

Applying Scrum and XP in an Enterprise Context

Dr. Uwe Bombosch, Dr. Stefan Bente, Dr. Margret Hesselmann, Gero Seifert

Tata Consultancy Services Deutschland GmbH

Messe Turm

60308 Frankfurt am Main

uwe.bombosch@tcs.com

stefan.bente@tcs.com

margret.hesselmann@tcs.com

gero.seifert@tcs.com

Abstract: This article describes the challenges in applying the Scrum and eXtreme Programming (XP) software development practices, summarising several years of industry experience. The crucial questions are: How applicable are these practices in an enterprise context, and do they show their well-known advantages under today's characteristic business conditions? The assessment takes the project management, enterprise, human, software engineering, and business viewpoint in order to shed light on applicability gaps, and best practices to fill them.

1 Introduction

In the past years, we consistently applied Scrum as defined in [BS02] and interspersed this approach with elements from XP, namely the requirement capturing by user stories, and practices like pair programming, continuous integration (CI), and test-driven development. We've seen many successful Scrum/XP projects, and do neither question its general applicability, nor hold that its value proposition is invalid under certain conditions. Nevertheless, applying Scrum/XP in an enterprise context poses a number of challenges that are not covered by standard Scrum/XP, and requires complementary approaches.

2 Challenges in the Project Management Dimension

2.1 Large and Distributed Teams

Large-scale projects are generally more demanding than smaller ones, but the following factors are particularly critical for bigger, distributed Scrum/XP projects.

Communication: Minimizing the number of involved sites, collocation per site and frequent mutual visits facilitate the ideal of face-to-face communication.

Visibility of progress: In addition to management-centered instruments like task tracking or status meetings, CI enables stakeholders to inspect the actual product and quite reliably determine the progress of product creation.

2.2 Tooling in Large Enterprise Projects

Adequate tooling and automation help keeping the aforementioned factors at bay.

For *project management* we have successfully used *Atlassian Jira with GreenHopper plug-in*. They support all essential Scrum/XP practices in distributed mode, are stable, very intuitive to use and come with integrated issue tracking.

For *Continuous Integration* there are mature open source tools. Yet, putting these tools together in order to run fully automated end-to-end builds, from SW version control to automated smoke tests in the reference environment, tends to be *far* underestimated.

For CI it is essential that developers integrate their work daily without breaking the end-to-end build. The CI set-up has to support this, e.g. by staging and by fast feedback (end-to-end build < 10 minutes). A “build manager” should instantly react to broken builds.

3 Challenges in the Human Dimension

3.1 Changed Roles, Skills and Mindsets

Scrum/XP enforces working in a cross-functional team. That is why job-descriptions, roles, privileges, and compensation models must be changed. There is, for example, no “Tester” role in Scrum/XP.

Soft skills are gaining in importance. Non-technical stakeholders enter the every-day-life of programmers, and take part in the development lifecycle. The individual’s work and its results are frankly exposed to the whole team. Perfectionists have to learn to provide simple solutions, and to enhance the design on stakeholder requests only.

3.2 Cultural Aspects

Cultural factors may impact the introduction of Scrum/XP, which requires a high degree of autonomous, self-responsible thinking and acting. This tends to conflict with hierarchical team structures and autocratic leadership common to Asia. On the other hand, the strong preference for teamwork in Asian cultures is beneficial to Scrum/XP.

4 Challenges in the Software Engineering Dimension

While Scrum/XP on the one hand washes away some idle production of up-front specifications that is inherent to the waterfall approach, the principles and practices for doing architecture, design, programming, and testing must not be thrown overboard under the “we’re agile now” pretence.

4.1 Architecture and Design

4.1.1 *Large Systems*

Architecture work on a large system cannot be done on the Sprint planning day or ad-hoc while doing pair-programming. Large systems or even enterprise application landscapes cannot be developed without applying a suitable architecture framework.

In Scrum/XP, we plan and execute architecture work as ordinary Sprint tasks. The modeling of business processes, for instance, is formulated as a user story, broken down into tasks, and then phased into a Sprint.

The challenge here is to find the appropriate heartbeat for the architecture work. Overspecification must be avoided, but there always has to be sufficient guidance to support the project execution. On the planning day, the architecture must be clear enough to derive concrete implementation tasks.

One of the most wide-spread and complete frameworks is the TOGAFTM Architecture Development Method (ADM) owned by the Open Group consortium [OG09]. Its iterative, incremental nature, and the openness towards tailoring and stripping-down makes ADM a suitable choice for large-scale Scrum/XP projects.

The need for an intentional architecture is a natural limitation of the stakeholder’s freedom to choose user stories for implementation in the next Sprint. Picking a user story that is architecture-wise not covered triggers the next cycle of architecture work, and therefore comes with a high price tag.

4.1.2 *User-Interface Design*

Transforming a user story or a use case into a UI specification requires a dedicated design step, involving intense discussions with the stakeholders. The use of a UI mocking tool like Balsamiq facilitates this process, but a plain paper prototype also works.

A visual representation enables stakeholders to envisage the end-product, and uncovers omissions and ambiguities in the verbal requirement description.

Such UI specifications are dedicated artefacts that should be created during the Sprint. A possible process for doing it is to shape the specification and mockup in Sprint N, review it by a usability test and a customer demo, and phase it into Sprint N + 1 for implementation.

4.2 Testing

Agile has a strong affinity to test automation. Executing unit and end-to-end tests during the build is an integral part of the CI concept. If the quality assurance of an enterprise is not adapted to automated test cases, agile concepts are harder to put into practice.

