Maximizing the Business Value of a Test and Measurements System in Support of a NPI Process

Filipe Altoe, PMP

Principal at TSXperts (www.tsxperts.com)

Introduction

One other scenario that is fairly common at the T&M industry is when, in support of a client's NPI, a T&M system needs to be designed and built prior to the actual device under test to be designed and built.

One may question the feasibility of the approach; however, the fact of the matter is that the vast majority of the NPI schedules always assume the test system will be ready to be used as soon as the product development is complete. This sound like an impossible task and a recipe for disaster as it violates the very root cause for T&M project failure; lack of well established requirements.

In fact, it can be an impossible task and a recipe for disaster, depending on how the project gets executed. Historically, organizations who execute test and measurements projects have been applying a typical Waterfall approach to managing these projects. The concept of Waterfall Development lies in the idea that the overall project can be broken down into different phases and that work needs to be brought to completion in one phase in order for work on the next phase to be initiated. For instance, all requirements need to be fully defined before system design can start.

Waterfall Development thus emphasizes more time spent up front in the project lifecycle, highlighting the importance of requirements and design documentation. It is easy to see why organizations try to brute force this methodology into NPIs. NPIs obviously have overall constraints, such as schedule, budget and scope, much like a project has. The test and measurement project is just one of the projects that need to be managed by the NPI program. As such, the NPI manager will definitely need to have some idea on the test and measurement schedule and budget in order to plug that information into the higher level NPI project management process.

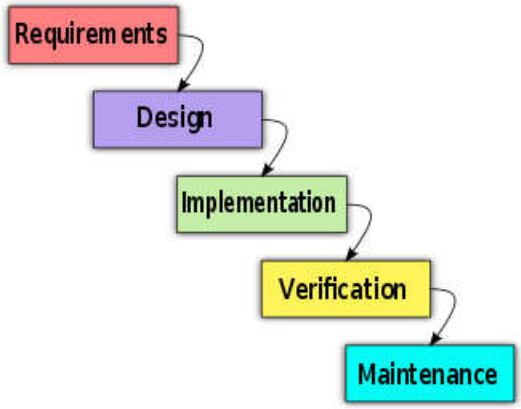

Figure 1 – Waterfall Development

One can see the conundrum of this scenario. How can the T&M system requirements be defined and well known if even the DUT requirements themselves are still in a high state of flux?

What ends up happening is that the organization will follow its usual waterfall model by using the information that is available about the DUT at the time of proposal generation. It will use technical assumptions to make sure it is contractually covered in the event of changes to the DUT. The project budget is set aside for that portion of the NPI program, and things move full throttle ahead.

Well, changes to the DUT not only are expected, but are certain to occur as the product development process is still underway. A project being executed under this scenario needs to be managed extremely well. However, one first issue that will most likely happen is the fact that the original budget that was put in place for the T&M system will most likely not be accurate due to the fluidity of the product development. Depending on the depth of the changes in the DUT, the T&M system may need to change significantly to match the new revision of the DUT. This always carries a cost and a schedule impact.

Using the last sentence as a hook to the next argument, the original implementation schedule that was included in the proposal will most likely not hold, for the very reasons that were presented above. Now the expectations need to be realigned, not only within the test department, but throughout several other departments in the organization that were assuming the T&M system original schedule as Gospel. This communication needs to be very tactfully managed.

One other fact that is very often overlooked is that, usually, there is a lot of value in having the T&M system partially delivered to the client and in a configuration where the product development team can use it to execute characterization exercises. The T&M system has instrumentation that usually can used by the R&D department to collect data that will help in making design decisions about the device under test.

If a pure waterfall development methodology is used, the test and measurement project manager will most likely object to having these intermediate deliverables as they can be very disturbing to the actual flow of development of the T&M system. They most likely will add to the project costs and will create an impact on the overall deployment schedule.

The following figure summarizes the consequences of this challenge.

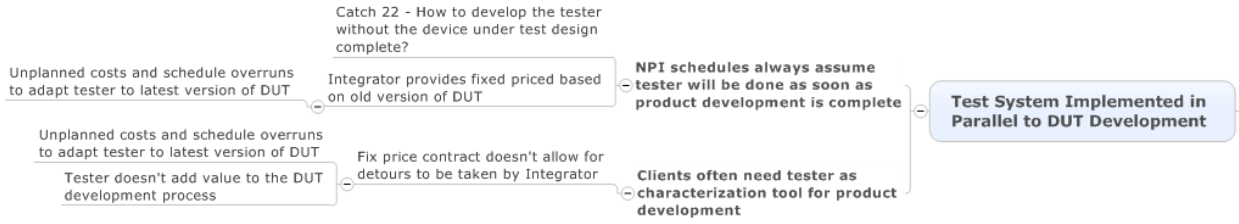

Figure 2 – Issues with Test Being Developed in Parallel to DUT Development

The Agile Methodology

In 1970, Dr. Winston Royce presented a paper entitled “Managing the Development of Large Software Systems,” which criticized sequential development. He asserted that software should not be developed like an automobile on an assembly line, in which each piece is added in sequential phases. In such sequential phases, every phase of the project must be completed before the next phase can begin. Dr. Royce recommended against the phase based approach in which developers first gather all of a project’s requirements, then complete all of its architecture and design, then write all of the code, and so on.

