Q&A from “Transitioning from Waterfall to Agile” Web Seminar

- How does this method allow you to provide the client with a budget that they can depend on at the start of the project?

  ASK: Because the Agile approach assumes that projects are “time-boxed”, the budget is not subject to change (assuming you aren’t adding people or other costs as you go). Time-boxing establishes a specific schedule for the project, and so Scope is what is managed on time-boxed projects; delivering more or less as resources and time allow.

- Isn’t organizational change - cultural change - a challenge? Any suggestions for an approach on that?

  ASK: Absolutely! The Agile approach is a shock for many stakeholders (even for developers). Getting people to act differently (to change their behavior) is much more challenging than learning the details of the Agile practices. The Transitioning From Waterfall to Agile course is focused on this issue. In this webinar, we touched on the benefits that you would use to entice people and the challenges you will encounter in trying to move toward Agility. The 2-day class by that name spends most of its time on those points because that is the challenge!

- Are there samples of flexible contracts to use for agile development?

  ASK: I have not yet seen any.

- Are agile projects always completed on time?

  ASK: Following the assumption of time-boxing (see the 1st question, above), yes! The operative question is, what precisely will be delivered by the project’s end, and will it be everything that the customer needs?

- What is the difference between agile coach and the traditional project manager?

  ASK: Because Agile teams self-direct (planning their work, checking their status and taking corrective action), the Agile coach is not absorbed in these mechanical aspects of management. Instead, the coach is ensuring that the team is doing these things as well as collaboration and the technical work as effectively as possible. In other words, freed of the mechanical aspects of management, the Coach can focus on the other things that managers agree they should be doing, but often have little time left for.

- How can you make sure the sum of the iterations are delivering to the overall project scope, schedule, and budget?

  ASK: That is the reason why the customer is so critical in planning what gets done when in an Agile project. But as a Coach, I would be coaching the customer too; ensuring that the progression he or she is leading the team on is indeed tracking toward fully meeting the business need.

- Who will train the team to self-manage and how?

  ASK: Everyone who will participate on an Agile team should be trained on his and her roles in this new approach. So developers need to be trained to write good stories, estimate them,
break stories into tasks, etc. But training is only part of the picture – the other part is the doing of it. The Coach is the person who is ensuring that the team members are doing their part to manage themselves appropriately.

-Who defines requirements aside from customer, information architects, usability designers or developers?

ASK: The Agile approach defines the role of “Product Owner” – an individual who represents all of the customer and business needs to the team. This approach does not address the question of whom this person interacts with and how he or she gets all of the relevant information. This is the reason why the role of the Business Analyst (BA) is still relevant on Agile projects. Finding all of the inputs the project needs may be its own problem, deserving of someone’s full-time attention!

-What about when a system or product feature simply can't be engineer to a shippable state in under 2-6 weeks?

ASK: 99% of functionality can be sliced into stories that fit nicely into short iterations. There is no need for these “thin-sliced” stories to be fully usable in a production environment; only that they be demonstrable to our customers so we can get feedback from them. Each iteration ends with production-quality software, but not necessarily a feature set that could be used in production. For this reason, most Agile teams only release to production after every 3rd or 4th iteration.

-What is the basis of defining each iteration? Is it based on individual requirements or use cases, a combination or something else?

ASK: If you mean “How do you decide on an iteration length?”: It is arbitrarily chosen by the project team. Most teams either choose a short iteration (e.g. 2 weeks) to maximize feedback, or they choose an iteration length that matches their status reporting period (e.g. monthly).

If you mean “How do you decide what will be done in each iteration?”: It is based on priority, with the highest priority items that have not yet been addressed going into each iteration. Priority is mainly based on Business Value to the customer, but can also include addressing technical risks or questions early to increase the likelihood of project success.

-If Development deliver pieces of code to QA.. How do you know from a daily basis that something is ready for QA, or is soon coming to QA? Would we use the Time Boxes, and the Radiators?

ASK: There are two approaches to independent QA that organizations with QA groups choose among:

1) Integrate the Tester into the team. This way, there is no throw-it-over-the-wall, because there is no wall. The tester knows the status of everything that is being worked on, so he or she automatically knows what can be tested each day.

