

## ISSUES AND CHALLENGES IN INTEGRATION OF AGILE PROCESSES TO TRADITIONAL SOFTWARE PROCESS MODELS

Harsh Taneja\*

Ashima Singh\*\*

---

### ABSTRACT

*Agile software development is finding a new dimension in software development process models. As tools and techniques used to develop software has been changed drastically. Software development processes need to keep pace with changing technologies and new demands from clients. Today's changing business needs have given rise to adaption of structures, strategies, and policies to suit the new environment by the organizations. These organizations need to upgrade their information systems to meet their changing requirements but the traditional and plan-driven software development methodologies lack the flexibility to adjust the development process as the process runs in the defined or standardized manner. Agile development method has the ability of overcoming the limitations of traditional approach. A number of software development methods such as extreme programming (XP), feature-driven development, scrum and adaptive software development are part of agile development method [3,7]. We cannot neglect traditional methods as they are the base of software development but they need to be altered with respect to the demands of agile software development process thus making software industry move towards agility.*

---

\*Department of Computer Science and Engineering, Thapar University, Patiala, Punjab.

\*\*Assistant Professor, Department of Computer Science and Engineering, Thapar University, Patiala, Punjab.

## INTRODUCTION

Traditional methodologies such as waterfall, prototype continue to dominate the development process, numerous surveys clearly demonstrate the growing popularity of agile development process. Thus creating a lot of confusion about the methodology to be used for software development in the organization. According to Boehm, “organizations must carefully evolve toward the best balance of agile and plan-driven methods that fits their situation”[2]. Organizations cannot ignore the agile process but for those using traditional development methodologies, adoption of agile methodologies will pose several challenges, since the two software development methodologies use completely different approach for software development.

### **Barriers in merging agile with traditional development of software**

Participants at the 2004 USC-CSE Annual Research Review have identified three categories of real and perceived barriers to implementing agile processes in traditional one[6] as:-

#### **1. Nonproblem**

Nonproblem indicates that either the perception about integrating agile development method with traditional methods is an illusion or that organizations can easily remove the issue without major changes to either the agile or traditional ways of software development. Some of them are:-

- a. Quality assurance system.
- b. Agile is inadequate for managing defects.
- c. Quantitative management
- d. Agile process is unmanaged.

#### **2. Problems only in size & scope**

- a. Configuration management.
- b. Asymmetry in focuses of both the processes as agile focuses on business value while traditional one on activities to be followed.
- c. Documentation planning in agile.
- d. Training the developer and tester in agile.
- e. Risk management difficulties due to lack of proper documentation in agile.

#### **3. Significant issues**

- a. Capital evaluation.
- b. Non functional requirements.
- c. Milestone review in agile.

- d. Integration with other methods.
- e. Roles and responsibilities.
- f. Problem of agile in legacy systems.
- g. Cost estimation.
- h. Maturity assessment.

### **Another view of problems in integrating agile with traditional methods**

#### **Development process problems**

Problem while developing software is how to merge agile processes with standard industrial processes without either killing agility or ignoring the traditional approach.

Some of such problems are:-

- Variability
- Difficulties of using agile in legacy systems
- Different life cycles
- Requirement analysis using both agile and traditional approach.

How to overcome these problems?

- Conduct a significant analysis of existing and proposed processes to identify mismatches in process requirements.
- Develop architectures that support agile and traditional teams. Identify an “agility level” characteristic for the architectural design. Look for areas where requirements will change rapidly or where the design approach is not clear, use these as agile opportunities [6].
- Build up processes rather than tailoring them. Look at the project’s needs and select only those process assets that seem important.
- Implement agile practices that support existing or new processes. Practices that support existing processes might include prioritizing requirements that is helpful in keeping on schedule when new requirements emerge.

#### **People Problems**

People issues are the most critical in improving management of engineering and development of skills of people involved. Addressing them is vital to the adoption and integration of agile methods and practices into the processes. Some of the problems related to people are:-

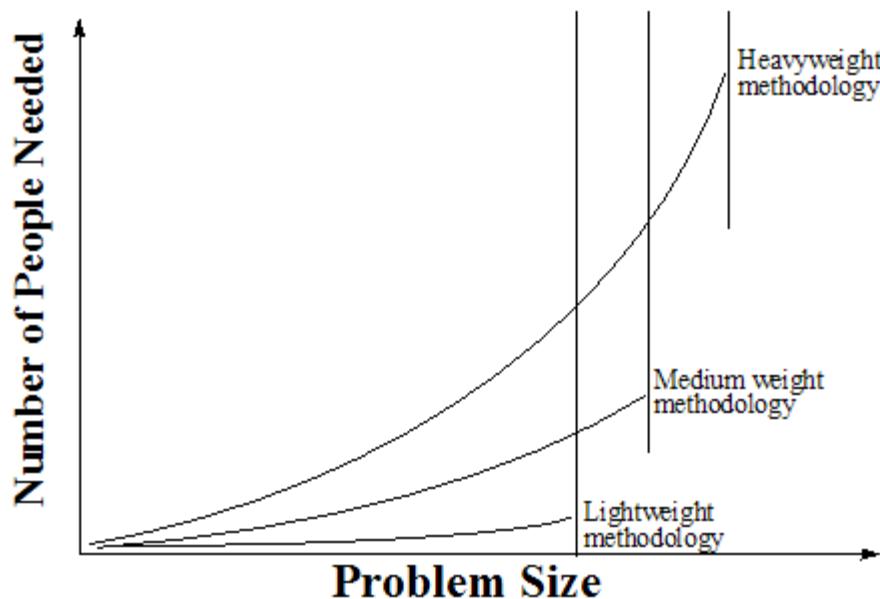
- Management issues
- Change management
- Handling main projects

How to overcome these problems?

