Method Tailoring as Negotiation

Fredrik Karlsson [1]

[1] Dept. of Informatics (ESI), Methodology Exploration Lab
Örebro University, SE-701 82 Örebro, Sweden
fredrik.karlsson@esi.oru.se

Abstract. The need for method tailoring is widely accepted in the field of information systems development methods. Today much attention has been devoted to viewing method tailoring either as (a) a highly rational process with the method engineer as the driver where the method users are passive information providers, or (b) as an unstructured process where the developer makes individual choices, a selection process without a driver. In this paper we view method tailoring from a negotiation perspective using Actor Network Theory. Our narrative examples depict method tailoring as a more complex process than either (a) or (b) show.

Keywords: Method tailoring, Method Engineering, Method-in-Action, Actor Network Theory, Negotiation

1 Introduction

As Fitzgerald et al. [1] conclude 'it is now widely accepted that [information systems development] methods should be tailored to the actual needs of the development context.' This statement is acknowledged by, what appears to be, the two schools of information systems development methods [2]: method engineering [3] and method-in-action [4].

These two schools view method tailoring as (a) a highly rational process with the method engineer as the driver, where the method users are passive information providers, or (b) as an unstructured process where the developer makes individual choices, a selection process without a driver. However, Riemenschneider and Hardgrave [5] show that acceptance of methods is dependent of 'the opinions of developers' coworkers and supervisors toward using the methodology.' Madsen et al. [6] conclude that far too 'little research has addressed the details how the unique and local method emerges and why it takes the form it does.' They unfold the emergent method through three different perspectives, one of them being the interactive process perspective. But, still the details of the negotiation aspect of the emergent method are not evident. Consequently, method tailoring is poorly understood as a social activity. The purpose of this paper is to exemplify method tailoring as negotiation, the interplay between humans and artifacts. For this purpose we apply Actor Network Theory (ANT) [7].

2 Actor Network Theory as Research Approach

The research presented in this paper is an interpretive investigation of method tailoring. The empirical base is an information systems development project undertaken in a public organization. A content management system was implemented and the organization's existing web site was migrated to the new platform. The author participated as one of thirteen regular team members in this project, spending 1 200 man hours during eighteen months. The project was carried out at three different locations, two locations inside the public organization and one location at a consulting firm. The actors are categorized according to the main roles they had during the project: project manager, systems administrator, implementer, requirements engineer and content manager.

Walsham [8] concludes that ANT is both a theory and a research method. ANT contains a conceptual framework to use during data collection and analysis. Our interpretation of this framework is inspired by Walsham [8] and our framework has a specific characteristic; it does not contain any a priori distinction between human and non-human actors. Both concepts are viewed as active makers of actor networks and are specializations of the actant concept. Furthermore, networks are changed through translation, an establishment of a new relation between actors. Latour [7] describes it as coexisting in a network to achieve of a common goal, for example when the system analyst and the tester agree to document the information system's external behavior using use cases. Often translation requires enrollment where an actor seeks to influence how another actor should act, for example to use a specific technique. Enrollment and translation can result in inscriptions, where interests are inscribed into written material or technical systems.

This research is based on several data sources from the systems development project: intermediate project artifacts, e-mails and project notes. Intermediate project artifacts show what has actually been documented. Furthermore, these artifacts are time stamped making it possible to analyze how they have evolved over time. That is to say, they show the result of method tailoring. The e-mails and the project notes are used to capture tailoring decisions and arguments behind these decisions. Consequently, these documents contain traces of the team members' different viewpoints about the emergent method.

3 Method Tailoring – The Analysis

3.1 The First Example

Our first examples concerns method support for requirements engineering. Much of the requirement work during this project concerned the web page templates. Simple sketches were used to capture layout and functional requirements. This work was carried out by the requirements engineer together with the content managers. However, when it came to more advanced web page templates, containing interaction

possibilities, static sketches did not provide enough information. The method's inadequacy is salient in the e-mail conversation between the requirements engineer and the implementers, concerning the design of forum templates: 'what are the options in this listbox', 'how are these [web] pages linked to each other', and 'how shall we display the thread overview.' These quotes illustrate the problems to capture and communicate the design.

