

Conflicting Value Systems in Agile Software Development Projects

Jörg Pechau
Universität Hamburg
Fachbereich Informatik
Software-Technologie
+49 (0) 151 52 58 29 21
jop@2isnot3.eu

ABSTRACT

Agile software development projects executed in larger project environments, that is, within a larger project and/or organization, often struggle to succeed. This can be the case despite overall framework conditions being good. This often can be related to conflicting Agile value and non Agile value systems. This article is the continuation of [7]. It is a result of my research regarding patterns of value based conflicts of Agile software development projects. The pattern listed in this paper shall help to identify and to address aforementioned conflicts when using Agile approaches.

Categories and Subject Descriptors

D.2.9 [Management], K.6.3 [Software Management]: Software Process

General Terms

Management, Human Factors

Keywords

Agile, Patterns, Value Systems, Software Development, Software Project Management

1. Introduction and Overview

Agile software development projects executed in larger project environments, that is, within a larger project and/or organization, often struggle to succeed. This can be the case despite overall framework conditions (e.g. stakeholders experiences, skills etc.) being good. This often can be related to conflicting Agile value and non Agile value systems.

Agile software development projects refer to projects accomplished using Agile methods like XP [2]) or Scrum [8]. Agile development roots itself in „The Agile Manifesto“ [6]. Today greenfield software development projects are an exception. Instead, projects are executed in more complex project environments. [5] defines a project environment as: „The environment in which the project begins and will be

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission. A preliminary version of this paper was presented in a writers' workshop at the 18th Conference on Pattern Languages of Programs (PLoP). PLoP'11, October 21-23, Portland, Oregon, USA. Copyright 2011 is held by the author(s). ACM 978-1-4503-1283-7

accomplished“.

This article is the continuation of [7]. It is a result of my research regarding patterns of value based conflicts of Agile software development projects in non trivial project environments. The research question itself was based on observations coming from investigating a number of case studies.

Values define meaning and purpose of a group. They define the measure and are directing our actions, enabling us to evaluate different means of acting. [3] writes: „Thus, the values of an organization are the foundation of not only what is done, but also how it is done.“ The choice of methods and tools is an expression of the value systems of an individual, an organization etc. This applies to any discipline and is not restricted to the software development domain. Different value systems are accountable for structural conflicts, but this is not always obvious nor can they be solved in every case. Since values are the determining element of a culture and accountable for special categories of conflicts, we cannot evade the inherent conflicts and therefore have to take underlying value systems into account when applying Agile development approaches in larger contexts.

Agile development has hit the mainstream [10]. Almost all projects are executed in non trivial project environments. These projects interact with there project environment on different levels, e.g. economical, technological, communicational. At this point of contact Agile teams encounter different cultures hence they encounter different value systems.

Agile methods are self referential and so are traditional phase oriented project management standards. The actual Agile literature supposes enterprises to transform themselves into Agile organizations as a critical factor of success. This usually is beyond the level of influence of an Agile software development project. There is little help to be found in the literature, what can be done to identify and handle aforementioned value conflicts.

An Agile project in such a context cannot avoid the described area of conflict. My approach integrates Agile software development, value based conflicts, and project environments, by describing the identified problems and solutions as patterns. My point of view is that of an Agile software development project, working in a non trivial project environment, which I cannot influence or change in a significant way. Part of this environment are the value systems of other person and organizations, which makes it imperative to examine the Agile values as well.

2. Audience

The intended audience of this article and of the described pattern approach are people applying Agile methods within non trivial project environments as well as people introducing Agile methods into non Agile organizations. It addresses people who are working in a non Agile context and having to interact with Agile teams or to integrate Agile methodology into their set of methodologies, and who want to understand potential conflicts having their cause in this.

Beyond this, the article might be of interest when educating people in Agile approaches.

3. The Patterns

The patterns itself can be categorized as a organizational pattern [3] or a process pattern [1]. The pattern format I chose, addresses an immediate solution approach in a section called „Short-Term Solution“. Since these conflicts have their root causes in underlying conflicting value systems, I have added a section „Long-Term Solution“, to describe how to address the underlying value conflicts.

In most cases the Agile projects are using Scrum as the Agile approach, but the patterns described are not bound to this.

