Tear Down the Barriers: A first report into the relationship between people and agile software development.

Steve Adolph, Gagan Dhillon, Philippe Kruchten
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, B.C.
Canada

stevea@ece.ubc.ca, gagandhillon@hotmail.com, pbk@ece.ubc.ca

Abstract

This paper describes the first step of a qualitative investigation into the determinants for effectiveness of agile software development. During a workshop at Agile 2006 21 participants were queried for what they liked or disliked about agile software development in their own personal experience. This data was then analyzed to begin developing a Grounded Theory of how an agile methodology contributes to the effectiveness of a software development team. Our early results show barriers erected as a result of both individual and organizational deficiencies block feedback and threaten the success of an agile software development project. Further research is required to understand these barriers and how agile methodologies reduce these barriers.

1 Introduction

"Agile is for people" write Cockburn and Highsmith[1] and agile development is certainly capturing the imaginations of people. But if agile is for people, then what about the people? What are the determinants, the specific factors that make agile work and make a contribution to the effectiveness of a team? We almost take it for granted that people variables such as management support, training and coaching, are determinants to the success to the adoption of agile software development, but what contribution does an agile methodology make to the effectiveness of a software development team?

This paper describes early results in a new research project investigating the contribution agile software development methodologies make towards the effectiveness of a software development team. The results reported in this paper are based on data collected by Scott Duncan during an Agile 2006 workshop. This paper presents an initial hypothesis and concludes by describing the next steps in our research.

2 Our Research Methodology

Our interest is in building a theory that explains phenomena observed in agile software development projects. We are specifically interested in understanding the contribution the use of an agile software development methodology make towards the effectiveness of a software development team. Grounded Theory[2] is a systematic qualitative method for explaining phenomena observed in social processes such as software development and constructs theories that are supported, or "grounded" in the data. This is a different approach than other research methods where hypothesis testing or theory proving is the objective.

Grounded theories are constructed iteratively, data is collected, coded, and categorized. A theory begins to emerge from the categories. Through a process of theoretical sampling, more data is collected to develop the properties of the categories. This process continues until no new properties emerge. The category is said to be saturated. This paper takes us through the first iteration from which a theory about agile software development begins to emerge.

3 Our Initial Data Set

The data set analyzed during this first investigation is a list of opinions of active industry practitioners. Our initial data set was collected by Scott Duncan during an Agile 2006 workshop at in Minneapolis and who gave us his permission to analyze the data. Twenty one people attended the workshop all claiming experience trying to implement agile methods. These opinions were collected anonymously from the respective individuals, that is, without being associated with the name of the individual or his/her company's identity.

This ensures there is no ethical issue regarding the collection and use of the data.

During the workshop participants were asked to write down how they felt about a variety of issues both pro and con from the customer, developer, and support perspective, regarding their experiences during the introduction of agile methods into "traditional" environments. This data represents the verbatim comments of the workshop participants including the perception of the experience pro(+) or con(-) and categorization of the experience as either support (S), developer (D) or customer (C). The group then divided into S, C and D groups and further combined their feedback, eliminating duplicates. One hundred and ninety two comments were collected.

The opinions have been organized into different categories as follows:

- **D+**: Positive comments from a software developer's point of view.
- **D-**: Negative comments from a software developer's point of view.
- **C+**: Positive comments from a customer's point of view.
- **C-**: Negative comments from a customer's point of view.
- **S+**: Positive comments from a support point of view.
- **S-**: Negative comments from a support point of view.

Examples:

D+	Developers really liked "minimum necessary ceremony" (e.g., skip huge formal doc for lighter techniques)
D-	Coach permitted damaging acts of heroism under pressure from customer.
C+	Real production ready code in each iteration. Customers could see a real product far ahead of release date.
C-	Difficult to get customers to write tests, as they have "real" work to do.
S+	"This is the first time anyone asked our opinion before the product was shipped."
S-	Development did not create any documentation or diagrams, so support had difficulty understanding the new product release features

3.1 Data Analysis

3.1.1 Qualrus: A Tool to Support Analysis of Qualitative Data

Qualrus[1] was chosen as the tool for coding and analyzing the data set. Qualrus is not an automated theory builder, rather, it reduces the tedious physical cut and paste effort of coding.

