I.J. Education and Management Engineering, **2016, 5, 1-14**
Published Online September 2016 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2016.05.01

Available online at http://www.mecs-press.net/ijeme

Agile Methodologies in Software Engineering and Web Engineering

Arun Kumar Kamepally [a], Tejaswini Nalamothu [b]

[a] *Department of Computer Science, Kennesaw State University, Kennesaw, Georgia, USA*
[b] *Department of Computer Science, Kennesaw State University, Kennesaw, Georgia, USA*

Abstract

Agile development methodologies have been gaining their importance in the present scenario. The usage of agile methodologies has been increased over the past few years for the development of software applications. It is a new paradigm transdisplanary research area. This paper helps the researcher to get a better understanding of agile methodologies in both software and web engineering: introduction, adoption, usage and analysis of agile in both streams. The rest of the paper summarizes the differences in adopting agile in software engineering and web engineering and also it explains the limitations of using agile in software engineering and web engineering.

Index Terms: Agile methodologies, Software engineering, Web engineering, Web development.

1. Introduction to Software Engineering

"Software Engineering" This term first appeared in *Software Engineering Conference 1968 NATO* was meant to provoke thought regarding the current "software crisis" at that time. Since then, he has continued as a profession and field of study dedicated to creating software that is of higher quality, more affordable, maintainable, and faster to build. According to Don O'Neill, "Software engineering can be defined as the systematic design and development of software products and process management"[1]. There is still much debate about software engineering than it actually is, and if it fits the classic definition of engineering. It has grown organically from the limitations of visualization software as just programmed. "Software development" is a term widely used in the industry, which is more generic and not necessarily subsume the engineering paradigm. Software engineering is a framework in theory that describes the activities and tasks that need to be performed in a sequence to optimize the cost and productivity of software development. Software projects can be so large that we have to do careful planning. Implementation is no longer just writing code, but it is also following guidelines, writing documentation and also writing unit tests. But unit tests alone are not enough. The different pieces have to fit together. And we have to be able to spot problematic areas using metrics. They

* Corresponding author.
E-mail address: akamepal@students.kennesaw.edu, tnalamot@students.kennesaw.edu

tell us if our code follows certain standards. Once we are finished coding, that does not mean that we are finished with the project: for large projects maintaining software can keep many people busy for a long time. Since there are so many factors influencing the success or failure of a project, we also need to learn a little about project management and its pitfalls, but especially what makes projects successful. The Software Engineering Institute (SEI) has established a comprehensive model that is based on a set of software engineering capabilities that need to be present as organizations reach different levels of proficiency. To meet the software development goals, organization must incorporate an engineering strategy that encompasses the underlying process, methods and tools. This strategy is often referred to as a Software Engineering Paradigm. These methodologies or methods are chosen based on the nature of the challenges and requirements of software.

There are several traditional methodologies for developing software applications. As business requirements are becoming more complex day by day these traditional software development methodologies became obsolete. Many methods have been proposed in software engineering there are to meet business requirements. The debate over which methods are the best way to take to software development continues. Software quality is the benchmark for defining what methodology is efficient. In this paper, we introduce about agile development and agile web development. In later sections we briefly describe the main differences and advantages between agile web engineering and agile software engineering. Finally, this paper gives the conclusion whether agile web engineering is advantageous over agile software Engineering.

2. Introduction to Web Engineering

Web Engineering is a direct result of the growth of the Internet and the World Wide Web. The web was first conceptualized by Tim Burners Lee in 1989, as a way to share information among scientists. Developed in 1990, the World Wide Web contained web pages originally written in basic HTML, or Hyper Text Mark-up Language. Since then, the programming languages used and the requirements for content have exponentially increased in both options and complexity [2]. Web development has become more dynamic than its origins, and due to the explosion in mobile applications, cloud computing, and other various technological advancements, the need for quick, efficient web development is now more vital than ever.