The patterns address the needs of an Agile teams executing an Agile software development project as part of a non trivial overall project. Instead of being part of an overall project, the Agile project could be executed directly in a non Agile organization, which would lead to same problems. I have chosen the project environment, since this was the majority in my case studies.

3.1 Open Up Information Sharing

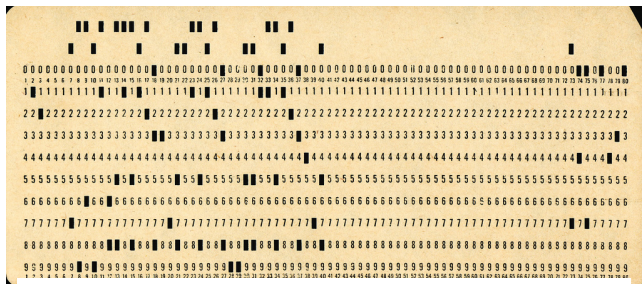


Figure 1. „Knowledge is Power!“, Sir Francis Bacon

3.1.1 Context

By definition, a self organized team must be able to act autonomously. For this they need to have fast and free access to information they need for their work.

3.1.2 Example

In the context of a large project for a large european compulsory health insurance institute, which was led by a PMI (Project Management Institute, www.pmi.org) trained program manager, one of its sub projects was an Agile software development project. The Agile projects goal was to build administration functionality for a huge health care application. The Agile project depended on early information about API and semantic changes with regards to the health care backend, being developed in an other sub project. Officially the backend projects project leader assured the pro active delivery of needed information. In reality this never

happened. Information was shared late, causing delay in the Agile project and as a consequence causing a negative impression of the Agile project. As a side effect the Agile project in trouble drew off the attention of shortcomings in the backend project.

The Agile projects leader (in this case not Scrum) spent most of his time hunting for information needed and in escalations to obtain information. This even worsened the bad impression of his project, since people recognized him as „finger pointing“ resp. „blame and complain guy“.

3.1.3 Problem

Although required information should officially be or should be made available to the Agile team, it is still withheld. The Agile team only is able to access it, if at all possible, when they are determined and insistent, spending a significant time on chasing information. Furthermore, uncertainty about whether the information is complete and accurate leads to additional validation cycles.

Additionally this might cause insecurity, whether the information is complete and accurate, leading to additional validation cycles. Altogether the ability of the Agile team to work autonomously is weakened.

How do we handle this kind of information hiding?

3.1.4 Forces

- Agile work is about self organization and frequent feedback in short cycles. Therefore the Agile teams need quick and transparent access to information.
- An information stakeholder (someone who has an interest in a certain peace of information) may be willing to share information but not be aware of the Agile team information needs.
- An information stakeholder may not have the information and/or the means to share them.
- An information stakeholder may not be allowed to share the information following company guidelines or an existing communication plan.
- An information stakeholder may simply follow a companies culture of controlling information.
- An information stakeholder may fear to lose control over his/her information.
- An information stakeholder may seek advantage in controlling information.
- An overall project manager is interested in sub projects sharing information they need to reach their goals.

3.1.5 Short-Term Solution

Create a solid foundation by negotiating and defining a communication plan - if it does not already exists - with overall project management. The communication plan briefly captures information needs of a whole or parts of a project. When put in place it makes deviations e.g. information withheld visible. If this „contract“ will not be honored, remind the information hider. If information still is not shared, you might have to escalate this matter.

In detail: In a first step assure support from the overall project manager, by explaining the need for at least in some aspects formalized, documented and measurable communication. In order to underline the importance of the Agile teams need, the consequences and/or project risks if information exchange does not improve has to be pointed out by the Agile team to overall project management. If it does not exist, the Agile team need to

create a so called communication plan, or take it as a starting point if one already exists.. A communication plan formalizes information needs of a project or parts of it. It therefore can cover stakeholders, process, artifacts, dependencies, scheduling, and documentation of information exchange. It is part of „classical“ project management planning (standards). The communication plan should at least for to the Agile project should be kept clear and simple.

Since this plan captures the Agile projects point of view, how communication/information access would be ideal for the Agile project, it has to be negotiated with affected stakeholders. If done the communication plan should be taken in place by the overall project manager.