3.1.2 Data Immersion

Data immersion is a fancy way of saying we read and studied the data. Before commencing coding, we simply read, and re-read the data to get a feel for what is being expressed by the comments.

3.1.3 Open Coding

After reading through the data we began the process of open coding, assigning one or more labels to the participant comments that captures the essence of the concepts expressed by the statement. From our immersion into the data we developed some preconceived ideas for labels.

D-	Majority of IT staunchly supports waterfall	• Resistance to Change

3.1.4 Axial Coding

After initial coding we organized our resulting codes into categories and sub-categories. This is done in Qualrus by defining categories as codes and linking them together by the means of a single link. For example, the codes "Lack of Commitment" and "Lack of Discipline" are both examples of what we called "Individual Deficiency". Therefore we define individual deficiency (a category) as a code linked to both the codes using Qualrus's "is-a" link.

Individual Deficiency
• Arrogance • Looking Inwards • Lack of Commitment • Lack of discipline • Inability to adapt • Lack of confidence • Lack of experience • Lack of training

[1] www.qualrus.com

3.1.5 Selective Coding and Emergence of an Initial Hypothesis

The final stage is developing a hypothesis from the major categories constructed during axial coding. The foundation for the development of the hypothesis lays in the links between these major categories. Through this process a hypothesis emerges which is rooted or "grounded" in the data.

The following are the relationships between the major categories:

- Deficiencies **build** Barriers
- Deficiencies **block** Adaptation
- Blocked adaptation **leads to** Ignoring of feedback
- Barrier **leads to** Ignoring of feedback
- Ignoring of feedback **leads to** Failure

Summarizing the relationships depicted between these categories gives us the hypothesis embedded in the data. Upon further analysis of the above picture, we can word the hypothesis as follows:

> ***"Individual, team and organizational deficiencies build barriers and block adaptation. This further leads to ignoring of feedback which might result in failure of agile software development methods."***

This is not a daring hypothesis because in any social process we can easily imagine how the construction of barriers impedes feedback. This result does not challenge our perceptions and beliefs regarding agile software development. However what is interesting are some of the comments in our data set that suggest an agile methodology may contribute to the erection of barriers.

D- Obsession with some "Extreme Normal Form" in code lead to loss of velocity.

D- 52 one week iterations was too much!!

D- We got our iterations down to the point of successful functional improvements each time, but we forgot to look at the big picture, and hit a bottleneck that was impassable as a result

How a software development methodology contributes to the erection and removal of barriers is a phenomena warranting further investigation. Barrier is a category that warrants further data collection to elaborate and refine our emerging theory.

4 Discussion

This project successfully developed a hypothesis embedded in the data by applying qualitative research methods to our data by using Qualrus. However, is this hypothesis valid? Our intuition and beliefs about agile software development may incline us to answer yes because the results we found appear to support our beliefs but that does not make this result valid. From a Grounded Theory point of view, we have only analyzed one set of data and we have not yet compared new data to this hypothesis to see if it explains newly observed phenomena. Developing the properties of the Barriers category makes a case for further data collection and analysis by developing a set of directed questions develop the hypothesis emerging from this data. This would provide more confidence in the correctness of the hypothesis and eventually go a long way in helping individuals apply agile development methods.

The next step in this research is to enter a second data set Scott Duncan collected at XP2006 in Oulu Finland. Following the analysis of this data set we will acquire more field data through a pilot study of a group which has successfully completed an e-commerce project using an agile software development methodology. This step will involve both individual interviews of participants drawn from an agile project and a focus group. Personal field notes, observations, and retrospective data shall also be included.

5 Summary

This paper described the first step in a research project to understand the contribution agile methodologies make towards the effectiveness of a software development team. While the primary purpose of this first step was to familiarize ourselves with the research methodologies and tools that may be used in this project, we have found a result that is consistent with our view of agile software development. We also discovered the Barrier category which we intend to further develop.

6 Acknowledgements

The authors gratefully acknowledge the contribution made by Scott Duncan of the The Westfall Team in Ellerslie, GA

7 References

[1] A. Cockburn and J. Highsmith, "Agile software development, the people factor," *Computer*, vol. 34, pp. 131-133, 2001.

[2] A. Strauss and J. Corbin, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Second Edition*: Sage Publications 1998.