

Challenges: Agile Values Meet Different Value Systems

Jörg Pechau

CoreMedia AG
Ludwig-Erhard-Straße 18
20459 Hamburg, Germany
jpechau@acm.org

Petra Becker-Pechau

University of Hamburg, C1 WPS
Vogt-Kölln-Straße 30
22527 Hamburg, Germany
becker@informatik.uni-hamburg.de

Abstract

One might expect that nowadays object-oriented software development and agile methods are mainstream in industrial practice. Especially the handling of method mixtures as we face it when different organizations and disciplines have to work together should have been addressed by now. Actually conflicts arise exactly at that point, in the worst case leading to failed software projects.

Our assumption: The choice of methods and tools is an expression of the value systems of an individual, a team, an organization etc. This applies to software development as well as to any other discipline. Different value systems are accountable for conflicts.

The question is: how do we, as software developers who value agile methods and object-oriented approaches, handle such structural conflicts?

Categories and Subject Descriptors D.2.9 [Management], K.6.3 [Software Management]: Software Process

General Terms: Management, Human Factors

Keywords practice of software development; agile; project management; values; industrial practice

1. Workshop Theme

One might expect that nowadays object-oriented software development and agile methods are mainstream in industrial practice. Especially the handling of method mixtures as we face it when different organizations and disciplines have to work together should have been addressed by now. Actually conflicts arise exactly at that point, in the worst case leading to failed software projects.

Copyright is held by the author/owner(s).
OOPSLA '08, October 19–23, 2008, Nashville, Tennessee, USA.
ACM 978-1-60558-220-7/08/10.

1.1 Introductory Example

To explain the assumption, here is an example of a project that has not failed but was full of conflicts: A worldwide roll out of a digital rights management (DRM) system functioning as part of an online music download service of a telecommunication provider (Telco). It was a subproject of the overall project to provide a music download service to the Telco customers. In this subproject many disciplines and organizations had to cooperate depending on each other without being integrated into one project organization. In detail:

- There was a standardization group to define an open DRM industry standard which affected nearly all technology providers of the project. Participants of this group not only came from the project but from nearly all significant players in the telecommunication industry: operators, handset vendors, backend vendors etc. Work was done following a classical top down approach, which struggled to find a way between a technically clean solution and the interests of the companies involved. The music download service project's requirements and feedback due to the standard were handled within this standardization group.
- Several handset vendors were asked to support the music download service including the DRM standard. These vendors had to implement client software in their software departments and had to synchronize these development tasks with their handset manufacturing lines in order to issue the software in time, so that handsets supporting the DRM standard would be available at the music services launch.
- A backend vendor built the DRM server side using object-oriented tools and agile methods, having to ensure the compatibility with the handset vendors' client software. While the development was still in process in par-

allel the backend vendors' consulting department had already started customization. They were working on the same topic following a classical top down approach, not an agile one.

- The project initiator, a major Telco, a global organization, had a program manager for the overall project and local subsidiaries, global and local operation teams, all of them defining fixed milestones for deployments.

This real world example incorporates many different disciplines, and even within identical disciplines such as software development, different methods and tools were used. Naturally this project was not free of conflicts: between different vendors and between different suborganizations within an organization.

1.2 Shortcomings in Software Development

It is not the task of software development to provide a solution for all these conflicts, but it would be most helpful if we were aware of the world outside our software development scope and could reflect on this in our methods and tool sets. At the moment the software development practice focuses mainly on the relationship between software development and customer. We tend to ignore the interactions and interferences with further project participants, aware only of a subset of interfaces of a project. Besides of having an "onsite customer" or "product owner" etc., agile methods provide only very unspecific answers like: "communicate in time, frequently, and open", "be flexible but plan carefully", and "have a good change request management". In our opinion this is not enough. Software development does not really have a good understanding of the mechanisms of the above mentioned conflicts.

1.3 Values Conflicts

Our assumption: The choice of methods and tools is an expression of the value systems of an individual, a team, an organization etc. This applies to software development as well as to any other discipline. Different value systems are accountable for structural conflicts. This might not always be obvious nor have a solution. It means we have to take underlying value systems into account when working in larger contexts.