To convince overall project management or to negotiate with affected stakeholders, support from someone not directly involved, with some credibility in project management for moderation should be considered.

If the communication plan has been established it can be used as a measure and baseline reference, if information exchange works. It exposes and documents deviations and makes it difficult, to conceal information hiding. The plan provides us with facts in case of a potential escalation.

In a way a communication plan defines an communication interface to the Agile projects environment.

3.1.6 Example Resolved

In this case, the Agile team’s scrum master asked an external PMI certified project management professional from his organization for counseling. Together they outlined a communication plan and approached the overall program manager of the project to implement this plan. The program manager understood the approach and asked the backend project leader to join. Without blaming or finger pointing the overall projects communication plan was extended with a lean plan covering the communication between the two sub projects. Special attention was placed on the frequency and documentation of information exchange.

Table 1. Simplified Communication Plan Extract

Meeting Type	Frequency	Time	Accountable	Attendees	Agenda	Notes
Synch Meeting	Weekly	Monday, 10:00 - 10:30	PM (B)	Mandatory: Agile Delegate (A), Architects (B) (A) Optional: PM (C)	Exchange projects status with regards to the backend interfaces, availability of develop-, test- and integrations backend. Check proposed CAs with regards to backend interfaces. Check last meetings action items.	Minutes including action items on file share, link on intranet by mail, CAs/Issues
Planning Meeting (B)	Bi Monthly	last Thursday, 14:00 - 15:00	PM (B)	Mandatory: Agile Delegate (A), Architects (B) (A) Optional: PM (C)	Integrate (A) demands into next iterations plan update for (B)	Updated project plan, CAs/Issues
Review Meeting	Bi Weekly	Friday, 10:00 - 11:00	Agile Delegate (A)	Mandatory: Customer, Agile Team, PM (B) Optional: PM (C), guests	Review of last sprint	
Planning Meeting (A)	Bi Weekly	Monday, 10:30 - 11:00	Agile Delegate (A)	Mandatory: Customer, Agile Team, PM (B) Optional: PM (C), guests	Check assumptions for upcoming sprint, check on shared milestones, integrate (B) plan details	Sprint Backlog, shared milestones

3.1.7 Forces (Continuation)

Information hiding can come from several directions. As described in [7] (pg. C4-10):

„Classical project management standards like [11, 12] „plan“ communication and the organization of information. Communication and the flow of communication radiates from and to project management.“ In a company following a control paradigm the goal in planning communication is to control information. How to handle this I have described in the pattern „We need to know“ [7].

Beside that there are many reasons why information might be withheld:

- It can be unawareness of the information need.
- It can be fear of losing control.
- In hierarchical organizations very often it is the exercise of power.

Withholding information if on purpose or not puts the withholder in a position of power, since others are depending on him/her.

As an Agile team we cannot simply delegate this to overall project management or above and wait for the information to be delivered, after finding its way up and down through the hierarchy of the project organization. If we would do so, we would lose our ability to act autonomously and our responsiveness would be shortened. A typical reaction are to resist, to circumvent, or to avoid the situation. Avoiding to address the conflict puts us out of Agile business sooner or later. Resisting and circumventing does not necessarily solve the problem, but has the potential to create new conflicts. This time we might be able to access the information we need, but we will not be sure whether we obtained the right information at all. What is more, we don’t change the way the information hider acts. On the contrary, if noticed s/he might recognize this as a challenge, which can lead to open conflict and/or most likely an increase of his/her behavior on future occasions. So this would even worsen the situation.

The Short-Term solution provides an approach to improve the situation at hand without addressing the underlying conflict itself. To address this, we have to understand the conflicting value system and to acknowledge that there are no immediate solutions for these conflicts.

Information as means to exercise power vs. Transparent information as basis for self organization

3.1.8 Long-Term Solution

If we are facing an „information hiding“ situation in principle there can be two main categories of reasons for an information stakeholder to act like this: It can be destructive or power oriented behavior, or it can be fear of losing power resp. influence, control, and/or reputation. Working with the information stakeholder on the short-term solution is next to its core purpose a means to the end to begin understanding the information withholders motivation.