The waterfall methodology assumes that every requirement of the project can be identified before any design or implementation occurs. This clearly goes against the very foundation of a NPI process where the T&M project needs to start way before the unit under test development is completed; making impossible for the project requirements to be defined beforehand.

Agile development methodology provides opportunities to assess the direction of a project throughout the development lifecycle. This is achieved through regular cadences of work, known as sprints or iterations, at the end of which teams must present a potentially shippable product increment. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as “iterative” and “incremental.” In waterfall, development teams only have one chance to get each aspect of a project right. In an agile

paradigm, every aspect of development — requirements, design, etc. — is continually revisited throughout the lifecycle. When a team stops and re-evaluates the direction of a project every two weeks or any other short interval that makes sense in the context of the project, there's always time to steer it in another direction.

This "inspect-and-adapt" approach to development is a perfect match to the fluidity of the NPI cycle. Once the requirements to be implemented by each sprint is aligned with the incremental progress that is made in the device under test itself, progress can be made on the T&M project on areas that have a lower risk of being changed due to a change in the device under test being developed. Since each sprint is of a short duration, it gives stakeholders recurring opportunities to calibrate releases for alignment with changes in the device under test development. This allows for a much leaner implementation of the T&M system, reduction of waste and rework due to unit under test changes and consequently maximization of the business value for the test system.

Scrum is the most popular way of applying the Agile methodology on a project. The responsibilities of the traditional Project Manager role are split up amongst roles (any similarity with what the T&M framework proposes for its organizational structure is not mere coincidence). The main roles defined under scrum are: the Product Owner, the scrum Master and the project team.

The project team takes a much more active role as far as accountability for the project success. The nature of scrum is that the teams are self organizing. What that means is that since the duration of a sprint is short, the number of requirements to be implemented is much lower than the total number of requirements for the entire project. With that, each team member has better visibility of where the finish line for that sprint is; which allows them to make more independent decisions as to the tasks that need to be accomplish and their priorities.

The scrum master is a facilitator. Her main task is to remove the impediments to the ability of the team to complete the spring deliverables by the end of the sprint.

The role of highest interest for the discussion of this chapter is the Product Owner. The product owner represents the voice of the customer, or in our T&M project case, the voice of the stakeholders. Her main task it to ensure the value to the business is being delivered by the project. The product owner is the one that writes what is called "user stories", which can be seen as features to be implemented, rank and prioritize them, and add them to the "product backlog".

The product backlog is an order list of requirements that is maintained for the project. This can be seen as the overall project requirements. The main idea behind scrum is that a number of these items from the product backlog, the ones with higher priority numbers, are selected to be worked on by the project team in the course of the next sprint. Any requirement that doesn't make the cut for the next sprint is seen as an out of scope item and is not touched at all.

Now, going back to our NPI scenario, one can infer that a structure like the one mentioned above can be extremely advantageous to the project. The initial requirements list, or, to use the Agile terminology, the product backlog can assume the test system will be built by taking the current state of the device under test as if it has been already completed. The product owner would them prioritize and rank the product backlog in a way that the stable features would be addressed first.

At the end of each sprint, the product owner would then audit the current state of the device under test, speak with the product development stakeholders about any possible upcoming changes and collect the overall direction of the product development effort. This information would then be used for another exercise in re-prioritiznig/re-ranking the product backlog remaining items for the requirements of the next sprint to be formed.

This practice may not fully eliminate rework altogether since, depending on the fluidity of the particular NPI cycle, there still may be cases where the T&M project team is asked to take a step back. However, this will certainly increase the odds of minimization of these step backs along the life of the project. This result not only increases the odds of overall project success around business value, cost and schedule, but also brings another very important byproduct. It reduces the overall frustration of all people involved in the project:

- The stakeholders will see tangible progress being made, sprint after sprint, towards the final test system to be delivered
- The project team members will probably not come across the “users don’t know what they want” syndrome as heavily
- The initial project budget and schedule, if defined with the Agile mentality, is probably maintained to a much closer level of accuracy than in the usual waterfall method of implementation for both client and Integrator

Modified Agile for Test and Measurements

Let’s go back to the product backlog for a moment for a more detailed account on what it is and how it is created. The product backlog is a prioritized features list, containing short descriptions of all functionality desired in the product, or deliverables. One of the most predominant Agile’s tenets is that it is not necessary to start a project with a lengthy account for the project requirements. In fact, the process of creating an Agile product backlog is by having the project team and product owner writing down everything they can think of for the product feature set and prioritization.

