

# Web Applications Development and Software Process Improvement in Small Software Firms: a Review

Haroon Tarawneh  
Al-balqa Applied University  
[haroon@teacher.com](mailto:haroon@teacher.com)

Sattam Allahawiah  
Al-balqa Applied University  
[Sattam\\_amro@yahoo.com](mailto:Sattam_amro@yahoo.com)

## ABSTRACT

The web development industry worldwide is dominated by a myriad of small firms. This presents a challenge in terms of determining the current practices of industry participants, and in devising improvement initiatives which are feasible for small firms. Currently, the level of adoption of best practice among web developers is unknown. To help improve the web industry, it is necessary to determine the current status of use of practices and techniques. Furthermore, the effectiveness of assessment-based software process improvement for small firms needs to be evaluated.

In the last couple of years, one of the major trends for software development organizations was the move towards web application systems. Their processes usually are immature and ad-hoc. Often this is coupled with a less than positive attitude towards software engineering practices and, especially, software process improvement initiatives and software metrics collection. In particular, code metrics (such as lines of code, code complexity etc.) and process improvements standards (such as the Capability Maturity Model) are often viewed as obsolete and assessing processes and metrics.

KEYWORDS: Web applications, Extreme programming, Software process, Small firms.

## 1. Introduction

The models underlying traditional SPI programs such as software CMM were designed for large organizations undertaking extensive projects. However, while these models have been evolving over many years, the software industry has changed dramatically with a large increase in the proportion of small software development firms. To help in reducing the difficulty of building web-based applications we need a process model that describes the phases of web-based applications. A process model should help developers to address the complexities of web-based applications, minimize risks of development, deal with likelihood of change, and deliver the application quickly, while providing feedback for management as the project goes along.

But there is, as yet, little assistance from the research literature to be gained in addressing these issues.

Small software development firms recognize that software process assessments play a valuable role in improving a firm's processes and products, but most feel that SPI costs too much and takes up resources needed to deliver products. The main objective for this research is, how does our web application development framework deliver web-based applications and related software process improvement on time, within budget.

Many practitioners and researchers recommend that this problem can be addressed by the adoption of lightweight iterative and/or incremental approaches, such as extreme programming (XP) [1].

## **2. Web-based Applications Development Methodologies for Small Firms**

The literature showed different types of methodologies used to develop web based applications. Most of these methodologies required a lot of resources and need multi disciplinary development teams. According to Hofer [2], small software firms face the following problems:

- 1- Problems with customers.
- 2- Changing project goals and requirements.
- 3- Incomplete specifications.
- 4- Project management problems.
- 5- Staffing problems.
- 6- Lack of project control.
- 7- Lack of tools.
- 8- Lack of training.
- 9- Lack of a unique process.
- 10- Lack of methods.
- 11- Lower communication of team members.

Due to the previous problems, and to the literature analysis, small firms need a lightweight development method. The fittest development method to be used in these firms is agile development method. Agile development methods address the following characteristics [3]:

1. Modularity on development process level.
2. Iterative with short cycles enabling fast verifications and corrections.
3. Time-bound with iteration cycles from one to six weeks.
4. Parsimony in development process removes all unnecessary activities.
5. Adaptive with possible emergent new risks.
6. Incremental process approach that allows functioning application Building in small steps
7. Convergent (and incremental) approach minimizes the risks.
8. People-oriented, i.e. agile processes favor people over processes and technology.
9. Collaborative and communicative working style.

We have reviewed different agile methods like, scrum, crystal, feature driven development, adaptive software development, and extreme programming. Each of these methods has its weakness and strength. Extreme programming (XP) is a method that has evolved from the problems caused by the long development cycles of traditional development models [4].

The contribution of XP to software development is expressed, among other ways, in the quality improvement of both the entire process of software development and of the software quality itself. Currently, XP is used mainly in small-medium size software projects.