2) Run QA as a parallel Agile team. This way, the developers build and fully test their work in iteration X. At the end of that iteration the customer accepts the functionality. Then in iteration X+1, the QA team does full QA testing of what was accepted (while the developers are developing the next increment of code).
The second option is less Agile (because it builds a wall between the developers and QA), but it can be necessary when QA testing is a significant effort due to the need for regression or other specialized testing.

-Does plan iteration include wireframes and specs? Or is this just development?

**ASK:** Each team (or organization) must define what “done” means. If that means we need documentation of some sort, or we need the customer’s feedback on a wireframe before the final code it laid, then yes, we include those things.

**BUT** we must be careful that we don’t require these things just because that’s the way we do things in traditional methods. With Agility, investment in each story is low (because they are so small), and feedback comes quickly. This means that a lot of the formality we need in traditional methods is simply not useful on an Agile project.

-What is your definition of the size of a small team?

**ASK:** No more than a dozen technical contributors (counting all disciplines).

-How do you accommodate key resources off (e.g. time off, sick etc.) for the daily standup?

**ASK:** When someone is out of the office, he or she misses that day. When the individual returns, he or she can look at the project status to get a quick update on what was missed. Also, talking with other team members and/or the coach may be necessary after an extended absence. (This is really not different from any other project method.)

-How are functional requirements handled?

**ASK:** Functional requirements are defined as User Stories. Any action the system must take can be stated as a User Story. For example, “As the corporate Security Officer, I need Level 3 Password strength enforced for all users, so I can ensure the safety of corporate data.”

-Are you not training the customer that they have their own development team? As the customer learns and gets instant gratification, is there really ever any end to an Agile project?

**ASK:** Time-boxing (refer to question 1, above) is **usually** designed to ensure that the customer gets what they need by the date they need it. But it can also be used to limit the customer’s access to the team. In organizations where IT is viewed as a “free” resource, it is useful to establish a time-box for each project to ensure that the various customers who are queued up for projects each get their turn. “This project will be four months long. The team will produce whatever you as the customer deem to be most valuable in that time. At the end of that time, the project is over and we will move on to the next customer. If you need more done, you must get back in line.”

-What is CMMI?

**ASK:** I’m sorry! I keep forgetting that not everyone has been eating, breathing and living in the same world as I have! CMMI is Capability Maturity Model Integration®, a model of mature processes for engineering projects. Agile practices satisfy most to the CMMI’s goals if they are interpreted properly. Unfortunately, many CMMI Appraisers don’t understand Agility well enough to properly interpret what an Agile project is doing.
-What is the difference between Agile and RUP?

**ASK:** RUP has **many** of the same attributes as the Agile methods. It is **not** Agile because it does not include incremental delivery and acceptance. Agile Unified Process is a variation of RUP that adds incremental delivery, making it an Agile method.

-Can you provide a sample of a contract that would replace the original scope of work & budget contract?

**ASK:** I have not yet seen one.

-In Agile, who is the coach? Is that the same as the scrum master in the scrum method?

**ASK:** “Coach” is a generic term we use to refer to what Scrum calls “Scrum Master” and what eXtreme programming calls “Tracker”. Most organizations move either Project Managers or Team Leads into this role.

-How does this method work with teams distributed across the world (aka India/China). Very difficult to have daily meetings.

**ASK:** When work is distributed so far that the teams do not share any work hours, it is often best to run them as two parallel Agile teams (e.g. using Scrum-of-Scrums to coordinate them).

-I would like to get some guidance on "thin-slicing" requirements to make this work. Seems like an art in itself.

**ASK:** There is a bit of art to it, not unlike a **lot** of programming work! It is a matter of getting creative in looking for ways to partition functionality along boundaries that still allow each slice to be demonstrated. For example, if “Add a new customer” is too big because there are so many validations and business rules to apply, I could slice it like this:

1) Add a generic customer with no error checking
2) Add basic error checking to the Add a Customer function
3) Add the Business Rules for a class of customers to the Add a Customer function
4) Add more Business Rules … and so on

Each of these has value to the customer, can be tested and can be demonstrated to the customer and accepted by them.

