White Paper:

Agile Product Development

In the Life Sciences Industries

Monday, January 26, 2009

Table of Contents

1. Introduction

> *"This is your last chance. After this, there is no turning back. You take the blue pill - the story ends, you wake up in your bed and believe whatever you want to believe. You take the red pill - you stay in Wonderland and I show you how deep the rabbit-hole goes." Morpheus, The Matrix, 1999*

Agile development models and techniques are in increasing use in software development. These techniques may be considered one of the most important people- or management-related technologies of the last decade. We have witnessed huge gains in productivity, quality, schedule reliability, and development team credibility when development teams have heeded our advice to switch – even if partially – from older waterfall development processes to more agile processes.

But there is one industry that, developmentally speaking, still lives almost entirely in the dark ages dominated by waterfall thinking – the Life Sciences industry. It is not hard to understand why, really. This industry is heavily regulated by one of the largest regulatory agencies in the world: the US Food and Drug Administration (FDA). Although it is true that government agencies are not known for their innovation or for riding on the leading edge of technology or management theory, we can get much more specific as to why Life Sciences companies have remained behind this crucial part of the technology curve.

The first and most obvious reason is that Life Sciences companies are, understandably, consumed with passing FDA inspections and receiving FDA certifications for their products. This being the case, as long as the required quality and traceability can be achieved, productivity is quite literally a secondary consideration. This mindset stands in sharp contrast to what seems to be a "release first and fix bugs later" mentality of the faster moving silicon valley breed.

Secondly, the standing FDA guidance on software development is, as of this writing, now over 7 years old. Given how long it takes to get a new regulatory guidance document written, reviewed, and released, and given that the prior guidance document was released in 1997, it is a safe bet that work began on the 2002 Guidance in the late 1990s. So although this document does not *require* waterfall development techniques, it is most certainly written in waterfall language. Combine this fact with the intense focus of Life Sciences companies on compliance and you get an industry that feels safer with known compliant approaches than with venturing into the unknown – regardless of the potential payoffs of the higher risk approach.

At Provaré Technology, we do not believe that the adoption of agile development methods by Life Sciences companies entails anywhere near the risk that seems to be widely feared. We have read and studied the governing regulations and guidance documents, talked with development organizations in Life Science companies, and talked with former-FDA friends. As a result, we have every reason to believe that the adoption of Agile techniques by the Life Sciences industry is an idea whose time has come. If you are developing software in an industry that is regulated by the FDA, we are quite certain that once you experience the productivity and quality gains that we have seen in other industries, you will agree. In fact, we believe that you will go way beyond mere agreement. Your thinking about development, quality, and productivity will be irreversibly changed for the better.

2. The Problem – Waterfall Development and All of Its Challenges

As we pointed out in the introduction, although it is not mandated by the governing FDA guidance, the Waterfall language is used. This seems to have led to the perception that waterfall is at the very least assumed. But the development of a product under the regimented weight of the waterfall process presents some very serious challenges – some of which frustrate the very goals of quality compliance efforts themselves.

In any development endeavor, one of the primary reasons that bugs are released to the field and are then so difficult to reproduce once discovered is the excessively long amount of time that passes between the creation of bugs and the testing of the software. With the waterfall development model, testing often gets its first shot at finding bugs in newly developed code when release candidates are built. Even then, because of the necessity of proving test coverage to compliance inspectors, testers may spend the bulk of their time on regression testing and verification against requirements rather than on creatively working to break the software before it is released so that customers *cannot* break it.

Another problem with the waterfall model involves interruptions and feature creep. Whenever a development process exceeds more than a few minutes in length, priorities will change, new market information will be revealed, bugs from the field will be reported, and new sales opportunities will present themselves. All of these things ultimately result a set of requirements that will change many times over the course of a development project. For a regulated environment where

1. Requirements must be fully documented, and
2. Everything must be traced back to requirements,

the effort required for any change in requirements is amplified many times over by the ripple effect it causes.

But the regrets of waterfall development don't stop at the frustrations along the way. What often happens in the real world is that while the requirements at the end of a project look nothing like they did at the beginning, the release date imposed by management looks ***exactly*** the same. This has the disastrous result of squeezing two of the most difficult and critically important phases of development –integration and release testing – into ever shrinking time boxes. So not only are these two critical phases of development made difficult by the fact that the entire project's worth of changes have to be "eaten in one big bite," but now the time to chew and swallow that bite has been reduced by the encroachment of the dual realities of feature creep and shrinking schedules.

3. The Recommendation: Adapting Scrum to Life Sciences

The cure for the ills of Waterfall development is a total transition to a development process that is based on Scrum. Features of such a process might include:

1. Product development is broken into short (2 to 4 week) "sprints." The goal is for the development in each sprint to be as self-contained as possible (i.e., to break features down so that it is relatively rare for a feature to take multiple sprints to develop).