In an e-mail the requirements engineer concludes 'that several options exist', but she suggests the use of either use cases or storyboards. Both her suggestions are enrollments of techniques used in other methods. In addition, through her proposal she tried to enroll the content managers and the implementers in use of one of these techniques. Two of the implementers express that they prefer storyboards to use cases: 'use cases tend to become cluttered … difficult to show how [web] pages are related.' Consequently, the two implementers seem to be concerned with the amount and the type of details that are capture if they chose to use cases. Furthermore, the implementers needed to know how web pages were related to each other in order to determine possible navigation paths and when to provide a specified functionality. According to the implementers use cases insufficient means to this end.

In an additional e-mail to the implementers, the requirements engineer referred to a discussion with some of the content managers (it is unclear with whom). She stated that the content managers preferred storyboards as well, 'since they are easier [to read].' Hence, the three actor groups have made a translation and later use of documents show an inscription in the method.

3.2 The Second Example

Our second example concerns method support for testing. The content managers, who are responsible for testing the web page templates, had just begun their work. They either gave oral reports to the implementers or documented the flaws on post-it notes. At initial stage the implementers concluded that the method lacked proper support for test reports and tracing flaws.

One of the implementers addressed this issue with the content managers via e-mail. He argued in favor of one shared artifact for documented bugs, since 'I believe we cannot keep trace of all the bugs we have found.' The content managers' answers to this e-mail can be divided as follows: (1) two actors did acknowledge the problem (2) one actor did not acknowledge this as a problem, and (3) two actors did not answer. In an e-mail reply to the content managers the implementer proposes 'a simple Excel sheet … on a shared domain.' The implementer received three positive replies to this enrollment. In one reply we find '… we have to discuss the layout [of this document].' The person who did not acknowledge the need for a formal test report document did not answer the implementer's e-mail.

The implementers discussed the need for a formal test report document with the project manager, arguing that they were not able to manage the change requests with the current way of working. Hence, they enrolled him in the method tailoring process. At this meeting the implementers presented a document template, which was later e-mailed to the content managers. The e-mail conversation shows a translation between

the implementers and the project manager, where the latter act as delegate: 'I believe this [document layout] looks good.'

At a later meeting a decision was taken to use this document template. This decision was explicitly supported by the implementers as well as two of the content managers. Consequently, an inscription in the method was made. When analyzing the use of the template we identified that two of the content managers disagreed with the decision and the inscription. They continued to report bugs via post-it notes, e-mail and orally. The other actors did not express any problems with the new modifications. Hence, they aligned with the method evolvement.

4 Conclusion

In this paper we have depicted method tailoring from a negotiation perspective, using Actor Network Theory. Most of the existing literature view method tailoring as either (a) a highly rational process with the method engineer as the driver where the method users are passive information providers, or (b) as an unstructured process where the developer makes individual choices, a selection process without a driver. Our narratives show that method users are not passive information providers during method tailoring. But these narratives also show that method tailoring is not about individual choices either. Accordingly, this is clearly an interesting venue for further research, investigating negotiation patterns in method tailoring and how they can be used in construction of approaches and tools.

References

1. Fitzgerald, B., Russo, N.L.,O'Kane, T.: Software Development Method Tailoring at Motorola. Communications of the ACM 46(4), 65-70 (2003)
2. Ågerfalk, P.J.,Fitzgerald, B.: Exploring the Concept of Method Rationale: A Conceptual Tool for Method Tailoring. In: Siau, K. (ed.) Advanced Topics in Database Research, pp. 63-78. PA: Idea Group, Hershey (2006)
3. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and Software Technology 38(4), 275–280 (1996)
4. Fitzgerald, B., Russo, N.L.,Stolterman, E.: Information Systems Development - Methods in Action. McGraw-Hill, London (2002)
5. Riemenschneider, C.K., Hardgrave, B.C.,Davis, F.D.: Explaining software developer acceptance of methodologies: A comparison of five theoretical models. IEEE Transactions on Software Engineering 28(12), 1135–1145 (2002)
6. Madsen, S., Kautz, K.,Vidgen, R.: A framework for understanding how a unique and local IS development method emerges in practice. European Journal of Information Systems 15(2), 225-238 (2006)
7. Latour, B.: Reassembling the social: an introduction to actor-network-theory. Clarendon lectures in management studies Oxford University Press, Oxford (2007)
8. Walsham, G.: Actor-Network Theory and IS Research: Current Status and Future Prospects. In: The IFIP TC8 WG 8.2 international conference on Information systems and qualitative research, Chapman & Hall, New York (1997)