Optimizing Scrum-Fall Whitepaper
# Table of Contents

Abstract .................................................................................................................................................. 2  
Our Thinking ......................................................................................................................................... 3  
A comment on tools for Scrum-Fall ................................................................................................. 6  
Conclusion ........................................................................................................................................... 7  
References ............................................................................................................................................ 8
Optimizing Scrum-Fall Whitepaper

Author: Miles Faulkner, Co-CEO

Abstract

Objective

This brief paper will walk the reader through our thinking around how to make the best of "Scrum-Fall" (Scrum blended with Waterfall), which is sometimes a necessary evil in "big" IT. We aim to provide a formula for a better implementation in larger more rigid, risk averse corporate IT environments. It's important to note that Blended Perspectives takes an agnostic position on methodologies, noting their existence and making suggestions on how to optimize their implementation.

About Blended Perspectives

Blended Perspectives is an Atlassian Platinum Enterprise certified Solutions Partner. We are a Canadian specialist consulting firm, operating out of Toronto, Canada, with offices also located in St. Louis, USA. Blended Perspectives specializes in consulting around enterprise processes such as SDLC, ITSM, PPM and Project Management that drive collaboration, innovation, and productivity. Our solutions encompass strategy, processes, application configuration and technical implementations. Our mission is to enable Corporations to unleash the power of their teams and to leverage the true potential of their business via enhanced tools and processes.
Our Thinking

Scrum-Fall is a hybrid methodology composed of Waterfall and Scrum approaches. Scrum-Fall really ought to be used by organizations as a stepping stone to Agility, but the reality is most of the larger organizations we work with have some form of Scrum-Fall. Hence, we think there are some important best practices to optimize how this approach should be implemented. The first point is an obvious one - with Scrum-Fall; any planning is better than no planning at all. Right or wrong, having at least some Agile is better than none, providing the segments of the SDLC that are Agile, harness its best principles.

One of the foundations of Agile is the ability to learn from experience - i.e. build something and learn what your velocity is and how hard it is to succeed. Remember that one of the main reasons Agile has taken over from Waterfall is the failure of the "cone of uncertainty", a primary driver behind Waterfall stage and gate approaches. The cone seen below indicates a process where estimates start out being wildly wrong and as work continues on scoping and requirements, those same estimates shrink in variability. In most cases, this is what is driving the Waterfall method. The premise is that with a detailed design - coders can build to spec and uncertainty is driven to zero. The problems with this of course are:

a) Everything has changed during requirements and they never settle due to changes in business needs

b) Even though everyone says that an estimate is + or - x% - typically senior management always only remember the number and forgets the variability factor

c) Technically the design can be flawed or when the development team starts building the product, it turns out to be more difficult than thought
In contrast to the above model, Agile depends on a different paradigm. This model takes into account the inherent riskiness of building something as complex and intangible as a software system. The model below helps to better explain the approach:

![Diagram showing the Agile development cycle]

The idea here is to start development work as early as possible and to determine quickly the feasibility of building an end product. Traditionally re-work is considered evil and wasteful. Yet where learning counts, re-work is simply the price of better accuracy. When it comes to software development there is lots of evidence that this is simply a more predictable and stable method of production. Too often though, being an essential element in Agile, seems to imply that project disciplines don't apply.
We couldn’t disagree more. All the traditional tools of a PM should be present. There are still risks, issues and decisions, it’s just that the nature of iterative fast-paced development cycles, can be used to address these problems.

All of this matters because when you take the two approaches together, you often see the following pattern. Waterfall at the front end - Scrum in the middle as a phase and then Waterfall at the back end, when then, with open arms the testers rigorously prepare the application for production. Naturally enough this corrupts the purity of the Agile estimates, since for instance testing can introduce new requirements and delay the project considerably. It does however satisfy the broader constituents of stakeholders who may feel their interests are under threat for instance technology owners and operations. It may also sound like a heresy but in essence, a set of Agile sprints can be considered to be a phase or a release. Thus, it can be represented on a Gantt chart as such:

A much more important problem with the above model is simply that no matter how much quality solution architecture work is done up front, none of it may matter if development proves to be difficult. I have seen in Agile projects, multiple "sprints" dedicated to preparation and architecture sign off and so forth. Then, only later, to establish that 50% of the architecture had to change based on what we learned in the actual sprints. We would rather see a further "hybrid Scrum-Fall" approach that accelerates development and de-risks testing. A better approach could be as follows:
In this way, development starts almost immediately and informs the solution architecture. In addition, there would be at least two chunks of production testing to minimize nasty surprises at the end of a successful set of sprints. There is no exact formula for Scrum-Fall, rather a pattern aimed at minimizing the risks of complex programs and technologies by adopting Agile approaches in concert with the more rigorous needs of solution architecture validation and production testing. Finally, over time we would prefer to see organizations moving to a more streamlined DevOps-oriented approach to faster releases on a continuous basis. Yet at least the above model is a good start for organizations on the Agile journey.

A comment on tools for Scrum-Fall

Out of the box, Jira is not strong in supporting custom planning hierarchies, which will predominate with Scrum-Fall approaches in enterprises. The Portfolio for Jira-add on tool from Atlassian is also more oriented to pure Agile iterative approaches, rather than hybrid planning needs. We suggest add-ons such as Big Picture and other Gantt charts that can simultaneously capture sprints and additional stages of work in a single plan. The snapshot below shows both Gantt chart views, a customized hierarchy, and other status reporting features - all in Jira.

It’s our view that the Atlassian tool set can support any type of planning; Waterfall, Scrum-Fall, Agile, Kanban - all the flavours.
Conclusion

If a hybrid approach is present in your organization - the following are our key recommendations:

- Push teams to start development early. As they say in ITSM - "shift left"! Re-work is much less expensive early than at the rightward end of the spectrum
- Break up production testing into as many chunks as you can to minimize additional requirements unknown by the team
- Don't throw out all the good disciplines of project management - risks do need to be managed explicitly and factored into overall timelines
- If teams use Jira; blend sprint reporting and velocity with Gantt charts by add-on vendors such as Big Picture to ensure a tightly coupled reporting model of Waterfall and Scrum activities
- Be flexible based on planning needs - the name of the game is predictable success on a sustained basis; disciplined planning and Agility concepts can work together - have your tools be enablers
- Don't allow developers to plan in Jira, PM's to use MS Project and resource managers to use IT accounting solutions, like Clarity – it’s a recipe for an IT status "smog"

And remember - "Plan Together"!
References

- [http://blog.radta.com/are-you-doing-water-scrum-fall](http://blog.radta.com/are-you-doing-water-scrum-fall)
- [http://blog.radta.com/are-you-doing-water-scrum-fall](http://blog.radta.com/are-you-doing-water-scrum-fall)
- [http://www.agiledocumentation.co.uk/2015/02/what-is-water-scrum-fall-and-is-it.html](http://www.agiledocumentation.co.uk/2015/02/what-is-water-scrum-fall-and-is-it.html)