As WebApps become larger and more complex, – Informality remains, but some degree of requirements gathering and planning are necessary – Urgency remains, but it must be tempered by a recognition that decisions may have broad consequences – Intuition remains, but it must be augmented by proven management and technical patterns – Art remains, but it must be complemented with solid design. As society shifts from traditional software development to web based development, previous methods of standardization and streamlining application development are no longer applicable. The requirements are shifting towards the presentation layer, which necessitates a change in how developers handle things. The next few sections will describe what web development is, and how it relates to Web Engineering, in more detail.

A. Web Development

Web development is no longer considered as a simple document authoring task. In fact, it requires the same rigorous methodologies and tools as in the successful *software engineering* discipline [3].”

The following information discusses the various aspects of web development. The first topic covered is the approaches of Web Engineering, followed by the general attributes of Web Applications and the final section briefly overviews the general classifications of Web Apps.

B. Web Engineering Approach

Web Engineering is web development that is performed using a “scientific, engineering, and systematic” approach [3]. According to Hussein Al-Bahadili's review of “Web Engineering: a Practitioner’s approach,” Web Engineering has become “a new engineering discipline that emerged to deal with proposing and developing efficient, effective, and reliable framework for building cost-effective and successful Web-based applications and systems" [4]. As Web Engineering is still a relatively new practice, there are many different propositions as to how to approach it efficiently. One is to implement reusable components with a standardized platform. This is known as "mash up software," or "mash ware". As more and more mash ware is developed, the gap between software and web engineering can begin to lessen [5]. Another is to use patterns found in development to create new applications, through the use of MDD, or Model-Driven Development. Model Driven Web Engineering is a trend that "raises the abstraction level of the construction process, allowing developers to focus on conceptual models instead of code" [6]. It also allows developers to use time more efficiently, and reduce the amount of redundant coding. It directs for not only model-based design, but for the use of models in the other phases of development, such as implementation. Finally, MDWE, or Model-Driven Web Engineering, focuses on the "interoperability of the currently existing methodologies" of web application development. Advances are being made in the area of web browsers and networking, but most web developers are aware of the basic limitations of what can be done with a Web App. This leads back to a few attributes left out from Pressman's list of Web App attributes: Network intensiveness, concurrency, unpredictable load, performance, and security. These attributes add to what is known of the limitations that developers must work within.

C. Attributes of Web Applications

The common attributes for web applications include the following:

Data driven: Many Web Apps access databases to provide information to users, and even without databases, the data that Web Apps present are important to their survivability.

Content sensitive: Content does relate to data, but Pressman refers to the quality and aesthetic nature of the content being presented. "Content is King" [7].

Continuous evolution: For a Web App to survive, it must evolve quickly to satisfy the users. There are a few points to add taken from Matheson [8]: 1) Low cost of change, 2) Any change is instantly accessible, 3) Easy distribution. Those three points do support reasons that a Web App can evolve continuously. And Coda mentions that "websites have an inherently dynamic nature" [8].

Immediacy: There is more pressure for a Web App to be brought to market quickly (a few days or few weeks is not unheard of). Web App projects often have compressed timelines [9].

Aesthetics: The look and feel is of greater importance to a Web App than other applications, except for Video Game Applications which thrive upon visual appeal. It is of interest to note that some video games (many MMORPGs – Massively Multiplayer Online Role Playing Games) have supporting Web Apps in which aesthetics is heavily emphasized. Huang *et al* [10] notes that a key feature of Web Apps is usability and user interface. All software projects should be aware of usability and user interface, but more importance is placed upon these key issues in Web Apps. Constantine and Lockwood state that UI design and usability are weak points in SE [11].

Low cost of change: Matheson does not go into detail, but based on this author's experience, many changes to a Web App can be implemented quickly. Simple text changes can be accomplished in a few minutes, if not seconds. With the proper use of cascading style sheets, sweeping changes can be made to the whole Web App. Some changes do require more time, such as a complete change to the business logic of a function, but even then, the change can be made relatively quickly.