The first part might find its causes in the person or in an organization itself. Anyway if we don’t address the conflict, it will not go away on its own. Since the information hider is successful with his behavior, he will reuse it, when in doubt. If the only motivation is exercise of power for its own sake, this behavior will not change. But if we avoid addressing the conflict, someone else will set the rules to our disadvantage. To address this, we can point this out to the overall project management or even above to be a problem for our project. We can show the consequences, and try to mitigate the impact on the Agile project by applying the solution above. The short-term solution provides us with the facts needed for this. The chance for this to work are good, since nobody likes to expose himself as being destructive. The core conflict we cannot really address, since most of us are no therapists. If the organization has a culture of exercising power on its own, this is beyond our scope.

3.2.4 Forces

Manage Changes vs. Embrace Change

Traditional phase oriented project management standards are set up to deliver a certain scope within time and budget constraints. The assumption is, if a project is delivered according specification, the project owner as customer will be satisfied. Therefore changes to a project are handled as exceptions to be managed closely and mainly are considered as possible way to increase revenue. Project managers are held to look out for additional change requests which can be billed, and to avoid any other change since these might pose risk to the project, without bringing commercial benefit to it.

On the other hand Agile approaches are putting the end user as customer in the middle of their approaches. Kent Beck has put it in the title of his book [2] „Extreme Programming Explained: Embrace Change“. Having a customer being part of the Agile project helps to provide software to him he really needs, but might not have been specified upfront in that way.

In this case the project owner as customer of the overall project management might not be the same, compared to the end user as the customer the Agile team is talking to. In the first case, it is the project owner who ordered the project and who will pay for the overall project and therefore as part of it for the Agile project as well. The Agile project usually wants to talk to the future end users of their software as their understanding of a customer, to find out about their users needs.

Business Studies vs. Agile Software Development

Traditional project management is a discipline of business studies, whereas the Agile project uses the tools of software engineering. Project management standards do not have a specific domain, it is a domain on its own. Within those standards, end user satisfaction is not listed as one of the core values. By the project management standards understanding customer satisfaction is something to be achieved, when the project has been successfully executed. Success in this case means, that a certain scope has been implemented within a given timeframe and a given budget.

Agile software development approaches, are putting the end user as customer in the middle of their approach. They are agnostic to the commercial implications of a project, which can lead to Agile projects focussing on addressing the customer needs, but about this forgetting the projects time and budget constraints.

3.2.5 Short-Term Solution

Define and grant clearance to the Agile project, to decide on certain change requests on their own, thus giving them leeway to act autonomously and address the different customers needs.

In detail: The Agile project together with overall project management identifies project invariants, hence mile stones and artifacts which are mandatory to the Agile project. Both together check their understanding, what kind of changes are considered neutral to effort and the project invariants and therefore could be decided upon in the Agile project. The overall project management defines their needs of information and documentation and marks exception to the aforementioned rule, representing the project owner. This way the Agile team knows about the level of clearance it has, when it comes to change requests and how they keep the overall project management informed.

Upon change requests coming from the end user representative, the Agile project checks if the change can be realized neutral to effort and project invariants. If a change request is neutral to effort and the project invariants, the Agile project lets his customer which is the end user representative officially approve the change to priorities or scope.

The Agile project communicates and documents this change to overall project management and works on it. This way transparency is assured and the overall project management has a chance to keep track and to intervene, if they see an exception to the clearance they negotiated with the Agile project.

If the Agile project team cannot decide about a change request, possibly due to the effort, impact on the schedule, or its special nature, it takes it directly to overall project management.

Overall project management can use the end users feedback in review meetings of the Agile project as an additional measure for overall customers satisfaction.

3.2.6 Forces (Continuation)

Manage Change vs. Responding to Change and Customer Collaboration

This is a conflict which mainly arises out of a different understanding of customer satisfaction and overall project priorities. In the case of Agile approaches, the customer collaboration and the willingness to respond to change are in the very core of the Agile value system [6]. On the other hand „classical“ phase oriented project management standards do not even mention customer satisfaction in their corresponding values system [11, 12] and focus on planability, economic and legal aspects. So very basic assumptions are in conflict: Customer satisfaction to be achieved by delivering software of optimal usefulness on the Agile teams point of view opposites customer satisfaction by delivering in time and budget as the top priority. In the first case the end user is recognized as the customer and in the second the project owner. This different understanding who the customer is, leads to different assumption how customer satisfaction can be achieved.