The product backlog usually changes and grows as more is learned about the project. Here one can see the first potential problem with applying a purist Agile approach to a T&M project, especially if they are being executed by an Integrator. As it was mentioned previously on this book; both clients and Integrators like to fix price T&M projects. Clients like the certainty of

project schedule and budget determined upfront so these numbers can be plugged into the overall NPI. Integrators like fix price when the scope is defined, so they can leverage their own internal tools and efficiency of people who do these types of projects for a living.

The very definition of the creation of an Agile product backlog is a showstopper for a fix price engagement with an Integrator. No Integrator would ever fix price a project based on a very lose scope of work, defined by an initial product backlog where no rigor was applied in the definition of some level of project requirements that can be translated into a defined scope of work.

Another point that is worth making in this initial discussion is that Agile product backlogs usually are created through the utilization of the so called user story. A user story in Agile is very often wrongly confused with use cases (refer to the article named "Test and Measurements System Modeling: Addressing the Root of the Problem" for a detailed explanation not only on use cases but also on other UML diagrams utilized in the modified Agile approach for test and measurement projects).

A user story is a very short description of something that a stakeholder will do when they use the system, focused on the value or result they will get from doing it. They are written from the point of view of a person using the system, and written in a language that the stakeholder would use.

A use case is a description of a set of interactions between a system and one or more actors, where an actor can be a person or other systems, as it was defined by last chapter. Use cases are made for requirements to be elicited; whereas user stories are made so the initial, loose, product backlog that will kick off the Agile process.

With this in mind, it is somewhat easy to see why the pure Agile product backlog is not the best way to develop a T&M project that will support a NPI and will be done by an Integrator.

However, as it was mentioned, user stories are often confused with use cases. The fact of the matter is that use cases can almost be seen as a deeper level of abstraction than user stories. User stories are meant to be superficial and simple. Use cases are meant to be descriptive and all encompassing.

When one adds activity diagrams to further details the use cases, the initial project requirements can be very closely modeled by just these two diagram types. The whole idea here is the creation of a modified Agile product backlog. This modified version of the product backlog will include the use cases and the requirements detailed by the activity diagrams, as opposed to superficial user stories.

There will be a translation process of information coming from the diagrams into a list of tasks, features and activities; which will be our modified Agile product backlog. Once this list is

completed, as it was mentioned on the previous chapter, the list is then prioritized, ranked and split into multiple sprints of defined scope.

What this article proposes is that, at that point, a process close to Agile in the standpoint that sprints will be executed and potential changes to the original product backlog may come about, due to the fluidity of the NPI process itself. It is being stated here that the process is close to Agile and not Agile due to the fact that the product team has much less flexibility to come up with their own changes within the scope of a sprint. There will still be a structure close to non-Agile projects for the scope defined by a sprint to be implemented. However, the nature of this fine granularity breakdown of a large task into small ones that can influence changes on the following upcoming ones is very much in the heart of the Agile movement.

Once the initial product backlog has been created and the sprints identified with their corresponding scopes from the product backlog, the Integrator can fix price each of them separately, as if they were separate projects altogether. This will give the client an initial idea of project budget and schedule that can still be plugged into the NPI master schedule and budget. However, the client needs to understand that this initial project schedule and budget will most likely change due to the fluid nature of the NPI process. This provides the clients with several levers that can be adjusted based on the overall parameters of the NPI

For instance, NPIs that are not extremely aggressive as far as final product launch can afford waiting longer for the T&M project to start. These types of NPIs will obviously benefit from a more stable device under test when the T&M project is kicked off. This extra level of stability will lead to a more accurate initial product backlog. This more accurate product backlog will lead to a initial project schedule and budget that will most likely have less variation than one where the NPI launch schedule is very aggressive.

On these types of NPIs the T&M projects absolutely need to be completed at a very near point in time to the completion of the device under test. For those, the initial product backlog will be as unstable as the device under test is at that point in time. This will lead to more variability of project schedule and budget that needs to be taken into account by the NPI management team.

Conclusion

This article presented the main issues with the typical frameworks of execution of test and measurements projects. It explained why a waterfall based approach doesn’t work well under this circumstance.

It introduced the concept of Agile methodology and went on to explain how a modified Agile approach is ideal to execute test and measurements systems that are in support of NPIs. The modified Agile methodology uses system modeling, which is a strong feature of the TMPM framework, a framework that is tailored for the execution of test and measurements projects. The user can refer to the article named "The TMPM Framework: Tailored Project Execution for Test and Measurements" for more details on the framework.