

## Abstract

*In this paper I have selected a few of the key considerations that need to be in place in order to set yourself up to succeed with Agile.*

## Introduction

Having worked with number of agile teams as a Scrum Master, Product Owner and coach for both internal and client teams, I have noted some common challenges for Testers. I thought that it would be worthwhile to share some of these with you so that you would hopefully not need to share the pain.

## Change the mindset

The first, and I think the hardest in most instances, is to change the mindset of the tester. There are distinct patterns of behaviour that are exhibited, often split by the seniority of the tester. Let me try to explain what I mean for those of you that have not seen this happen as drastically as me, for those that have, you will know what I mean.

First the more senior testers: they are probably more entrenched in the traditional methods. Seventy page Test Strategies and/or Plan which take weeks and weeks to prepare. A work breakdown structure probably in Microsoft project which has hundreds of dependencies, duplicated in an excel spreadsheets or some Test Management tool lab for at least the test execution, with names against each test. Do you get the picture? Or are you seeing yourself? Frighteningly this is how they think you plan. Once the Strategy / Plan is signed off, off you go ( or in worst come cases this is only signed off just prior to production when it is completely out of date, as it has not maintained ). Apologies for the extreme picture, but this is one I see all too often. In summary in traditional environments they have been taught that big up front plans is the only way to work so therefore the only thing they know. Agile is different as it relies on "just in Time Planning". It has to be said that I have seen good Test managers in traditional projects doing continuous planning reacting to the risks identified or residual risks, however this is rather and they probably do not update their Test plan only the project plan.

The first thing they find difficult is that pages and pages of plans are not required. A one page or couple of slides are good enough. The more important aspects of planning are to have the ongoing discussions about the stories (the requirements regardless of how they are recorded) as they are picked off the backlog and elaborated to understand better. The plan is a living artefact and the activity planning is ongoing as details become clear, risks are understood and impediments are highlighted and hopefully resolved. This is not just the remit of the one individual, the whole team has responsibility to contribute based on their level of knowledge and expertise. So now the new paradigm is to learn how to do "Just in time" planning. Key point here is the whole team concept and that measure (KPI's) are now team based not individual centric. Again a different paradigm.

I have also observed that they find it difficult to let go of the management and control activities and what it means to do them, i.e. its power and control, or more bluntly a lack of trust that the team can do the right thing without supervision. The Agile team needs to engender trust in the environment therefore not requiring the control. You see them going around asking the other people conducting testing. "Where are you up to with that test", "Do you think you will have finished all you need to today?" It is a very hard

habit to break. This can have a very negative impact on the team if this is not stopped, as it creates an atmosphere where you do not trust the team to get the testing complete. It also means that if you are running around managing others then you are not fully contributing to the team delivery or getting your tasks complete. There is a very fine line between monitoring and coaching, sharing your experience and knowledge and directing. It is important that the members of the team are self managing and allowed to fail in order to learn. Many people like the term "manager" in their title and to be told they are now part of the team seems like a demolition to them. It is often about status and prestige or impressing others instead of an egalitarian approach to one succeeding we all succeed!

The above is one of the roles to consider when forming the team, but there are also the more junior testers. Being told that you have to plan your work, take tasks as you finish your piece of work and make a decision on when testing is complete are very daunting when you are used to getting directions from your Test Manager/ Lead. Often this can be helped by some coaching or mentoring from a more senior member of staff. Once they have build up the confidence and having made some decisions they will grow. Engendering an environment where there is a no blame culture; the team takes collective responsibility of decision both good and bad and where failures are not punished, but seen as area for improvement and learning will also help. It is important for all members of the team to feel that there is open collaboration and they feel "safe" discussing areas that they are not sure about or comfortable with, as with all learnings this may make early iterations a bit slower, but they will speed up and this early investment will pay dividends. As with all learning, like riding a bicycle, the first few times you will be wobbly, slow and may fall off, but if you persevere you will get faster and more proficient.

One of the most critical aspects with regard to testers getting the right mindset, if agile is fresh to them or even if they have worked in an Agile environment before where practices were not so robust, is to acknowledge that the first few iterations are likely to have lower velocity even disregarding the "getting to know and learn to work with each other aspects".

Is this about forming a team, or more about the building blocks you're using to create it? Remember the "forming, norming storming" team building approach. This is about creating team cultures from a group of different "individual" building blocks, in which all the blocks may not fit. Those with entrenched behaviours, Inflexibility to change or trying new things will not succeed, so they blocks need to be removed and placed into teams in which their abilities will enable them to be successful. I'm writing a blog post on this which I'll send you for review.

