that was not quite so Labor intensive we probably mentioned that SQUARE the nine steps took us three months of elapsed time for a medium size project, so in that regard, it is a lot of work, and it could be considered too heavy for some organizations, if you already have some process in place, then you might consider using SQUARE-Light. This is five steps, instead of the nine steps, assuming that some things have already been done so, we still need to focus on definitions assets and security goals. But perhaps you already have artifacts in place and so there's no need to create them, that would be great. Perhaps you already have a risk assessment method that considers security, in which case you can just go ahead and do that risk assessment. And perhaps you already have an analyst rotation method or other similar method that you use in to actually identify the requirements. And you can go ahead and do that, and then, of course, no matter what you do you'll want to prioritize those requirements we implemented SQUARE-Light in one case study for medium sized company it had about 1000 employees and there were at least 100 were engaged in software development. Unfortunately, they did not have a development process in place, they were strictly scheduled to urban and so they were having the idea that they could incorporate this as a start for developing processes. it didn't work you have to have a basic framework to insert this into you can't just take the shortcut and hope that that will be satisfactory. Over the course of that project we saw a lot of people leave the company, the ones who stayed were very frustrated. And they had quite a bit of turnover employee, so it wasn't a good situation doesn't mean that SQUARE-Light is bad, but it does mean that you can't just use that as a quick fix you really have to have on the processes in place before you add this in.

34p

Now very interesting these days is SQUARE for acquisition. We had the idea of modifying it for use in acquisition and we also thought that whatever we came up with should be consistent with other acquisition processes that are being used by the company.

35p

Interestingly, it was not a single method we evolve to three cases in the first case, we will be discussing each of these little bit the acquisition or organization. Has a typical client role they're kind of hands off from the actual development of the software which is most likely to be custom software, to some extent. And so that's sort of an easy case to deal with because it's very traditional the second case, is one that we often see when they are incremental processes in place for actually developing software and that's where the acquisition organization, does the requirement spec. And then hands it off to the contract. So that's a little bit different. Sometimes it's a tiered acquisition, where one contract and might be the work to do, the requirements engineering and somebody else is actually doing the development. And then the third case which we know is have a lot of interest these days is acquiring cuts software, it could be a standalone or it couldn't be a part of a supply chain and we've certainly seen enough issues with the use of cuts software and supply chains

36p

So with that let's get started in case number one the contractor is responsible for requirements definition. We expect them to be on board at the start of the contract and the acquisition organization is playing the role of a typical client. They kind of hands off they get involved in the review process, they get involved in looking at it from a user and client perspective but they're not actually developing the requirements.

37p

So here's one example of how the workflow might look. On the client side they're still agreeing on definitions and identity identifying the assets and security goals. The contractor might go ahead with the remaining steps of SQUARE. And then, when the requirements have been reviewed in house, they can schedule a review session with the client. If you've worked on older style projects more of a waterfall process, this is the kind of thing that you might see. And hear what we're indicating that the contractor could be using SQUARE we've had to add an additional step we might not need that additional step if the client is working closely with the contracting organization.

38p

And those are the important points that we see here, moving on.

39p

In case. The contractor decides to use a different process than it's kind of a black box. The acquisition organization is still performing the first two steps the contractor comes up with the requirements and then they have a joint review process. In both of these cases it's also possible that the acquisition organization might be involved in the risk analysis and look at the results. Of that risk analysis so it's not a hard and fast rule, these are just typical cases that we might see when the contractor is doing all the work and the acquisition organization is only doing a review. Once again it's typical of what we might see in custom software.

40p

Moving to A-SQUARE we've got two cases. The first is the incremental development case where the acquisition group specifies what the requirements are as part of the RFP. We’d like to see them use SQUARE or whatever they are defined security requirements process is so that they have a good set of security requirements for the contractor that saves everybody a lot of pain and harvick. Now we have the same question that we had before what we were looking at SQUARE, and that is, we don't want to be too prescriptive. And we don't want to specify the requirements in so much detail that the contractor can only implemented one particular way unless it's really necessary. So here, you have to be careful about granularity if you've got that contract in place but you're not the one who is doing the development.