-Are there some general ratios on lengths of projects vs. design changes and refactoring?

**ASK:** I know of none. When considering if we will need to plan for a stabilization Sprint to refactor the system, I look at how many technical risks and unknowns there are. If there is a lot of risk, it is a good bet I should plan for refactor time at some point in the project!

-Can you please give examples of "Information Radiators", actual types of status and format?
-What % of the IT industry is currently following Agile way?

ASK: I have not seen statistics. The percentage is quite low at this point, but gaining momentum. I expect to start seeing statistics on Agility within the next two to three years.

-Are there types of projects that better fit an agile methodology? What types of projects are those?

ASK: The more uncertainty there is about the project, the more appropriate Agility is. If you can predict with certainty precisely what the project will produce, when, and at what cost, then Agility doesn’t buy very much for you. (But I have never seen such a project!)

-At the end of each iteration, is the software discarded most of the time? when is it, when is it not?

ASK: The result of each iteration should never be discarded. It should be production quality, and be the baseline on which the next iteration is built. Thus an Agile project grows the system by accretion, just as a clam grows its shell. The only time the results of an iteration would be discarded is if it was a total failure (produced nothing of value).

-How do you use scheduling tools, such as Microsoft project in the Agile method?

ASK: The Agile approach makes every possible use of the fact that most of the work on software projects can be done in just about any order. When there are dependencies, the prerequisites become higher priority (and are done earlier). So it is rare to see a schedule (in the traditional sense) for an Agile project.

-Is change control and impact of the change in the requirements, part of every iteration? if so, who is responsible for assessing the impact of the change to the dependent systems?
ASK: Indeed! Change control is the essence of what adapting to change at the beginning of each iteration is all about. The developers articulate the impacts and estimate the cost of the proposed changes, and the customer uses that information to prioritize the change into the existing project backlog.

-Do you think it only works with senior-level developers and testers?

ASK: Agility is just like any other method. The more capable my people the better. I would never want a team comprising nothing but rank novices! A few senior people on an Agile team is sufficient, because the collaboration will ensure that key technical decisions are discussed and those senior people have appropriate input.

-How do hardware infrastructure projects embrace Agile methodology?

ASK: The Agile methods as they have been optimized are designed for software development and maintenance. But they can be used to manage anything. (Scrum doesn’t even mention software – it is an Agile project management method.) Applying the Agile methods to any non-software project requires re-inventing the details for the domain you are in.

For Hardware Infrastructure, the key tailoring I can envision is in how you write the User Stories, and you thin-slice them into small deliverables. (I have never managed that kind of project, so I have no answers off the top of my head.)

-When do we do the final testing? after the final product is done?

ASK: The intention is that we do all testing within each iteration so that the product that comes out or each iteration is production-quality, and could be released if the customer chooses to do so. If your product is so big or has so many integration points with other systems that full testing during development is not possible, there are two options:

1) Run “final testing” in parallel with development. The test team is doing final testing of the software produced in iteration N during iteration N+1, while the developers are developing new stuff for iteration N+1. This way, you have deployable software one iteration after the customer has accepted it.

2) Do periodic “Stabilization Sprints”. After every third or fourth sprint, run a sprint that does nothing but testing and preparation for release. Again, this allows you to get the product into releasable form quickly.

-What mechanisms exist for defining customer pools to show iteration outputs to, allow them to prioritize requirements and help define iteration goals?

ASK: The Agile methods want a single person to be the voice of the customer to the team. How that person rationalizes the competing needs of all of his or her various constituencies is left to that person to figure out. Agility does not address this problem.

-What if the feedback and new requirements from customer drives the development need beyond scope and timeline? To what extent do you welcome feedback and new requirements?
ASK: There are actually two questions here; scope & time.

Scope: If the customer makes requirements changes that change the scope of the project, that needs to be addressed at whatever level approved the project in the first place, because it is no longer the same project. The new definition of the project must be approved and resourced appropriately.

Time: Agility assumes time boxing (see the first question), so when our customer wants to add something new, it will generally mean that they must trade something for it. Nothing is free. If changing the project’s time constraint is appropriate, it would have to be approved just like a Scope change, because again, we are going outside of what was originally approved.

