Change Management is core to Agile

Author

Melanie Franklin

Director

Agile Change Management Limited

Contents

Executive Summary

As Agile adoption grows, it needs to mature. For me, this maturity involves integrating the advantages of Agile with other commonly used management approaches. Specifically, in this paper I address the integration of Agile with Change Management and the implications this has on roles, responsibilities and processes.

This integration is important because Agile continues to be used by many organisations as a delivery mechanism but delivery without the ability to utilise what has been delivered fails to realise benefits.

We need to involve a wider community than just the Agile team and we need to help everyone understand what it takes to move something from existing to being used and making a difference.

Agile Overview

Skip this section is you are an experienced Agilist and go straight to the argument for integration of Agile and Change Management. For those of you new to Agile this is a quick summary of how it works.

Agile is an approach for getting things done. An idea is captured as a series of User Stories which explain what needs to happen and why it is useful for this to happen. These User Stories are collected into a list of all the work called a Product Backlog.

To get the work done, these User Stories are prioritised so that the team work on the most important work first. This work is divided up into short amounts of time, known as Sprints or Timeboxes, often lasting about two weeks. In this time, the whole team works together to get something workable delivered at the end of the Sprint.

Sprint ⟩ Output Sprint ⟩ Output Sprint ⟩ Output Sprint ⟩ Output

These deliverables are then either ready for use or they are used as an input to the next thing that has to be created. Either way, something tangible has been created that generates feedback, hopefully from the end customer as that is the audience we are trying to satisfy. The feedback generates ideas for changes to the deliverables and identifies more deliverables, so more User Stories are written and added to the Product Backlog. This continual adding of new ideas is why everyone talks about Agile as being flexible and responsive to change. So the cycle keeps going, with the next Sprint creating the next deliverables, ready for use.

Although Agile began with software development, it is a way of turning any work into a production line, so there is emphasis on achievement, delivery and efficiency of work. At the end of every Sprint there is a Retrospective, where the team review how they did and work out how they can do

better next time by tweaking some aspect of their production line to make it go faster and to ensure it delivers less errors. Although a lot of the work to which Agile is applied is cognitive, the principles of lean manufacturing and continuous improvement are key to its success.

Why Agile needs Change Management

A faster and faster production line creates more change, and if we do not ensure that the recipients are ready to adapt to these changes, they just stack up, creating a queue of deliverables that have been paid for but are not being used, so benefits are not being realised.

Change Management is not an 'add-on', not something that we might get round to if we have a bit of spare time! Change Management is the collection of activities and techniques that minimise the risk of failing to realise benefits. As a risk management technique it is an essential partner to Agile, which tries to realise benefits as early and as frequently as possible, ensuring the maximum return on investment from each initiative.

Agile without Change Management is harmful to an organisation in several ways:

- It is the benefits that pay for the creation of the deliverables. If we keep the production line going without changing the business and getting the additional revenue or cost savings to pay back what it cost us to create the deliverables, eventually the Agile production line is going to run out of resources. The production line costs money to run, and can only keep running if the benefits exist to fund it.
- It is demoralising to work faster and faster but know that what you are creating is not being used. Agile teams are not dumb, they know when their outputs are being ignored by the business. In some organisations this is creating a negative atmosphere where those in Agile teams are angry with what they see as the lack of effort from their internal customers.
- On the flipside, it is demoralising to work in an environment which is subject to continuous change with no time to think through the implications of each change, or draw breath between making these changes. So this also creates a negative environment, as 'the business' or 'the internal customers' feel under pressure to keep changing how they work at an ever faster pace to keep up with the deliverables from the Agile teams.

