Organisational changes in migration to agile development strategies

A review of

“Challenges of migrating to agile methodologies”
Sridhar Nerur, Radha Kanta Mahapatra, George Mangalaraj in
Communications of the ACM, 2005, Vol 48 issue 5, pp 72 – 78

Leon Derczynski – aca00lad@shef.ac.uk
000174828

14/02/2006

Abstract

The adoption of agile development methodologies is progressing gradually but certainly in the commercial world. The first step to adopting agile development in an existing organisation is to migrate from a traditional development strategy to an agile one. This problem has been well explored in the past decade, and development teams wishing to try agile methods have a wealth of literature available.

The culture suggested by agile methodologies is still different from that typical across many organisations. Wider changes in an organisation may be required outside of the development team. As a result, the management structure of the entire organisation needs to be made aware of this potential for change and adapt to it, in order to achieve a smooth transition

Contents

Introduction

Changing technologies and new user demands require adaptive processes. Agile development methodologies are those that adapt to changing requirements and knowledge. These techniques have benefits in that they are designed to cope with constantly changing requirements, as opposed to traditional methodologies that assume requirements are fixed. This assumption leads to any requirement changes being very costly in traditional development.

Development occurs over time, and organizations are constantly evolving. This evolution leads to changes in requirements during the development process. Therefore, there is a need for development processes that recognise this change. The entire organisation may be affected by migrating to an agile development methodology, and certainly it may require some culture change, be it localised or global.

Literature exists on migrating from traditional to agile development but only for developers or perhaps development team managers. Differences in the implementation of agile methods in an organisation when compared to traditional methods are significant. Below, we will examine how Nerur, Mahapatra and Mangalaraj define the differences and their identification of issues in migrating to agile methods. Their paper is explicitly presented from an organisational and management point of view, and not that of a member of the development team.

Agile technologies

What are agile development methodologies?

Agile methodologies should cope with ever-changing requirements. Requirements need to be set for each part of a development process, so that people know what to deliver.

A phase of development can be known as an iteration. Usually the highest-priority tasks are decided at the beginning of each iteration (sometimes these are called stories, depending on which development methodology is used) and then delivered by the end of that iteration. Where traditional development has iteration lengths of anywhere from 3 to 12 months, the short iteration length used in agile methods (1-4 weeks) helps them adapt to any changes.

Very little formal documentation is required by agile development methodologies. Rather, knowledge about the project is kept in people's heads. The current state of affairs may exist in the form of a set of story cards on a notice board, or some diagrams on a whiteboard, but there are no extensive manuals detailing the behaviour of each class and so forth. The general philosophy is that projects and their code should be self documenting. A coding standard including verbose comments is often suggested.

The use of test driven development also helps document code, in that it declares how a module should behave. The tests are written from a specification, which serves as initially documenting what is required. The test 'suite' is run all the time, to ensure that everything works as intended after each change. This in turn allows for increased confidence in the code, as it can be shown that the system works according to an agreed specification. Test also stay permanently (unless the specification changes) and so will serve to detect a potential bug for the lifetime of the system.

The lack of a formalised central role removes manager's ability to assess progress. XP [Beck 2004] defines a coach, who has some ability to manage the team, though tends not to force team into a direction, rather just keep up to date on progress and planning. This role certainly has no absolute authority.

Agile methods develop a feature, chosen by the development team based on its priority. Traditional development methods define a set of processes to be used that should be followed in strict order, including the creation of many artefacts.

What is the case for adopting agile development technologies?

If requirements change during delivery, then by the time delivery occurs, the item produced will no longer match up with the requirements. Development is a time-consuming process, and requirements change over time; thus, a method that fails to take change into account is not an accurate match.

Changing marketplace alters the way that businesses provide their products. This leads to constant organisational change. An inability to keep up with change will lead to lost revenue and reduced effectiveness.

Traditional development has long iterations. If requirements change part way through developments, the iteration must be restarted. This is a massive loss in the case of traditional development. With agile technologies, this loss is vastly reduced by the shortened iterations – less time is lost before changing tack to take new requirements into account.

Development of the paper

Objectives

Despite having no clearly defined abstract or introduction, from the early part of the text, the aims here are:

- To present the impact of agile development methodologies on the structure of an organisation
- To compare agile and traditional development from an organisational and managerial viewpoint

Claims

The authors make a set of claims in the body of their paper. All are thoroughly backed up by referenced material – there is very little new data presented, and the most part of the paper is a summarisation of ideas from a specifically organisational point of view.

(a). Agile methodologies require a change in management style, from command-and-control to a more collaborative environment. The command and control structure places individuals in fixed roles in a firm hierarchy. Conversely, agile development methodologies typically allow assignment of more than one role to a person, and for people to move around. There is no central manager who is aware of everybody's work at a high level and determines which activities to undertake. Rather, these things are kept track of via tacit communication and artefacts (depending on the exact development method used).

(b). Agile development moves power to the development teams. As much project-critical information is tacit, and mentally retained by the team's members, but not explicitly documented, it is harder for people outside the team to access it. This can create a dependency on the development team for progress reports and so forth. It also removes the ability to assess progress and decision-critical information from the hands of managers and places it into the development team.