In some areas of testing, e.g. exhaustive performance tests, continuous testing is not possible in an economic fashion. Therefore, certain specific test phases have to be taken into account during Sprint planning, also with a respect to. the availability of test environments.

5 Challenges in the Enterprise Dimension

5.1 Process Incompatibilities

There are areas where incompatibilities between standard enterprise processes and a Scrum/XP approach can be expected. In some cases, the friction is due to the resistance of a waterfall-centric organization against Scrum/XP methods. In others, Scrum/XP methods just uncover hidden omissions and problems of an organization's processes.

5.1.1 Smuggling Scrum/XP Elements into a Waterfall Process

A pragmatic solution to overcome incompatibilities and the organizational resistance is to stick to a traditional waterfall process, but sneak in some Scrum/XP. Two examples for such elements, that have proved successful in practice, are:

- Dividing the long releases into informal Sprints, with a regular demo session, called a “stakeholder review” if Scrum/XP terminology is to be avoided.
- CI should be implemented on a moderate-effort-level, as a tool for ensuring quality.

5.1.2 Budget Planning

The enterprises budget planning typically has a time horizon of about a year. Promoters of a project candidate must in advance put a quite reliable price tag to it and convince the sponsors about the business value.

Proposals saying that „we’ll start off agile with N developers, and let’s see where we end“ are rejected by the planning board. Scrum/XP projects need a roadmap in advance that at least on the granularity of Epics stipulates the target of the endeavour. Likewise, efforts must be estimated and approved.

As with the need for an intentional architecture that we discussed in Section 4.1.1, this limits the freedom to pick, change, or add user stories for implementation, too.

5.2 Specifics of Scrum/XP Offshoring

Running Scrum/XP projects with a contribution from offshore teams poses special challenges. The predominant benefit and the major risk stem from the business model of offshore companies of minimizing cost per head, which implicitly leads to temporarily minimizing productivity per head.

Lower productivity per head implies a larger project team, which in turn leads to a more difficult communication. Moreover, the general communication issues for large-scale Scrum/XP projects remain (cf. previous Section), and the solution we propose further increases cost.

The importance of customer involvement can hardly be overemphasized for Scrum/XP projects. Scrum/XP suppliers must be present onsite with a significant number of project members for a substantial amount of time. This is a major cost driver, which again conflicts with the cost-saving goal.

6 Challenges in the Business Dimension

6.1 Project Acquisition Phase

The vast majority of requests for proposals (RFP) favour or even enforce a waterfall methodology. Yet, when the authors tried to acquire such projects and proposed Scrum/XP, many purchasers valued its benefits of flexibility and transparency. However, if substantial effort had gone into a waterfall-oriented requirements specification already most purchasers were not willing to directly invest more into discussing and prioritizing requirements.

The actors on the ground typically acknowledge that such a “fire-and-forget” project ramp-up will not work well. They experienced that the collaboration with the supplier on user stories and an architecture vision shows better results in the end. But applying the Scrum/XP paradigm of “collaboration over contract negotiation” before the first contract has been signed is rejected by most purchasers as it binds them to the supplier too early.

A possible mitigation of this conflict is “smuggling in” Scrum/XP elements; cf. section 5.1.1.

6.2 Agile Contract Models

The agile conception of a project scope gives purchasers a high flexibility in changing requirements or re-prioritizing them. But under business conditions, the service provider and the purchaser have to agree on a contract nailing down the pricing conditions.

The challenge is to choose a contract model that does not undermine the agile flexibility. "Time and Material" (T&M) is such a model, but unpopular with purchasers as it leaves most risk with them and does not force providers to work efficiently. An example of a better model is "Fixed Price per Iteration with Story Point Guarantee", which provides flexibility but also enforces that all deliveries have a certain comparable complexity.

A lot of contract models have been invented in order to mitigate the tension between agile flexibility and calculable prices [St09]. However, none of these models have gained market-acceptance yet.

7 Summary

As we've seen, the introduction of Scrum/XP to an enterprise context shows challenges in several aspects. The hot spots are probably in the engineering and the business dimension.

The authors have seen large projects in trouble because they neglected the need for an architecture framework, and silently assumed that the target architecture would show up by magic while "being agile".

However, the most demanding friction, at least from the perspective of a service delivery company, is the RFP acquisition process. Nevertheless it is observable that companies start considering this friction also as an opportunity to come to a different mode of operation, and a better collaboration with their service providers. Time will tell whether Scrum/XP must be enhanced to support fixed prices and RFPs, or vice versa.

References

- [BS02] Beele, M.; Schwaber, K.: Agile Software Development with Scrum. Prentice Hall 2002.
- [Co04] Cohn, M: User Stories Applied. Addison-Wesley 2004.
- [HMR09] Hastreiter, G.; Mathis, C.; Roberts, S.: Leuchtf Feuer Entzünden: Einführung von Enterprise Scrum bei der Allianz Deutschland AG. OBJECTSpektrum 2009; S. 54-60.
- [LV10] Larman, C.; Vodde, B: Practices for Scaling Lean and Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum. Addison-Wesley Longman, Amsterdam, 2010.
- [OG09] The Open Group: TOGAF Version 9, <http://www.opengroup.org/togaf/>, 2009.
- [Sc10] Schwaber, K.: The Home of Scrum; <http://scrum.org/scrumdeveloper>, 2010
- [St09] Stevenson, P.: 10 Contracts for your Next Agile Software Product, April 2009, <http://agilesoftwaredevelopment.com/blog/peterstev/10-agile-contracts>