-Do you find that in practice, during one cycle the analysis and design is performed, and the next cycle includes the development and testing of those pieces? (Meanwhile analysis/design of the next piece is being done - to feed next cycle)

ASK: Almost always, no. Each Story (requirement) should be small enough that it can be designed, coded tested and anything else that is required for “production-quality” within one iteration. There are times when this rule must be broken, but we keep those instances to a bare minimum. Define a small chunk of work, then do it quickly and get feedback for then customer ASAP.

-When do we get to the transition? This is more goodness about agile, but I am not seeing the transition guidance that was advertised.

ASK: In the webinar, I ended up spending more time on “what’s agile” than on the transition issues – mainly because of the time constraints. I alluded to the transition piece for each topic by listing the Benefits we use to sell Agility and the Challenges we face in moving toward it. In the 2-day class by the same name, the majority of our time is spent on the Transition issues.

-So developers write test cases? Is there a QA resource involved?

ASK: The Agile methods don’t mention testers – they place the entire testing burden on the developers (except for Acceptance testing, which is the Customer’s responsibility). Many organizations have testers, and the Agile community is in general agreement that this additional role has much to offer an Agile project.

The form that the Tester-developer relationship takes and the details of how it should work are current topics of discussion and experimentation. The one thing that everyone agrees on is that the existence of Testers on a project in no way reduces the need for the developers to focus on testing and other quality issues.

-How does budgeting and estimating work in the agile approach versus the waterfall approach? More specifically we traditionally year mark funds for mutli-year projects. How does this work in the agile approach?

ASK: This extra-project (or per-project) activity would not be changed by the use of an Agile project approach. That multi-year plan would identify the series of projects that will be undertaken to complete the product, then you can use an Agile approach to run each project.
-Documenting requirements - is that part of the agile approach?

ASK: Requirements are documented on Agile projects – just in a form and way that differs from what we traditionally expect. The requirements are progressively elaborated starting with high-level course-grained statements (e.g. 3x5 cards) that are used to plan the top-level shape of the project. More detail is added as needed during the more detailed iteration planning activity, and the final level of requirements detail is worked out as the code is being written and captured in the form of test cases.

-What is the BA's Role in Agile Dev?

ASK: The Agile methods do not mention Business Analysts. But they are often the right people to play the role of the Agile team’s “Product Owner” (the customer representative). Although the Agile team would prefer to interact directly with their customer, this is usually complicated because they have not one, but multiple customers, and those customers are often not as available as the team needs.

- Agile is obviously a culture shift for many organizations, I'm wondering about budgeting / estimating. What is the best approach at providing overall estimation when you are not doing detailed task analysis for all requirements (outside of iteration 1)?

ASK: Time-boxing (refer to the first question) combined with a set team size will generally mean that an Agile project’s budget is very predictable.

-Where do you incorporate the architecture pieces of the project with the Agile approach where this work typically doesn't have anything 'visual' to review with the customer?

ASK: The Agile methods prefer to allow the Architecture to “emerge” as the system is built. In small and less complex systems this is a good approach. But when dealing with a very large system, much complexity, or other complicating factors, up-front analysis is usually warranted. In those cases, an “Iteration Zero” precedes the actual development.

-How do you see training resources for end users fitting within and working with the agile team?

ASK: If training is required, then it should be identified as a deliverable of the Agile project. In that case, Training is one more discipline that is represented on the team, and the training materials that are needed to support each Story are included in the team’s definition of “done”. (The Story isn’t “done” until all of its Done-ness criteria have been met!)

-How many transitions from Waterfall to Agile fail?

ASK: I wish I knew. People only report successes. And to date, I know of no broad-based studies of this question.

-Approx what percentage of each iteration are the planning, requirements, construction and testing portions?

ASK: An Agile Team will generally time-box the Planning and Requirements activities at a day or two for a four-week iteration (5-10%) to keep from spending too much time on them. Testing is usually intermingled with construction, and practices like Test Driven Development (TDD) make the percentage mix vary widely.
-Do you have any good examples of creating user stories rather than use cases for requirements - something that shows the requirement, what a user story would be and what a use case could be?