- Translate agile and software issues into management and customer language. When we discuss technical issues with non technical people, issues must be described in terms that all the stakeholders can easily understand.
- Show the appreciation for the team's and individual work, regardless of the outcome.

Management and organizational
<ul style="list-style-type: none"> <li>• Organizational Culture</li> <li>• Management Style</li> <li>• Organizational Form</li> <li>• Management of Software Development Knowledge</li> <li>• Reward Systems</li> </ul>
People
<ul style="list-style-type: none"> <li>• Working effectively in a team</li> <li>• High level of competence</li> <li>• Customer relationships—commitment, knowledge, proximity, trust, respect</li> </ul>
Process
<ul style="list-style-type: none"> <li>• Change from process-centric to a feature-driven, people-centric approach</li> <li>• Short, iterative, test-driven development that emphasizes adaptability</li> <li>• Managing large, scalable projects</li> <li>• Selecting an appropriate agile method</li> </ul>
Technology (Tools and Techniques)
<ul style="list-style-type: none"> <li>• Appropriateness of existing technology and tools</li> <li>• New skill sets—refactoring, configuration management, JUnits</li> </ul>

**Table 1: Problems in integrating agility with traditional approach of software development[4]**



**Table2 : Problem Size and Methodology Affecting Staff needed[8]**

### Traditional vs Agile Methodology

Both the approaches can be compared with the help of a list of factors which can be listed as:-

#### 1. Fundamental Assumptions

Traditional systems are predictable and are built through extensive planning while continuous testing and coding is done based on feedback mainly from client in agile development process.

#### 2. Control based

Process centric control is there in traditional approach and in agile, we have people centric control.

#### 3. Role Assignment

Traditional process relies on specialization of the individual as compared to dynamic roles being preferred in agile methodology [12].

#### 4. Project Cycle

Project Cycle is guided by activities of the process to be followed in comparison to the product features being used by agile process.

#### 5. Management style

Command and control is the style used in traditional development process and in agile leadership along with collaboration is used as management style.

## 6. Communication process

We have to strictly follow formal communication method in traditional process but can go with informal way of communication in agile.

## 7. Development model

Any of the traditional methods such as prototype model, is taken while developing software using traditional approach as compared to the evolutionary-delivery model used in agile development.

Project Characteristics	Agile discriminator	Heavyweight Discriminator
<b>Primary objective</b>	Rapid Value	High Assurance
<b>Requirements</b>	Largely emergent, rapid change, unknown	Knowable early, largely stable
<b>Size</b>	Smaller teams and projects	Larger teams and projects
<b>Architecture</b>	Designed for current requirements	Designed for current and foreseeable requirements
<b>Planning and Control</b>	Internalized plans, qualitative control	Documented plans, quantitative control
<b>Customers</b>	Dedicated, knowledgeable, collaborated, collocated onsite customers	As needed customer interactions, focused on contract provisions
<b>Developers</b>	Agile, knowledgeable, collocated, and collaborative	Plan-oriented; adequate skills access to external knowledge
<b>Refactoring</b>	Inexpensive	Expensive
<b>Risks</b>	Unknown risks, Major Impact	Well understood risks, Minor impact

**Table 3: Agile and Heavyweight Discriminators [8]**

## CONCLUSION

This paper tries to find the features that bridge the gap between traditional process models and agile process. The opportunities and benefits that agile methodologies afford make them attractive but organizations should well consider integrating them with existing practices and processes [4]. Tools play a critical role in successful implementation of any software development methodology. Organizations planning to adopt agile methodologies have to

invest in tools that rapid iterative development, configuration management, refactoring, and other agile techniques [5]. But tools alone cannot make software development successful. Staff must be trained to use these tools efficiently. Measurement is important in agile methods. Organizations might need to reinterpret these measures to fit easily into existing traditional processes. With a bit of imagination, pair programming and test driven development, agile can be part of most traditional software development process.

## REFERENCES

1. Boehm, B. and Turner, R. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley, Boston, MA, 2004.
2. Boehm, B. Get ready for agile methods, with care. *Computer* (Jan. 2002), 64–69.
3. Cockburn, A. and Highsmith, J. Agile software development: The business of innovation. *IEEE Computer* (Sept. 2001), 120–122.
4. Nerur Sridhar, Mahapatra RadhaKanta and Mangalaraj George *Challenges of migrating to agile methodologies* (May, 2005)
5. A. Cockburn, *Agile Software Development*, Addison-Wesley, 2001.
6. Boehm, B. and Turner, R. *Management challenges to implementing agile processes in traditional development organizations*, IEEE Computer Society (Sept.-Oct. 2005)
7. S.W. Ambler, *Agile Modelling*, John Wiley & Sons, 2002.
8. Awad M.A *A Comparison between Agile and Traditional Software Development Methodologies*, School of Computer Science and software Engineering, The University of Western Australia,2005
9. M. Fowler and J. Highsmith, “The Agile Manifesto. Software Development Magazine” August 2001.
10. Pressman R. S., “Software Engineering”, 7th edition, McGraw Hill Education,2009.
11. L. Barnett, *Agile Survey Results: Solid Experience And Real Results Agile Journal*, 2006.
12. A. Mahanti, "Challenges in Enterprise Adoption of Agile Methods - A Survey", *Journal of Computing and Information Technology*, vol. 14, p. 10, 2006.
13. S. Nerur and V. Balijepally, "Theoretical reflections on agile development methodologies: The traditional goal of optimization and control is making way for learning and innovation", *Communications of the ACM*, 2007.