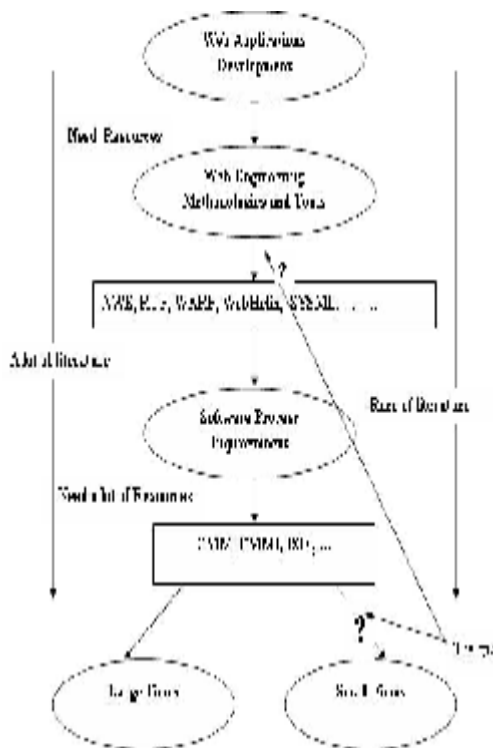
## **3. Software Process Models and Software Process Improvement**

One of the important objectives of this research is to investigate the software process frameworks for web-based applications development and its special features in the context of 'young' (start-up) small software companies, which is capable of being 'tailorable' to the particular stage of organizational development of small 'young' software companies.

The literature showed that the software process improvement in small organizations is a challenging task where the "smallness" brings a number of unique problems. The literature showed that there are several methods and tools exist for determining and improving the quality assurance function in software organizations. The most widely used are CMMI, ISO9001 and SPICE. However, small firms have difficulties in applying these models in their full extent [5,6].

So it is important to find a suitable process model that comply to software process improvement and to be capable of being 'tailorable' to any particular stage of organizational development of small software firms.

The literature showed that extreme programming is the lightweight process model that can help small firms in the implementation of software process improvement [7]. High software quality is one of the main principles that guide any software development methodology, and this is also correct with respect to XP. It is indeed important, to examine web engineering and software process improvement to determine what is already accomplished, what is not yet done, and how to fill the gap [8]. Figure (1) shows the gap.



**Figure 1.4:** Literature Related to Web Development and Software Process Improvement.

#### 4. Literature Analysis

In literature review we have investigated the major problems related to web based application development and software process improvement in small firms. Also we have investigated the current used methodologies for web based application development and their limitations. The literature highlighted the following points:

1- Web engineering is the process used to create high-quality web-based systems and applications that deliver a complex array of content and functionality to a broad population of end-users.

2- There are very few standard methods for the web developers to use. Hence, there is a strong need to understand and undertake Web Engineering.

3- Ad-hoc development of WBA has brought disasters to many organizations.

4- The difference between web engineering and software engineering.

5- The characteristics of Web engineering processes.

6- The lack of well-defined web development processes for the building of Web applications in small firms.

7- Theory-based and empirical work needs to be conducted because Web-based application development projects are growing rapidly.

8- Web development is seen primarily as an authoring problem rather than a software development problem to which well-established software engineering principles should apply.

9- Clearly, more research is needed in the area of web engineering.

10- The growth of the software industry has produced many small companies that do not do contract software, but rather compete in other areas.

11- Four significant development issues that have not been adequately addressed in software and web engineering literature: company size, development mode, development size, and development speed.

12- First step toward process improvement is identifying the strengths and weaknesses of an organization's software processes to determine effective improvement actions.

13- In adopting a software process model, many small software companies are ignoring standard process models and models for process improvement.

14- Software process improvement is required to increase the productivity of small software companies.

15- Software organizations of any size can advance their business by practicing a critical set of SPI elements.

## **5. XP and Software Process Improvement**

The literature showed that, different software development methodologies and quality assurance methods are used in order to attain high quality, reliable, and bug free software. Extreme Programming (XP) is a software development methodology that integrates many of the known ideas in order to achieve such software systems. Specifically, XP emphasizes code-unit testing (preferably before its writing), and thorough testing of software functionality [9]. The literature also showed that traditional approaches to software development, emphasize the importance of project plans and documentations, and try to control unexpected changes.

However, major changes in requirements, scope and technology are out of the control of development teams. The question often is not how to minimize changes in a project but how to better handle inevitable changes throughout its life cycle [10]. XP methodology responds to this expectation by adopting strategies designed to reduce the cost of change throughout a project.

XP contains several elements and those elements are in favor at SPI. The development process in XP is mainly based on the decisions made by the developers themselves, and this reflects how the improvement of the process is done.

The communication between the participants is central for sharing knowledge and information. Communication in XP projects is informal and are performed on several levels: between the programmers during pair programming, among the

developers during the standup meeting at the beginning of each day, and between all the participants during the planning meeting in the Planning Game. The communication between the customer and developers goes on all the time since the customer is available on-site and the acceptance test communicates customer satisfaction and development progress. Pair programming and collective ownership must be mentioned together. These are important means for sharing knowledge of the complete system and programming techniques. Programmers can learn from each other and they can be familiar with all parts of the system. Hence, the learning process is a natural part and will continuously improve the performance of the team. Short iterations, short releases, and continuous integration give continuous feedback to the team on how they are doing. This means that if something is wrong or inefficient, this can be discovered early and handled immediately.