**ASK:** The User Story is generally going to be a single sentence that captures the essence of the Use Case. “As a {user role}, I need {name of function}, so that {business need}.” When the team needs a more detailed understanding of the User Stories (e.g. to plan the tasks), they will often elaborate each Story into a Use Case.

-In agile approach, what is the process of "welcoming change requirements"?

**ASK:** Changes to the requirements are captured whenever they occur. Then during the next “Adapt to Change” phase of the lifecycle (refer to the Agile Lifecycle slide), it is estimated and prioritized, and the worked into the project according to its priority.

-At the end of each iteration, is the software discarded most of the time? when is it, when is it not?

**ASK:** Just the opposite. Each Iteration produces production-quality software that is at least the baseline for the next Iteration, and at best is actually put into production!

-How do you manage change during the course of an iteration? Should changes always be implemented during the next iteration?

**ASK:** See the answer two questions above here.

- Did I hear that delays within the iteration result not in delaying the deliverables, but in reducing scope or leaving out features, with the customer's approval?

**ASK:** That is right. An Agile team adheres with their time-boxes. They do this by managing the deliverables (collaboratively with their customer).

-Any templates for client status reporting?

**ASK:** No. Agile project usually track their status using the charts shown in answer to an earlier question. Any out-ward reporting is done as required by those who require it.

-What suggestions do you have for creating an SRS from user stories since in SOW work we need to provide an SRS for signoff from our customers prior to starting projects?

**ASK:** That is not Agile – that is plan-driven. If your customer refuses to collaborate with you on the project, then you can’t implement 80% of Agility.

-Are there characteristics of a project that make it a good candidate for Agile processes vs more traditional processes (building a new system vs enhancements, etc)?

**ASK:** The more unknowns there are (requirements, technology, team capabilities, business dynamics), the stronger the argument that an Agile approach should be used. The only projects for which an Agile approach is wrong are those where the customer refuses to collaborate. You can’t be agile under those circumstances.
-Is there a best practice for using MS Project while doing an agile project? Reporting?

ASK: MS Project is rarely used on Agile projects. A task list is usually sufficient because there aren’t that many dependencies on most software projects. (If there are, then you may be working with a bad architecture!)

-Can you still maintain CMMI levels by allowing teams to define what is needed for documentation?

ASK: Yes, assuming your CMMI Lead Appraiser understand the Agile practices and can interpret how they are contributing to achieving the PA Goals. (There are such Lead Appraisers out there, but many do not understand Agility, and will dismiss Agile practices out-of-hand.)

-How do you use scheduling tools, such as Microsoft Project in the Agile method?

ASK: See the answer, two questions back.

-How do you manage 2 - 6 week iteration if new scope is introduced during the incremental requirements efforts?

ASK: The team is insulated from change during each iteration. Changes are integrated into the project during the “Adapt to Change” phase of the Lifecycle (refer to the Agile Lifecycle slide) at the end of each Iteration.

-Is the daily stand up meeting the best time to capture remaining hours on a burn down chart?

ASK: Yes. The Team’s Coach usually updates any charts during or immediately after the daily stand-up.

-Instead of moving your organization to go completely agile, what are you thoughts around incorporating some agile practices into an existing traditional waterfall organization?

ASK: Gradual process change is often the way to go. The hard part is that until you move to iterative development of product increments (a major shift for most organizations), many of the Agile practices won’t translate well.

-Is there a risk of the number of iterations going out of control? Is there ever too many or too few?

ASK: The project time-box will define the number of iterations up front, and the Agile team will stick with that (unless the customer and management agree to change it). Fewer than four iterations in a project is a stretch, because you aren’t getting feedback from the customer often enough. There is no such thing as too many, but if the project will extend for more than a year, I would prefer to break it into a progression of separate projects. (But that’s just me.)

-Is there any difference between Agile and older, collaborative methods like JAD (Joint Application Development)?

ASK: Nothing in Agility is new. We have drawn practices from many older approaches. In general, a method is only “Agile” if it delivers working software regularly. (JAD usually does not.)
-Could you expand on your previous comment about "use cases not quite getting it done"? What suggestion would you make to use instead of use cases?