To address this, we have to raise the understanding on the Agile project about the commercial implications of a project. On the overall projects we need to familiarize project management with the value of „Responding to Change“ and the importance of end user collaboration to successfully deliver software as a result of the Agile project. We have to show, that both worlds can be integrated. Both parties need to understand, that there is more than one understanding what customer satisfaction might be and in this case both of them mutually need to be understood as important.

3.2.7 Long-Term Solution

To create a shared understanding of customer satisfaction the Agile team can define its "customer perspective", which identifies the different customers affecting the Agile project and capturing their understanding of customer satisfaction. This „customer perspective“ needs to be shared, discussed, and completed with the Agile projects environment, especially the overall project management. E.g. the overall project manager in a way is a direct customer of the Agile team, having time, cost, and an initial scope in mind thus he/she is a proxy of his/hers clients, paying for the overall project, making the project owner an indirect customer of the Agile team. Very often, the Agile team only will have limited exposure to the project owner, since he very often only can

contribute very little to the Agile project itself, but they will interact with future end users of the Software to be build.

Working on the short-term solution helps to identify and to understand the different types of customers and to differentiate from end users and other stakeholders, which interests need to be addressed by the Agile project as well.

In principle the „customer perspective“ provides a birds eye view on the Agile projects customer organization needs from an inside and outside perspective. The commercial implications are being addressed, by making the overall project needs against the project transparent.

To identify the different customers of the Agile project, the interactions and interferences with the Agile projects environment need to be captured. This can refer to „hard“ requirements towards an Agile project deliverable or more general expectations e.g. in terms of schedule or budget. Having identified the interfaces of the Agile project, identify the driving persons hence the stakeholders behind the interfaces or in other words the Agile projects customers. Next to their explicit interests, their overall motivation might from interest, e.g. „better usability“, „saving cost“ and should be captured as well if possible. Addressing the different understanding of „customer satisfaction“, techniques coming from stakeholder management can be used. Within this the Agile team needs to identify their customer.

For this customer perspective the customers can be categorized according the needs to be addressed best with constant interaction, feedback, etc. like usability of software being developed. Other customer might only need updates but not constant interaction, e.g. schedule or budget aspects. These categorizations can be organized, e.g. in a kind of responsibility assignment matrix. The ideal customer might be visualized for better understanding, how many different roles and needs are part of it.

Having this, the Agile team needs to synchronize their understanding with the overall project managements view on their stakeholders hence customers. Together they need to identify and resolve deviations and identify the Agile projects customers from each point of view. The result might be, that there are more then one stakeholder, forming „customer perspective“. Coming from this we need to understand what the agile project needs to deliver and to ask for to gain acceptance with the different representatives as part of the „customer perspective“.

All this different views are forming the customer perspective, which could be represented using a typical Agile role like the Scrum Product Owner [9].

Therefore the Agile team has a representation not only of the functional requirements coming from their end user but of overall project requirements like cost and time too. Within the product owner, each individual customer needs to see his/her needs against the Agile project being reflected, whilst on the other hand being able to understand, what else has to be addressed by the Agile project. E.g. the need for optimal software as a deliverable and therefore needing the constant interaction with this part of the „ideal“ customer can be shown to the overall project management.

The commercial implications become transparent to the Agile team by understanding the needs e.g. of the overall project management and indirectly of his/her customers. On the other a overall project manager has the chance to understand the need for tight collaboration, if he understands the motivation of one facet of the shared customer perspective.

3.2.8 Example Resolved

In this case an second senior manager, convinced of this approach, intervened and helped the Agile team to define the level of clearance they have to decide upon changes. They agreed, to let them free hand in effort neutral changes, with the aim to optimize user experience and user acceptance of the new user interface. Management should be involved only if changes would endanger schedule or change the functionality. The product owner created its customer perspective capturing his customers and not only interacted with the department, but kept the initial project sponsors in loop as well.