Values are nothing new in software development and beyond:

- The Agile Manifesto: "Individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan" [3].

- Scrum refers to the values of the Agile Manifesto and discusses: commitment, focus, openness, respect, courage [6].
- Referring to the Agile Manifesto XP mentions five values explicitly: Communication, feedback, simplicity, courage, respect [2].

All of them are suggesting values to be used in an agile approach, which focuses on relevance to the customer and quality of the software to be developed.

Compared to PMIs, "Project Managements Body of Knowledge" [1], especially their "Code of Ethics" [5], the potential of running into conflicts with an agile team working within an overall project, e.g. PMBOK, becomes obvious. PMBOK is a method of the project management discipline, not a method focussing on customer centric software development. It's values do not address e.g. anything like the quality of the produced project's deliverables.

A final example for postulating values is the well known and often quoted "Zen and the Art of Motorcycle Maintenance" [4]. It is a book about values, concentrating on quality. Quite a number of authors recommend this book to software developers, project managers etc. If working for an organization that considers quality as a cost driver only, adopting the ideas of this book will be challenging.

Returning to the case above, following is a brief example of confliction values: The customer organization was bound to many well defined and detailed processes, e.g. for operations, roll outs, marketing. The project manager was PMI certified, which in total led to a rather strict top down overall project plan. At this point the backend vendor's software development group, who tried to work close to the customer, responsive to changes, etc. got in conflict with the customers processes. "Follow the Process" met the Agile Manifesto's value systems.

2. Workshop Goals

How do we, as software developers who value agile methods and object-oriented approaches, handle structural conflicts coming from differing value systems? In detail:

- What are typical conflicts and how do we recognize them?
- How do we approach these structural conflicts? When are we limited to work on symptoms only, and when can we tackle the cause?
- If there is no solution, how can we mitigate the conflicts?

For this workshop we are aiming at a discussion between research and industry – the workshops' goals are:

- Initiate a discussion about value systems and their impact on our choice of tools and methods in state of the art software development
- Identify typical categories of value system based conflicts
- Discuss approaches to tackle or mitigate conflicts coming from differing value systems
- Find ideas how to improve software development methods and tools
- Discuss ideas to improve software development education
- Form a community to continue the workshop's discussions

3. Workshop Conduct

The workshop organizers aim at initiating a discussion and forming a community about value conflicts in software development. Therefore, most of the workshop's time is used for interactive activities. Attendees issue a position statement of 1 or maximum 2 pages. The organizers asked for submissions from people who have researched on these subjects or who can provide practical experience with regard to the problem. Initially, workshop participation is based on the acceptance of the position paper, but, if the available space permits, the workshop will eventually be open to all who are interested.

The workshop is intended to be highly interactive; therefore the workshop's format will be an issue of all participants. Initial suggestion: At the beginning of the workshop, each participant gives an introductory statement of 3 minutes maximum. The topics that the position statements bring about are discussed in open space focus groups. After a presentation of the focus groups' results, subsequent actions are discussed.

Upon acceptance, submissions will be posted on the workshops blog <http://oopsla2008workshop.2isnot3.eu/> and will be open for discussion. The workshop blog will be held operational beyond OOPSLA 2008.

References

- [1] A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Third Edition, Project Management Institute, 2004
- [2] Beck, K., Anders, C., Extreme Programming Explained, 2nd Edition, Addison-Wesley, 2005, pp. 17-21
- [3] Fowler, M., Highsmith, J., The Agile Manifesto, Software Development, August 2001
- [4] Pirsig, R., Zen and the Art of Motorcycle Maintenance.: An Inquiry into Values, William Morrow & Co, May 1979
- [5] PMI Code of Ethics, http://www.pmi.org/PDF/AP_PMICodeofEthics.pdf, October 2006
- [6] Schwaber K., Beedle M., Agile Software Development with Scrum, Prentice Hall, 2002, pp. 147-154