Agile for Everyone: Enabling Agile in a Waterfall World

By John Carter & David Vermette, TCGen, Inc.

By elevating product teams above functional allegiance, organizing work into sprints, and nesting these sprints within your existing process, any team, in virtually any industry, can become agile.

Agile isn't just for software anymore. The benefits software firms have realized from Agile are a result of applying a remarkably small number of practices. Many teams are surprised to find that the benefits of Agile rely largely – and unexpectedly – on restraining functional authority over teams. If you prioritize product teams over function you're well on your way to agility.

The next piece is to organize your activities by the triad of planning, executing and reviewing. Perform each of these activities iteratively, in short cycles called sprints. Working in sprints does not mean altering your existing phase review process. Agile and waterfall models of development are not mutually contradictory.

Agility: Not Just For Software

Whether it's chemicals, industrial machinery, headphones or the latest app, product development teams are competing against time. Each stage of creating new products, from strategic plans, to concepts on a road map, to project selection, to execution, all take time, and your team is on the clock. Shaving even a small amount out of one of these phases creates competitive advantage.

In our work, we've seen a certain amount of envy on the part of developers of non-software products as they gaze over at their friends in Software. With the unstoppable growth of mobile technologies like the iPhone, we've seen techniques that have helped to speed software products to market. Developers in every other industry may wonder if they've been left behind.

Many software teams have learned how to cut development time through a set of techniques dubbed "Agile." The fact that digital products have no inherent cost and are infinitely replicable has led to the acceptance of techniques that have accelerated software product development.

Hardware projects differ from software projects in important ways:

- They have components with long lead times
- They have tooled parts with as much as a three-month cycle time
- They have design activities of various durations

It is a misconception that these characteristics prevent teams from applying Agile to non-software products. The reality is that the most effective, high leverage components of Agile boil down to a handful of elements:

- High performance teamwork: where teams have more power than functions
- Rapid repetition of the triad of:
    1. Planning
    2. Executing
    3. Reviewing
- Nesting this triad inside the waterfall structure

Taken together, these elements, applicable to almost any product or service development environment, unleash the power of Agile.

Teams Over Function – Half the Battle

You might be surprised to learn that a great deal of what is touted as Agile is due to high performance teamwork. Teams perform at their best when:

- The roles are well-defined
- Communication flows
- Leadership comes from within the team, and
- There is genuine collaboration

Our partner Jeanne Bradford cites the following best practices for high performance teamwork:

1) *Grant complete authority to the team leader*
If you establish a culture of team accountability across your organization and empower your team to drive daily decision-making in support of the objectives of the project, you will see measurable gains in a team's ability to innovate and reduce cycle time. It is a best practice for senior management to make a contract with the team as to features, cost and schedule, intervening only when a break in this contract looks likely. Functional managers and executives must make the painful decision to get out of the way.

2) *Be clear and explicit about the roles and responsibilities of the cross-functional team*
Clarity around cross-functional deliverables and dependencies is a key driver of fast cycle time, especially as companies grow and expand geographically. Stating clearly who is doing what by when liberates teams to focus on the work required to innovate and deliver products to market. It seems that every team would do this as a matter of course, but numerous teams we have encountered lack sufficient clarity around roles.

3) *Implement a core-team model*
The core-team model consists typically of four to six functional leads, e.g. Project management, Product management, Engineering, Design, Manufacturing, and Quality assurance. The team can serve as an effective nucleus that: drives execution, helps resolve issues that arise during the project, and manages cross-functional dependencies.

4) *Create a culture of trust and collaboration*
Absent trust and collaboration, individuals shift their focus from the common good to their own survival. Teams achieve high performance when they know they can deliver bad news free of politics and without gaming the data or sugarcoating the message. They focus on solving the problem, not finding the guilty and blaming them.

If your organization excels in these four areas then you're already halfway toward agility

The Triad: Planning, Executing and Measuring

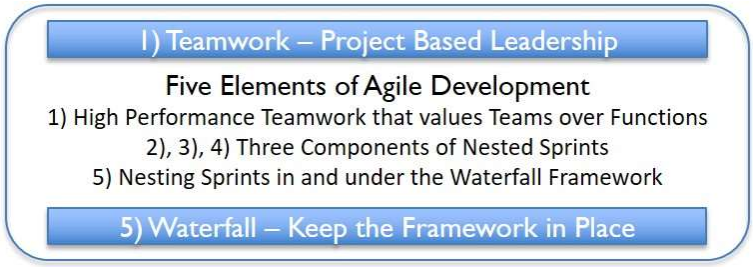
Almost all well-managed product development efforts involve planning, executing and reviewing. Transforming these activities into a series of

iterative sprints is the next stage. What are called sprints are short, bounded periods of work characterized by repeated cycles of planning, executing and reviewing. Iterative, short work cycles cut development time because they enable frequent reviews and a free flow of communication between team members. They encourage overall process discipline and provide an advantage when competing against less agile companies. You can begin to incorporate sprints into your product development process by defining each element of this triad:

1) Planning – Estimate the effort and define how progress will be measured within each segment, taking into account what was learned in the previous review(s).