ASK: I don’t recall saying that, and I can’t imagine the context where I would have. User Stories really are pretty much the same thing as Use Cases.

-Does up-front technology architecture still fit within the agile processes? At what phase?

ASK: See my answer about “Iteration Zero”, three pages back.

-So, for a fixed bid, you are really only fixing the number of hours. There is no guarantee of what will be delivered?

ASK: The guarantee is, “We will collaborate regularly with the customer to ensure that we deliver the greatest possible business value given the project constraints.”

-Rather than move your team to go completely agile, what are your thoughts around incorporating some agile practices into traditional water fall environment?

ASK: See my answer 6 questions back.

-Moving from Waterfall to Agile is a transition that affects all areas of the organization. What small steps can you take along the way to begin to move to a Agile approach knowing that it is a big shift?

ASK: The beginning point is in tearing down walls that separate the customer from the developers and the members of different disciplines. That will allow the collaboration and feedback that is needed to do the rest.

-Are there success stories that will prove Agile a much better approach than Waterfall?

ASK: Of course people like to talk about their successes! There are many anecdotal reports of Agile success. A colleague of mine collected a variety of research on the Agile approach into a research paper (http://davidfrico.com/rico08a.pdf). Please forgive its research-y-ness, as he just got his PhD, and still writes like he is in school.

-What is the best way to adapt this process to single person projects?

ASK: The key is to collaborate closely with your customer, deliver software to them regularly, get their feedback on it and adapt to what you learn from them. The team practices are moot with a one-person “team”, but the customer collaboration part is still important!

-You keep referring to project instead of product. does each release become a "project"?

ASK: I guess I wasn’t clear. The Agile project is iterative, going through the lifecycle loop every few weeks. Each Iteration produces production-quality product that the customer can accept and use.

-Are there any guidelines for capitalization of development hours?

ASK: The Agile methods don’t address these sorts of financial questions. And I don’t believe the answers would be any different for an Agile project than for any other approach.
-Any tips on thin-slicing a major deliverable?

**ASK:** You start with individual Use Cases. Often each Use case is a thin enough slice of functionality to work as a Story. If a Use Case is too big, it is usually because there are many exceptions and alternative flows that must be accounted for. In that case, you can do one User Story about the “happy path”, then follow it up with User Stories about the error and alternative cases.

-Is that mandatory that tester needs to involve completely during the Unit testing with developers?

**ASK:** No. The developer is responsible for Unit Testing. But having a tester work with the developer is a good way to ensure that the unit testing is effective!

-When there are delays within the iteration, do you find that project teams actually return to the customer to ask them to pay more or reduce features / scope? Isn't there a risk that the project team will instead start cutting corners, reducing quality?

**ASK:** The **right** thing for an Agile team to do when they cannot complete the promised work is to get together with the customer and de-scope. Asking the customer to pay more is not the usually Agile approach, although is the customer decides that is preferable over de-scoping, then it is the right thing to do. A **big** focus of Agility is “Technical Excellence”, so cutting quality corners should **never** be an Agile team’s response, and the team’s coach must watch to be sure it doesn’t happen!

-What are the challenges of implementing Agile for backend systems (i.e. legacy, mainframes, overall architecture), and how do you solve for those challenges?

**ASK:** There are two main challenges. 1) The tool sets in those environments often don’t provide sufficient support for automating the build and testing process. When those things are manual, the team can’t do them as often, which slows down the whole feedback loop. 2) Defining Stories that are small and demonstrate Customer value can be more challenging with big back-end systems. Thin-slicing Stories and still be done, but can take significant thought and creativity.

-For large projects in corporations (i.e. > $50,000 IT costs), is it ok to have multiple iterations going on at the same time? If so, how do you structure the separate teams and how do you assign features to those teams? What are the dangers?

**ASK:** Scrum (one of the Agile Methods) defines a team of teams structure for handling these sorts of projects. It is called “Scrum of Scrums”. This is one of the lightest-weight approaches I have ever seen for coordinating the work of many people.