## **6. SPI Difficulties in XP**

There are elements or characteristics of XP that give difficulties to SPI and the development process. These are described here. The difficult issues for SPI in XP are[11]:

- 1- Few SPI solutions fits all kinds of projects .
- 2- XP in large teams.
- 3- Fast introduction of XP.
- 4- Rapid pace of iterations.
- 5- Lack of documentation.

The introduction of XP into large teams may cause several problems. XP is best suited for small or medium sized teams, so too large teams can cause difficulties. However, many adjustments can be made to XP to suit large teams, and these adjustments gives potentials for SPI. The lack of communication in large teams can for instance be solved by organizing experience and knowledge sharing better. As suggested in previous

sections we should introduce XP only to a small part of the team from the start and expand when things are working. This also apply to every new XP teams, namely to introduce XP slowly, and not expect that a complete introduction of XP will work perfect from the start. It is also suggested that one should try practices that cause problems for some iterations before they are rejected. The start-up problems may be solved when the experience has matured.

## 7. Conclusion

In the last couple of years, one of the major trends for software development organizations was the move towards web application systems. These years also saw a large number of small firms emerging in this area. Unlike older software organizations, these small firms do not have established development practices.

Their processes usually are immature and ad-hoc. Often this is coupled with a less than positive attitude towards software engineering practices and, especially, software process improvement initiatives and software metrics collection. In particular, code metrics (such as lines of code, code complexity etc.) and process improvements standards (such as the Capability Maturity Model) are often viewed as obsolete and assessing processes and metrics. Other Software Engineering techniques, such as requirements engineering and semi-formal specification, are only practiced in an abbreviated fashion.

And finally we have concluded the following:

1- There are many development methodologies for web-based applications.

2- Methodologies, whether used for traditional systems development or web development, have their uses and also their limitations.

3- Most of today's web application development processes are extensions

of standard software engineering processes.

5- A few of today's web application development processes have been derived from a business-oriented approach to applications development.

6- In spite of the large number of Web engineering processes that have been developed over the past few years, none has been fully accepted as "the" web engineering process.

7- A common weakness of all of the SPI methods identified in the literature is that they do not identify specific best practices within the software domain.

## References

- [1] Lowe, D. A Framework for Defining Acceptance Criteria for Web Development Projects. in Murugesan, S. and Deshpande, Y. eds. *Web Engineering: Managing Diversity and Complexity of Web Application Development*, Springer-Verlag, 2001, 279-294.
- [2] Christian Hofer, Software Development in Austria: Results of an Empirical Study among Small and Very Small Enterprises, *Proceedings of the 28 th Euromicro Conference (EUROMICRO'02)*
- [3] Constantine L. L., 'Lightweights, Heavyweights and Usable Processes for Usable Software', *Keynote at Software Development 2001*, San Jose, 19 April 2001.
- [4] Beck, K. (1999 ). *Embracing Change with Extreme Programming*. IEEE Computer 32(10): 70-77.
- [5] Lawler, Edward E. III (1997). *Rethinking Organization Size, Organizational Dynamics*, Autumn 1997, pp. 24-35.

- [6] Winger, Alan R. (1994). Is Big Really Bad? *Business Economics*, 29, pp. 38-42.
- [7] R. Baskerville, B. Ramesh, L. Levine, J. Pries-Heje, and S. Slaughter, "Is Internet-Speed Software Development Different?," *IEEE Software*, vol. Sep, 2003.
- [8] Haroon tarawneh,. Asim El Shiekh ,". A Theoretical Agile Process Framework for Web Applications Development in Small Software Firms," SERA 2008, PP: 128-135 IEEE
- [9] Jeffries, R.E. (1999). Extreme Testing – Why Aggressive Software Development Calls for Radical Testing Efforts, *Software Testing & Quality Engineering*, pp. 23-26.
- [10] Lehman, Meir, "Rules and Tools for Software Evolution Planning and Management", *Annals of Software Engineering*, 11:2, 2001.
- [11] M.C. Paulk, "Extreme Programming from a CMM perspective", *IEEE Software*, November/December 2001, pp. 19-26.