

## Abstract

*One methodology, process, or practice does not fit all, adapting to the environment in which you find yourself is what is going to make your testing a success. A mix and match of practices is the best.*

## Introduction

I have been reading a lot recently about people getting up in arms about the various Agile frameworks. I have to caveat what I am about to say with the fact that I take a very practical view to Testing and do not have a strong preference for any particular one. I also have a view that if some of the practices from Waterfall, V-model or any of the other methodologies or processes works for you, in the situation in which you find yourself, then use them. The old adage of "IT DEPENDS" I think applies in most instances. Who knows the details of the environment in which you find yourself better than you do?

As you have probably worked out, I am a "who cares?" kind of gal, always willing to buck the system if it proves to be a better way forward. What works for me, works for me. Having said that, please don't think that I believe that one thing fits all, nor am I afraid to put my hand up and say I don't know or I don't understand. In my view, everyone should be learning something new every day if possible. If you don't try it, you won't know if it works. Also, if it worked well in one situation, that does not mean that it will work well for all projects or, come to that, work well for a similar project again. There are always some variables that change. I have found the risks or issues which, on first sight seem inconsequential, probably are the ones that you should watch as the big ones, after all, the whole project team is watching.

## People over Process

Now I have set the scene, back to Agile specifically. I find it ironic that one of the main principles of Agile is all about collaboration and we find ourselves in the Agile community arguing over which framework is best and if you are not doing all the elements, then you are technically not following it.

Again is it "People over Process" if the people want to use part of a process from here and another from somewhere else, why shouldn't they? At the end of the day, the team is supposed to be self-managing, so why shouldn't they choose what is best to suit them? This is an adaptive process, so if the team finds that one of the elements is not working for them, in say their retrospective, then why not stop doing it or try something else? That is not to say that I advocate chopping and changing for change's sake. Give each process a fair go. In my view, sticking to one process just because it says so in the book or an Agile 'guru' has said do it that way is just not right.

## Strong Processes

Having re-read the above, it almost seems like I do not follow any process at all. Not true, as anyone who has worked with me will attest. In fact, the opposite is true. I am a great follower of process, but the right process or, should I say more accurately, a blend of processes that fit the situation in which I find myself. Often process is what saves me as I frequently get the call when a project has gone into crisis and they want me to come in and wave my magic wand and make it all better.

I can tell you that Agile or Lean processes have saved my bacon on these occasions. In fact, this is where I really got introduced to Agile. In these situations, I really have to think on my feet and hit the ground running. I am also involved in a lot of reviews with companies around their Test Processes and interacting disciplines. If I come into work on a struggling project, I will often hold a retrospective within the first couple of days, even if it is not an Agile project. The team will tell me what has gone well (it is important to find some good even in a disaster project, as they have been living in it often for months) and what has not gone so well and then their suggestions for improvement. I have found that, if you get their buy-in early, you probably have it for the rest of the project. Also, who better to know what is going wrong than the team?

## Stand-ups

So what else works for me? Well, I always have a daily standup meeting in every project I run, even if we have been using the Waterfall model. In fact, I even have a daily standup for the staff that may be on bench (between project assignments) back in the office. I have found that this is a fantastic way to promote collaboration and team self-management. If team members report that they are blocked either during test preparation, waiting on some feedback or doing execution with defects, they will offer to help other team members who are falling behind schedule for whatever reason. These meetings do not need to be long and I try to keep them to 15 minutes even if I am managing a large team.

## Taskboard



Another fantastic idea borrowed from Agile (or was I doing it before?) is the task board or a slight variation on that theme, maybe some would call it a dashboard. Again, for all projects I have a whiteboard as close to the team as possible which gives a quick visual representation of where we are. This not only gives a clear and simple view of what is happening on the team but it also promotes collaboration within the team and self-management.

As is often the case, I am sure Senior Test Managers find themselves in endless meetings, the team members can clearly see if they have test cases blocked by defects where there are other areas that they can help their team. They will go and offer their assistance thereby helping the testers who have lots of work to complete. By the time I get back to the team the issue is

resolved between them with no management required from me. This also promotes natural cross-training, making team members more multi-skilled generalists rather than specialists in a certain area of the systems under test.

## Pairing



I also find pairing a very useful technique from my tester's tool kit. Where possible I advocate pair testing. Two testers get together during the preparation and they are given responsibility for a number of areas. This helps with cross skilling and usually the generation of the right coverage of the test cases through the collaboration and bouncing ideas off each other. More than one set of eyes looking at a problem plus discussions help in this. This is also a good technique to pair testers with Business persons so they have a better understanding of the value of the end product or with developers to have a in-depth understanding of the technology and its associated risks.

This also gives an additional benefit in the execution stage, where we all know that time is often tight. I now have greater flexibility to move team members around if an area is blocked, as they are familiar with number of pieces of functionality. If I am having issues with the turnaround time of defects or continually having defects circle around between development and test, then I will often pair the testers and developers to work together to solve the problem and test it within the development environment, before it goes through the often time consuming release process, particularly for the highly complex systems I work with. Not only does this forge closer relationships between the teams, breaking down the them and us barrier but it is surprising how many failures get asked and responded to much more quicker in this collaborative environment. Once the developers see us pairing successfully this often spreads to their team if they are not doing it already.

## Mindset

As with most Agile practices, once the team see the merits you will often get the greater benefits. I have found that introducing process slowly has worked well within the teams that I have managed. Often it is mindset of the people which is hardest to change, particularly if they have come from a traditional

background. They know (for the more junior members of the team) that they always have more senior person, the team lead or test manager, who will ultimately make the decision. This is not the case in Agile where the team is empowered to make the decision. These soft skills are the ones that companies often forget to support as they move to Agile, which are key to success.

If you can engender some of these soft skill traits from Agile into your traditional teams it will make life easier. As I have said, collaboration and self-management are key. Identify key individuals within your team to give more responsibility too. Often if the team sees others 'step up', they will follow. It is vitally important that you also support your team if they fail, and allow them the time to grow.

## Conclusion

In conclusion, there are some obvious variations between the different Agile frameworks, as with the traditional ones but there are also some crossovers. Agile is an adaptive method advocating continuous improvement and rapid feedback loops so why not blend the practices between Agile methodologies? As far I am concerned, what works well in your particular circumstances is what you should use, never mind whether it is part of Scrum or XP or any of the others. What's in a name? It is whether your team is successful, including the enhancement of your team's skills and capabilities, as well as their well-being, that is important.

## 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 after working as a Financial Executive. She is a specialist in the implementation of practical testing methods with a strong focus on client satisfaction.