41p

And I think we covered all of this once again the exit criteria is the final review and approval by both parties.

42p

Case three these days is the most interesting one we're going to be looking at acquisition of COTS says kind of a standalone application. But, as you all know, we see COTS being incorporated into various parts of the software supply chain and you really want to make sure that you know that you can trust that product. So some of these are things that you might do if you're buying it a COTS product some of these are things that you might want your contractor to do. If they are providing a larger system in which COTS products are incorporated, and these are just some obvious ones spreadsheets databases document management systems emails. And we have examples of all of those where people didn't consider security and, in some cases they had to throw out the product and start over again after they found out about it and spent a lot of money.

43p

So case three looks different. We’ve got our same first two steps agree on definitions identify assets and security goals. Then, what we want to do is to identify a candidate set of requirements for the security of that COTS product. What you're likely to find is that there may not be a single product that meets all of your requirements so you'll have to do some more work. The first thing you'll need to do is to review the specifications for the product. If you're lucky, and maybe it's open source. Then you can actually look at the code to see what it does, otherwise I would suggest you look at the vendor’s documentation and also asked for a DEMO of the system, or perhaps a short term time period to try it out and see what the features are and get an idea of what security is built into it. Once you've gone through that exercise, you can go back and decide what requirements are really important too, because there's a good chance that a single product won't meet all of your requirements. And that involves a prioritization scheme that you're going to use when you go through trade off analysis write your final spec and make a product selection.

44p

Here’s another way of looking at this particular case. The prioritization is really important, and you have to consider security versus other quality requirements when you're doing this. you've got a tradeoff analysis that you'll need to be done, I began suggest prioritizing both the security requirements and the other system requirements that go along with it, because you need to look at that entire group. And then, finally, the review process might help you to identify things that you hadn't considered at the beginning. Lots of people use very poor reasons for acquiring a COTS product. They might pick one because they've used that vendor for other COTS products well the particular application for this new COTS product may not be similar it might not be the best vendor maybe you need to have multiple vendors and pick a different vendor for a particular product.

45p

So what we talked about, we talked about requirements in general and the problems that we see on our project security requirements in particular. And lack of attention to security projects can also cause the projects to fail, we talked about SQUARE, which is one method for developing security requirements, I gave you some references for other methods and then we talked about SQUARE-Light and A-SQUARE.

46p

That’s kind of summarize here the one aspect of SQUARE that we didn't talk about in detail was P-SQUARE, which is interesting, but it's very much dependent on having a tool to support it. A lot of work has been done on privacy requirements, since we did that work, so if privacy requirements are in the area of interest, I will need to point you to some other references there's a lot of research work in that field, particularly in the last few years.

47p - Homework Assignment

Now this is what you've been waiting for a little homework assignment. And this is just for practice what I’d like you to do with your project team for the course is to identify a type of COTS product that you think you will need, and it could be something like an email, database, spreadsheet, document management. Doesn’t really matter just pick one identify two to three vendors of that product type. And then use A-SQUARE the seven steps that we just talked about to evaluate which vendor product is the best fit for this project. Now this is just for practice you don't necessarily have to use the results on your project, but I want you to get a sense for how it goes when you go ahead. And work through those seven steps and I guarantee you that some of the steps will be useful to you, even if you decide the whole thing is. Maybe not that useful or too much work.

It’s been a pleasure to be with you and I know that I’m going to see you again in just a short time, when we talk about threat modeling. So, once again, this is quite a bit on security requirements, but it's in a way it's just touching the top of it. There are a lot of references that I can provide for you a lot of materials, if you decide to do training in House we have training both available, and we have materials for training that you can take and use in house. I’ll be glad to help you in any way that I can in this area.

48p - Additional Resources

Here so some additional resources, there is a whole library of SQUARE documents on the SEI website. And there is a book that we wrote called software security engineering a guide for project managers that includes a chapter on security requirements also with the discussion of SQUARE. So these resources at the SDI library, of course, are available for free download. Amazon, as most likely they're paperback or e-copies. Of this book available if not I might be able to conjure up a copy but it's been out of print, so I don't know if I’d be able to do that or not, but I can try. So thanks for being with me I’ll see you again shortly, well, we talked about threat modeling.