2) Executing – Within each segment, capture the metric that best measures progress and display it publicly. Display the actual progress metric vs. the predicted progress for each sprint.

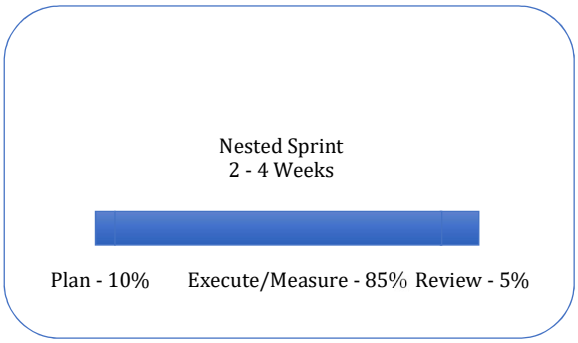
3) Reviewing – At the end of each segment, gauge the progress of the project and the effectiveness of the process.



Then, consider the work that must be performed in each phase of the development process, from early stage development to launch. Divide the time within each phase into defined and bounded work cycles, ideally of equal length between each major segment of the waterfall process. Then plan, review and measure within each of these short segments.

These short and bounded work cycles of perhaps 2-4 weeks break up a development effort of, say, 9 to 15 months into many smaller segments. In software, the sprints are uniform in length. In other types of development projects, the length of each segment varies but remains consistent within each phase. Varying the length of these segments within the phase runs the risk that they will tend to become longer rather than shorter. Making your work segments of more-or-less equal length within each phase will instill the discipline you need to gain the benefit of sprints.

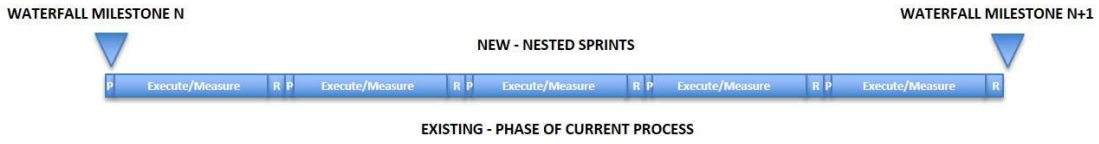
However, the length of each sprint will vary from one phase to another. For example, in the definition phase the segments/sprints may be as short as two weeks. During prototyping, where a team may have to wait to receive a circuit board from a vendor, a sprint may span six weeks. Segment lengths may vary more over a longer time period as compared with software product development. As far as possible, however, the approach that yields the best results is to keep the duration of the sprints equal within each phase.



Embedding Agile in Waterfall

A major challenge to adapting Agile to non-software environments is the perception that it conflicts with so-called waterfall processes. There is no such conflict. There is no need to move away from your existing phase gate process. Simply nest the sprints within each of the phases of your existing development framework.

While implementing sprints the team maintains the major goal posts – the phases and management review points – of its existing process. This way, your company preserves the organizational power and accountability of traditional phase gate processes while also gaining, within each phase, the flexibility and speed of Agile.



As your team moves from product definition to a focus on manufacturability and then product launch, the sprints will vary in length and focus. In fact, given the length of many hardware and systems projects, maintaining the discipline of periodic reviews enhances the effectiveness of the sprints.

Conclusion

There is magic in Agile. But it is not a mystery and you can embed it within your existing process. It is, however, a product of hard work that requires

managers to change. It is a set of principles and a related suite of practices that are intended organize the project by rapid learning and course correction. Such basics as superb teamwork, planning, light weight reviews, and tracking the right metrics for projects, can go very far toward realizing agility.

Above all, high performance teamwork makes a difference. If you do not have the level of high performance teams described above, then this is a place to start. Begin to implement best practices incrementally and then move toward greater process discipline. Agile is merely a tag, a name for a set of related tools for taking a process and wringing the time out of it. It's a race to the finish line and teams that collaborate most effectively have a better chance of winning.

If you'd like to discuss how to nest Agile within a waterfall model contact John Carter at jcarter@tcgen.com.