(c). Agile development risks uncomfortable comparisons with traditional development. Agile development requires competent staff and a high skill level to work successfully [Boehm 2002, Boehm 2004]. This higher standard immediately reduces the number of candidate staff eligible for agile projects. Further, staff working on traditional development methods may feel left out when others are selected for projects using agile methodologies.

(d). Decision making is harder with agile development due to a more diverse environment. Instead of having centralised control, the development team and customer representative make decisions. For example, under XP, instead of being assigned tasks, developers estimate them and then elect which ones to do. With such a wide group, everyone involved in decisions will have a different agenda and set of priorities. This could lead to common ground becoming difficult to identify and a lack of unified direction.

(e). Cooperative customers are required for agile development. The idea of having a customer on the development team creates ample opportunity for verifying requirements and rapidly getting answers to questions. The downside is that a customer representative needs to be found that fits in with agile methodology ideals – that is, somebody that's prepared to work on a level toward the same goals (collaborative), whose agenda matches that of all customer stakeholders in the project (representative), who can legitimately and accurately speak on behalf of all stakeholders and make decisions (authorized), who will see the project through to the end (committed), and who has enough knowledge about their organisation and requirements to be able to answer questions (knowledgeable). This is a fairly stringent set of attributes for a customer to have, and such a person may be too much of an investment for the client.

(f). Migration from traditional to agile methodologies requires investment in procedures, structures and communications. Any change in process will require some overhead. The difference between agile and traditional methodologies is large, and so requires a significant overhead.

(g). Investment is required in new object oriented tools. Traditional technologies will work with a number of programming paradigms. However, agile methodologies strongly favour object oriented languages. As a result, migrating to agile development may bring with it the cost of moving to a new development platform – this would include new development tools and staff training. Agile development also stipulates practises such as unit testing and version control; there will again be a cost and culture change associated with introducing these factors to the development environment.

Current literature

Literature on the topic of agile methodologies and introducing them into an organisation is cited by the authors. They mention Boehm’s key works which seem to be widely regarded as authoritative in this field, which in itself is fairly narrow. The total range of cited documents remains small.

Notably, there is only one citation of a practical experiment that provides results. All the others are presentations of concepts regarding agile development and introductions to the topics, from for example books. This is reflective of the available literature in the field and so not a huge fault with the paper; it simply presents known issues with migrating to agile methodologies, and does not attempt to discover new ones or assess the quality of previous studies.

Older literature [Larman, Newkirk] is available that covers many of the issues presented by the authors, although this has not been drawn on. This literature covers the introduction of agile methodologies to organisations, from a management perspective, sometimes in very granular detail.

This older literature offers methods for migration and solutions to some of the problems mentioned in the authors’ work, instead of simply highlighting the potential scope for problems.

Some current literature is discussed, mainly from the managerial / organisation viewpoint that is being used. There is further literature available on the topic.

Assumptions and basic description

Most of the points made in the paper draw very heavily on the references and are often direct quotations. The descriptions of the problem and background are accurate and concise.

Overall view

Migrating to agile methodologies is costly from a managerial and organisational point of view. This concept has been dissected at a surface level and plenty of pointers for further investigation have been provided. These pointers, rather than further reading, are details of where problems and unforeseen issues may occur. It remains the reader’s responsibility to examine their own organisation with these high-level pointers at mind and create their own process for managing a migration.

Evaluation

The conclusions are backed up with references, though no entirely new concepts are presented. Rather, a collection and summation of existing work is put across, with a strong and concise message to take heed and be aware of costs when migrating from traditional to agile development methodologies.

The ideas given are convincing; they have a thorough grounding in previous work, and cite this heavily. The style of the paper suggests that it is not targeted at a reader with a strongly technical and development background, but rather at a manager, which is in fitting with the overall viewpoint declared at its start.

Some metrics regarding particular changes may help assess the impact of various changes. For example, KSLOC, productive hours, velocity during initial weeks, productivity / error rates of agile / traditional teams. Maybe the same metrics could be carefully monitored to discover the impact on a traditional development team when agile technology is introduced partway into an organisation.

Also, the examination or referencing of some case studies would provide valuable further reading, as well as back up the claims made. Case studies are available. As it stands, the paper offers suggestions for potential trouble but no more; examples of where others have fallen down – and how badly – would have much more value than a generalised warning of costs.

Also, examining generic work in how change affects organisations in order to identify issue specific to agile methodologies would back up some of the presented material. As it stands, the costs involved in any organisational change are mixed with those induced by migrating to agile methodologies. Once costs that are caused by any generic change are used, there may be some agile-specific costs and trouble spots visible.

References

Beck, K. *Extreme Programming Explained: Embrace Change*. Addison Wesley, 2004

Boehm, B. Get ready for agile methods, with care. *Computer* (Jan. 2002), 64-69

Boehm, B. and Turner, R. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley, Boston, MA. 2004

Larman, C. *Agile and Iterative development: A manager's guide*. Addison Wesley, 2003

Newkirk, J. Introduction to agile processes and extreme programming. *International conference on software engineering* (2002 Vol 24), pp 695-696