Create a balance between 'creating' and 'implementing'

To address these problems we need to recognise the value of two complementary areas of work and treat them as equally important:
- Creation – designing, developing and testing tangible deliverables
- Implementation – designing, developing and trying out new ways of working that make use of the new tangible deliverables

Initially it may appear as a doubling of the workload, but implementation activities must always happen. In many initiatives, they are a hidden cost, carried out after the Agile team have finished their work, resourced by those with 'business as usual' responsibilities. As this effort is not

recognised as part of the Agile initiative, the Agile production line runs faster than it would if implementation were included. This increases the problem of deliverables queuing ready for use.

If change management activities are identified as essential pieces of work in every Product Backlog and are included as User Stories in every Sprint then the resources and time required for the implementation activities are not an afterthought and can be considered up front when the team are formed and the Product Backlog is created.

To achieve this recognition of the need for implementation activities, I think we must:

1. Identify the implementation activities from the right source
2. Reconceive the roles and responsibilities of Agile team members to define what they need to do to help implement what they create.
3. Explicitly define what change management activities are so that there is real understanding of what is needed, and change management is seen as 'just another piece of work' and not a specialist subject that can only be done by a select few.

Identify the implementation activities from the right source

Every team member needs to ask themselves 'what do I need to do to make sure that what I create will be used by the business?' To know who is the 'business' individuals must view their role in a much wider context than 'team member' or even as part of a department. Instead they need to appreciate how their work contributes to how their organisation services its customers. In some cases this means the external customer, who pays for the product or service:

Deliverables from the Agile team → External Customer
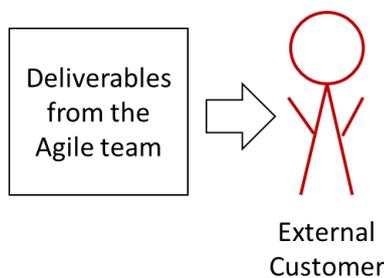
Sometimes there are so many internal customers in the value chain, it is more useful to recognise each of these and how work from one Agile team is used as inputs to work further down the chain.

Deliverables from the Agile team → Internal 'customer' → Internal 'customer' → External Customer
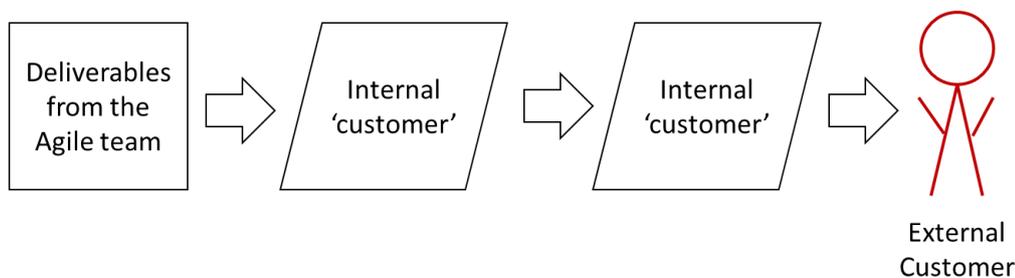
The team need to agree who their customer is, otherwise they will not understand how the customer will have to change what they do to make use of the deliverables created by the team, which means the team will be addressing the wrong implementation activities.

Reconceive the roles

Change activities reshape the existing business as usual, to create new ways of working which many in Agile teams do not see as part of their remit, nor do they feel they have the authority to get involved. I definitely have some sympathy for this view, because the siloed nature of many organisations means that those creating change are often not the ones who are going to live with it in the longer term.

The obvious approach is to partner those impacted by change with those creating it. However, many Agile approaches have missed this opportunity, because the voice of the customer is concentrated in one role, that of Product Owner. In part the name describes the problem – owner of the product, not the processes which source the inputs, run the process or send the outputs. That is why so many requirements or User Stories are about how something works, but not how it is used. We need to include ideas about how to use the products that Agile create if we are to get value from their existence.

Some of my Agile teams have developed this idea, making room for a Change Manager to be part of every Agile team. This role specifically creates the change related User Stories, through debate with Business Analysts and those in the business who will have to work differently. These User Stories are included in and prioritised in exactly the same way as every other idea in the Product Backlog.