In order to get a buy in of the first senior manager, the Agile team agreed to provide some measures regarding changes. They tracked the total number of change requests and the figures of change requests accepted, rejected or escalated, as well as qualified statements of users, about the quality of the user interface to come. It turned out, that a huge amount of changes were accepted and realized, without changing the goal or timeline.

Finally senior management and the board of the company, encouraged by the user demand and feedback, decided to change priorities of user stories and to pre release a lean version of the user interface. The goal then was to start the sales cycle on short term notice. This was important, since some of their customers wanted to use the product by the end of the year, to spend remaining budget. The customers departments were waiting for an updated user interface for quite a while anyways.

3.2.9 Consequences

Benefits: This way we can merge the commercial and the software engineering perspective of customer satisfaction, using the different views to our advantage, to increase product quality and to help an Agile project team to create value.

Liabilities: It is difficult to find the right level, when looking for approval of a change request and to make commercial goals understood. If the level of approval is chosen to low, self organization ends. If it is chosen to high, an Agile team might lose itself in perfection, in the aim to satisfy their customer but ignoring the commercial interests of their own organization.

Having more then one person representing different customer interest would cause more overhead and effort and might even slow down the project if they would interact directly with the project.

4. Summary and Next Steps

The described approach addresses value conflicts on three level: Firstly in discovering and understanding the root causes of the conflicts experienced. Understanding the underlying conflict enables us to adapt our actions accordingly. On a second level, it provides immediate solutions, without addressing the root cause, but with the root cause in mind to provide short term improvements. Finally if at all possible I am describing a way to reach a sustainable solution.

Value based conflicts are challenging. Working on them means to work on changing the attitude of individuals and organizations. This is not always possible and usually it requires a sophisticated change management. Addressing the most obvious conflicts, e.g. conflicts over resources only provides temporary solutions. Working on changing the project environment is usually beyond the scope of a simple Agile project. Still, even if we cannot change the environment, we can try to influence it, at least we can take the conflicting values into account.

In future work I will investigate, how to address non constructive solvable value conflicts. Furthermore more pattern candidates need to be evaluated. Additionally I will merge my patterns found into a pattern language.

Finally I will am researching on additional value based conflicts. For this any feedback with regards to the validity of the pattern presented, your own experiences, or known uses of the pattern you might have encountered.

5. Acknowledgements

First of all I would like to thank my shepherd Jason Yip for the great support. Thanks, to my PLoP 2011 writing group, headed by Richard P. Gabriel, for their very valuable feedback and a great discussion.

6. References

- [1] S. Ambler, 1998. Process Pattern. *Cambridge University Press*.
- [2] K. Beck, C. Anders, 2005. Extreme Programming Explained, 2nd Edition. *Addison-Wesley*.
- [3] J. Coplien, N. Harrison. Organizational Patterns of Agile Software Development. *Prentice Hall*. pp.311.
- [4] J. Coplien, N. Harrison, 2004. Organizational Patterns of Agile Software Development. *Prentice Hall*.
- [5] DIN 69901-5 Projektmanagement, 1.2009. Projektmanagementsysteme; Begriffe. *Deutsches Institut für Normung e. V.* (in german)

- [6] M. Fowler, J. Highsmith, 9(8), 2001. *The Agile Manifesto*. Software Development
- [7] J. Pechau, Value based Conflicts of Agile Software Development. In *EuroPLoP 2011 Proceedings* (to be published)
- [8] K. Schwaber, M. Beedle, 2001. Agile Software Development with Scrum. Prentice Hall
- [9] K. Schwaber, 2004. Agile Project Management with Scrum. *Microsoft Press*, 53-66
- [10] D. West, T. Grant, M. Gerush, D. D’Silva, 2010. Agile Development: Mainstream Adoption Has Changed Agility. Forrester Research.
- [11] A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Third Edition, 2004. *Project Management Institute*.
- [12] Managing Successful Projects with PRINCE2, 2005. *TSO*.

7. Copyright Notices

Figure 1: “Brainstorm at INDEX: Views” by Jacob Boetter (@boetter - flickr) under a Creative Commons License (CC BY 2.0)

Figure 2: “Used Punchcard” by Pete Birkinshaw (BinaryApe – flickr) under a Creative Commons License (CC BY 2.0)