## Documentation

The next challenge, is the quality of the input documentation for testing. Again I can split this into two categories for discussion: too much and too little (or not at all).

When I say too much what I mean is that there has been too much upfront requirements elicitation resulting in documents resembling the mammoth toms of traditional projects (I hear the groins from some of you and the cries of "that is not agile" but that is often what you have to deal with in the real world). Yes, through education, training and coaching we can make it better for next time, but we have to deal with this here and now.

With this one I would get the teams together and try to 'chunk' the document up preferably in some priority order if the requirements have been so annotated or if not with the product owner. The team

must have smaller pieces to work with or they are not going to start delivering product. Chances are that once they start the requirements are going to change anyway so get started.

Another approach is to ask the product owner, what are the key risk areas that they would like to be priority, i.e. is going to give them the biggest Business value. Then, just look at those requirements in the document as a starting point. Repeating this exercise at the start of each iteration during the planning stage. So the team is only focusing in on "just in time" requirements. Eventually the original mammoth document will become obsolete, and the continuous rapid feedback loop will generate the requirements.

The other extreme is too little or no documentation. This is not an issue when you are working in highly collaborative environment where you have almost constant access to the product owner, or there is a low audit or legislative requirement. Very few of us would be able to tick all of those boxes, may be a couple of them. This is one of the most frequent questions I get asked and always results in the most debate. Unfortunately, the answer in most cases is 'it depends', which is not a satisfactory answer for most people and I have to agree with them on this point. The "it depends" answer is due to the fact that it needs to be answered based on the context in which you find yourself. It is really a sliding scale, dependent on the environment in which you work, as to how far along the scale you are. This may be different from company to company or even project to project within a company.

The simple fact is that you do not necessarily need lots of documentation to "communicate" what the requirements are, but you certainly do need to "understand" what the requirements are describing or explaining. With any requirements, one of the most critical aspects is that the whole team has the same understanding of what those requirements are and it aligns to what the business will get the most value from. Everyone in the project must have the same common goal and objective.



## Stakeholder involvement

The last challenge with moving a team to an agile approach is your stakeholder's involvement and buy in. Without this the testing is doomed, as well as the project as a whole, to be frank. So many times I speak with our clients, when they have decided to 'go Agile' and within 5 minutes of the initial conversation you know it is going to be an uphill, rocky road if not a complete disaster before they even start. There are a few questions that I ask up front to try to gauge their real commitment to Agile and the levels of support that the teams are likely to be able to get.

Here are a couple of sample questions that I ask and then watch for their reactions:

1. "Who is the product Owner?"
  - When they say we have multiple product owners, follow with "who is going to make the final decision then, as to the priority in the product backlog?" If this question is not resolved, it results in the team not knowing what to work on first.

2. How much time does the Product Owner (or their representative) have to spend with the team?
  - If the answer is they will not have much time to spend with the team then this may slow down velocity as well as the team delivering the wrong functionality. It is also likely that defects may not be found before production as testers do not understand how the system will be used.
3. Have Audit and /or Legal been made aware of the process change and all the implications, including but not least the reduction in project assets?
  - I normally get a huge look of dismay at this point. It is important that these stakeholders are brought on board with Agile as early as possible and understand what deliverable they are going to see (or not if this is the case). The stakeholders that are not part of the product development team are often not engaged, or even thought of as stakeholders.

## Conclusion

In order to give the team the best opportunity to succeed there are few key elements that need to be put in place to not only enable a software tester to be successful, but the team in general. Some of these elements need to be in place before starting the Agile project or very close to the beginning, such as during iteration 0. Others elements required will be ongoing learning and development opportunities.

It is important that all these key elements are recognised and managed appropriately both before and during the implementation of Agile by having an approach in place to mitigate these challenges when they arise. An approach that leverages coaching or mentoring is frequently the best mitigation strategy for effective management of these challenges, but conversely, if you are already working in an Agile project, have you considered all of these elements. If not, it's not too late!

## About the Author



Leanne Howard is an Account Director with Planit. Over the past 20 years Leanne has worked in the IT Software Testing industry across multiple projects including those using Agile. She is a specialist in the implementation of practical testing methods with a strong focus on client satisfaction. She enjoys sharing her knowledge through mentoring and coaching others, and is active in the testing profession.