Others are starting to widen the remit of Product Owners to include changes to ways of working and the activities needed to realise the benefits of what is created.

I think we can strengthen this approach by including change into the Agile concept of 'test driven development'. What this means is that before any development is undertaken, everyone involved in the creation understands what the acceptance criteria are. It is a simple yet effective idea that cuts re-work and false starts because if you know what good looks like at the start, you are more likely to achieve it at the end, because you will keep comparing your work to these criteria as you go. At anytime you think that your work is not going to meet these criteria, then you change what you are doing to get it right, which ultimately leads to an increase in the proportion of things that are developed 'right first time'.

Testers are essential to this approach, because they articulate up front what is required. By including what changes to current ways of working are required in the business to make use of the new deliverables and realise benefits, Testers can ensure change management is on the agenda of all work from the start.

I can hear some of you arguing that Business Analysts are very good at working with the business to identify all of these change activities. I don't disagree, I think Business Analysts have a vital role

to play in ensuring the business re-design their ways of working to take account of the new deliverables. My point is that this is a team effort, and it needs more than one role to make it happen, so the responsibility for implementation activities must be baked into the role description of every Agile role.

Key to all of this is the role of Scrum Master and Project Manager. Both of these roles have a responsibility for ensuring that teams are using Agile in the most effective way, and that nothing blocks the progress of an Agile team. Already some clever organisations have recognised the need to widen the definition of Agile from creation to creation and implementation and this is being reflected in Scrum Master and Project Manager job descriptions.

These are some of the responsibilities that forward-thinking organisations are including in their role descriptions for Scrum Masters:

"Work to understand business challenges and ensure these are reflected in User Stories."

"Strong customer service orientation and willingness to empathise with our customers about the impact of our work."

"Engaging with users and customers to maximise the benefits we can deliver to them."

"The management of end to end change assignments including reporting on benefits."

"Delivering business change that is able to adapt to changing customer needs, driving business value and great customer outcomes."

Explicitly define change management activities

We need to explain in simple terms what we mean by change management activities. If we are to achieve parity between creating and implementing, then implementing needs to be understood. After all, who is going to take responsibility for doing things they don't understand?

As with everything in the 21st century, if it doesn't have a brand name, there is a good chance what I am saying will get ignored. So my brand name for the work I am describing is BusOps. This pays homage to DevOps which is the IT aspect of deployment, which seeks to combine development with operations (hence the name) to automate the technical deployment of new code. It is the IT response to the pressure of continuous Sprints of development, to make sure no queue of completed works builds up because putting it into live use is too slow.

We need the same approach to deploying deliverables into new ways of working, making the updates of how people do their work as continuous as the creation of deliverables from each Sprint. Unfortunately, there is a key difference which makes 'BusOps' much more difficult than DevOps – people. Human beings cannot be re-programmed as easily as systems. We take time to work through our emotions when faced with change. Some of us are faster than others but all of us are slower than machines in adapting to change.

This is such a complex problem that we need a range of strategies to deal with it:

1. Capture all change activities into a Checklist
2. BusOps needs a long lead time
3. We need the involvement of lots of different people

Capture all change activities into a Checklist

To keep things simple, we should be explicit about what needs to change, by asking the same questions every time something is created. Making change happen is a complicated business, but if we fall into the trap of re-inventing the activities every time we create something it makes it harder to view change as a standard activity.