2. The overall release requirements become release goals (i.e., product backlog) and are divided into "must have," "should do if there is any way possible," and "nice to have if we have time." These goals are simply stated and should take only a handful of pages at the most for any release. These goals are converted into high-level "User Stories" and the beginning of a release process and enter the release backlog.
3. Full-fledged requirements are filled out by the development staff one sprint at a time.
4. At the beginning of every sprint, the team spends a day in a sprint planning meeting choosing high priority user stories from the backlog, breaking them into tasks, assigning the tasks, and estimating the time required to complete each task.
5. Testing and user documentation team members are involved in every sprint, although in some cases there may be a lag between the development of a feature and the manual testing of it.
6. In parallel with development of a given change or feature, validation staff can be
    - ✓ Helping to flesh out requirements as appropriate,
    - ✓ Planning tests and writing test cases.
7. During sprint planning meetings, all of the stakeholders that want to be involved (i.e., management, marketing, sales, customers, etc.) can be involved. These non-development representatives are there to voice priorities and preferences, to describe their visions of how they see a new feature working, etc. But they are not there to decide how much work will be done during a sprint. Only the developers can decide this because only the developers can possibly know how quickly something can be done.
8. At the end of every sprint, there should be several new features – either fully or partially implemented – that can be demonstrated to the stakeholders. The team's goal should be to demonstrate as much as possible. The purpose of the demos is to
    - ✓ Show progress,
    - ✓ Solicit feedback on implementation of the release goals.
9. At the end of every sprint, there should be a sprint retrospective where the entire team discusses what went well, what went poorly, how to do better, etc.
10. Once a sprint begins, there are daily 15 minute status meetings where each member of the team briefly states:
    - ✓ What I did in the last 24 hours
    - ✓ What I plan to do in the next 24 hours
    - ✓ What is blocking (i.e., preventing) my efforts
    - ✓ (In the purest implementation of Agile development, there are no chairs in the room where these daily meetings are held and they are called "stand-up" meetings or simply "stand-ups.")
11. At daily sprint meetings, only the team members doing the work (i.e., development, testing, documentation, product owners, and the ScrumMaster (project manager) can speak. Other stakeholders may attend, but are not allowed to speak.

The beauty of such a development approach is that it acknowledges that there will be interruptions, requirements changes, feature creep, shifting priorities, etc. and it provides a means of managing these realities and turning them into assets to a project rather than liabilities. The goal is that at no point in time should the team be more than 1 or 2 sprints away from getting a release out the door – with whatever features happen to be implemented at that time.

Adopting such a Scrum-like process will dramatically improve your quality and productivity. In fact, you could stop here and be perfectly happy with the result. But once you have such a process in place and working smoothly, there are other Agile development techniques that can be put into place, such as test driven development and continuous integration, to name but two. The above Scrum-based process will reduce the typical time between introduction and discovery of a bug to a few days. TDD and Continuous Integration will reduce that time for many bugs to a few minutes.

4. Unique Considerations for Life Sciences Applications

The biggest challenges encountered in transitioning to an Agile development environment in a regulated environment is

1. Meeting the documentation and traceability requirements, and
2. Convincing reviewers – both internal and external – that you are, in fact, meeting the documentation and traceability requirements.

The bad news is that such changes involve getting everyone involved onboard with a new paradigm. The good news is that it is ***only*** a matter of paradigm that stands in the way. In other words, these changes can be made while fully adhering to both the letter and the spirit of all of the governing FDA regulations and guidance documents.

We won't completely cover the details here, but we can summarize the necessary shift as a transition from attempting to create biblically comprehensive requirements, design, and test documents to documenting changes from one release to the next. Teams will have to become comfortable with writing requirements in parallel with development rather than trying to fully specify them before development begins.

But in the end, if stakeholders can make the shift, fully compliant documentation will be ready with every release of the product. Reviewers lives will be greatly simplified by the fact that they only have to review changes to document they last approved. And documents will be shorter and more readable.

5. How Provaré Can Help

Transition to an Agile development environment is a system shock to any team and is especially difficult in a highly regulated environment because of the paradigm shifts required. This transition is not something that any team wants to undertake without experienced coaching. Provaré has extensive experience both as a participant on Agile teams and in helping to coach teams through the transition process.

Additionally, attempting to transition to Agile development before the tools support such a high-velocity process would be a fool's errand. Provaré can help to ensure that the CM and team management tools are in place and are fully ready to support the transition.

6. Conclusion

As we stated in the Introduction, we believe that the adoption of Agile development practices by the Life Sciences industry is an idea whose time has come. We are seeing more and more discussions of this subject matter. For now, these discussions are mostly confined to online discussion boards and individual resumes. Next, articles will begin to appear in the literature. After that, one day soon, you'll see it mentioned in one of your competitors' quarterly earnings reports. And maybe after that, the FDA will get around to updating its guidance.

Would you rather read about the tremendous gains offered by the switch to agile techniques in your competitor's quarterly report or write it in yours?

References

U.S. Department Of Health and Human Services, Food and Drug Administration, Center for Devices and Radiological Health, Center for Biologics Evaluation and Research, "General Principles of Software Validation; Final Guidance for Industry and FDA Staff," January 11, 2002.

About Provaré Technology

Founded in 2004, Provare Technology's mission is to help engineering teams do what they already do, only better and faster. We do so by offering technology- and tool-agnostic services in quality assurance, process improvement, and data management.

Our very first client after founding the company was Theragenics, a life sciences company that manufactures radioactive "seeds" for treatment of prostate cancer. We were retained as an independent third party to test an automated QA machine to ensure that it was properly passing/failing manufactured product and to provide the necessary documentation to get the FDA approvals needed to place the machine into operation. We did so quickly and cost-effectively.

Since that time, we have maintained our relationships with those original customers, but in addition to our QA work, we have also done considerable work helping other development teams - both in and out of the life sciences community - to experience the benefits of adopting more agile development methodologies. From garage shops to AT&T, we've been able to measurably help our clients' engineers adapt to the increasing pressures of today's fast-paced budget-constrained world.

Please visit us at http://www.provare.com/ or call us at (770) 576-1926 and let us know what quality & productivity challenges your engineering teams are facing today. We might very well be able to help.