Create a list of all the questions to ask so that the answers can be brainstormed and the full scale of what is involved can be planned, resourced and scheduled. An example of these questions are:

- Are we changing the inputs to any of our processes? This includes the data we make available to systems, the format of this data, the time when it is available etc
- Do we need to change our operational processes? This includes removing steps from the process because we have automated the work, or adding in new quality reviews or sign offs.
- Do we need to change our maintenance processes? This includes contracts for support, timetabling updates etc
- Do we need to change our security processes? This includes who can access systems and information and what information can be changed or shared.
- Have we told anyone in supervisory, decision making or audit roles within our organisation about our new processes?
- Have we designed any training for staff?
- Have we re-designed their roles to take account of the changes in the processes?
- Have we designed new KPIs to align to the changes in ways of working?
- Have we notified our customers about different processes, different contact points or a shift in the balance between what our organisation does for them and what we expect the customer to do for themselves?
- Have we notified suppliers about any changes in our assumptions in what we expect from them?
- Have we put in place actions to review the Service Level Agreements or contracts we have with them?

Turn the answers from this checklist into User Stories. For example:

As a Customer Services Manager
I need developers to show me the changes to the data input screens
So that I can identify what steps to remove from the existing process

As a Call Centre operative
I need to know what the new information we need from customers
So that I can change what I say on the phone to customers

By having these User Stories ready for inclusion in any Product Backlog, the implementation activities can be prioritised alongside the creation work from the start of the initiative.

BusOps needs a long lead time

If we are to change how we do things, we need time to adjust to how that feels. Some of this time might be taken up with anger about the unfairness of having to learn new skills and the inconvenience of leaving behind perfectly good processes to move to something untried and untested. Some of this time is needed to try and argue our way out of changing, and when this doesn't work, maybe some time is required for sulking before we accept the inevitable!

It makes sense to raise the question of 'what needs to change' as early as possible. All team members can get involved with this, by holding early demonstrations to users and customers about what might be created. I know many teams already do this, but I am suggesting a change in emphasis. Instead of debating functionality, debate the impact of the proposed deliverables on current ways of working. Ask users and customers to discuss the list of questions I outlined in the Checklist section of this paper, and help them to define relevant User Stories.

We need the involvement of lots of different people

To really understand the impact of what needs to change, and to be able to define these changes in detail we need to ask those who will have to change what they do. This means that any Agile team needs to place itself within the community of those impacted by what it creates, not just the 'end user'.

A team I am working with have had a lot of success by working out who governs their users, and making sure these teams are included when they identify the implementation activities. This means they are now working with Internal Audit, the in-house legal team and the Compliance department who manage the relationship with external regulators.

Another team have used the SIPOC technique to create a picture of the Suppliers, Inputs, Processes, Outputs and Customers relevant to their work, extending the Suppliers and Customers questions to include 'who supplies our Suppliers?' and 'who are our Customers supplying to – are they part of a wider value chain?' This has more than doubled the number of viewpoints that they need to consider, but it has enabled them to understand at a much deeper level the upstream and downstream impacts of their work.

Conclusion

Agile needs to mature as it continues its path to global domination of the way we manage work! If we are to widen its appeal, we must ensure that it does not become a mechanism for throwing increasing volumes of work at users and customers. Just because we have the capability to create at high speed, we shouldn't assume that those affected can keep up. After all, how many of us, regularly logging into our favourite apps get frustrated that overnight a feature we liked and used regularly has been replaced and we now need to learn new shortcuts to do what we need to do?

If any of the themes in this paper have resonated, then take action:
If you don't know much about change management, now is the time to learn, and the Change Management Institute is a good place to start: www.change-management-institute.com

If you want more ideas for implementation activities to include in your projects, connect with me on LinkedIn as I share links and publish articles about change activities.

About the author

I have been responsible for the successful delivery of effective change and for creating environments that support transformational change for over twenty years. I have an impressive track record of successful consulting assignments and I am an acknowledged thought leader in Change Management.

I am the Co-Chair of the Change Management Institute UK and a respected author of text books and articles on change, project and programme management. I am a talented communicator with a reputation for delivering complex information with humour and passion. I draw on my wealth of practical experience to illustrate concepts and to engage my audience in lively debates on advantages and disadvantages of each approach that I outline.

To access more resources about change management connect with Melanie on LinkedIn
https://www.linkedin.com/in/melaniefranklin1/