Any change is instantly accessible: This does relate to the point above. When a change is made to a production Web App, the change is immediately available to any user that goes to that Web App. This has created higher expectations from users.

Easy distribution: This is essentially re-phrasing that changes are available instantly. There is no need for users to download and install updates to a Web App (there are exceptions – web-based applications, for example, may require new plug-ins for the client's web browser). Changes to a Web App are made on a limited number of computers (web servers). In many cases, the Web App resides on a single web server, but higher-end Web Apps with large user-bases reside in web farms (multiple web servers which balance the load of user requests).

D. Types of Web Applications

Eldai *et al* [12] described four classifications of web applications which give a broad understanding of the usage of Web Apps. The following table is a summary of the web application types.

Table 1. General Classifications of Web Apps.

Intranet Web App	Exists and is accessible only on a company's local network. It is not accessible from the Internet. It can be a static collection of pages of dynamic pages making heavy use of server-side technology (PHP, ASP and JSP – to name a few).
Web App	This is what most businesses use to conduct business through the Internet. Amazon.com would be a very good example of this class of Web App. These Web Apps can rely on a great number of technologies, both server and client side, in order to conduct business with customers.
Static Web App	Constructed using no server-side technology, the static Web App generally does not provide any dynamically created content. The information presented does not change unless the maintainer of the site updates the pages directly. Web Apps were all static in the early days of the WWW.
Extranet Web App	All of the above. Extranets are usually connected directly with a company. The Intranet would deal with internal concerns (providing functionality for employees and management). The Web App would be the public Web App for the company's customers. And the Extranet may contain many static pages in support of employees or customers.

These classifications help build a mental image of the complexity of the World Wide Web, but are too general for that image to be refined in details. It is definitely a "big picture" image of the Web Apps accessible through the Internet.

3. Web Engineering Versus Software Engineering

There are many similarities and differences between Web Engineering and Software Engineering. Below is a compilation of those differences, in the processes used, requirements for each, and finally in their application of Agile Principles in development. The main difference in the processes of web engineering and software engineering is that there are a few accepted processes for software engineering, whereas the few processes available to web engineering are not as widely implemented. Model-Driven Web Engineering (MDWE) and Mash ware are generally accepted as viable options in most web engineering circles, but the concept of web engineering is still new enough that there is no standard methodology. In addition to this, the difference in requirements detailed above requires that Web Engineering be treated slightly differently than Software Engineering. The use of multi-disciplinary teams creates a difference in the flow of development, with more emphasis put on the design phase, especially in attention to the User Interface. The requirement for speed in development, however, contradicts the requirement for more time spent on design. In effect, web applications must be designed well, created quickly and efficiently, and distributed as soon as possible. Web engineering and software engineering share similar characteristics. Table 2 highlights the differences between web engineering and software engineering.

Table 2. Difference between Web Engineering and Software Engineering.

Web Engineering	**Software Engineering**
Requires a complex backend that is hidden from the user.	Do not require a complex backend when compared to Web Engineering.
The User Interfaces are required to be simple, yet visually appealing.	Software Engineering's UI's are generally less of a focus in development.
Web Engineering requires skilled web developers for most, but not all, stages of development.	Software Engineering is generally executed by programmers or software engineers.
Well-designed Web Engineering components must be able to interact and integrate well with other aspects of the web application, but should also be able to stand alone.	Software Engineering components generally work together; there is less of an emphasis on reusability.
Web-applications are both developed and maintained daily.	Software is released much slower, and maintenance, if available, is just as slow.

Web engineering involves some programming and software development, and adopts some of the principles of the software engineering. Different phases like Requirements Planning, Design, Coding, and Testing are same as software engineering. Quality attributes are same for web engineering and software engineering. Both deliverables depend upon requirements. However many methodologies are proposed and adopting specific methodology depends upon requirement and developing environment.

4. Introduction to Agile Software Development

Agility is ability to respond to unpredictable changes with quick response and profitability [23]. Agile Development is a type of development that "encourages customer satisfaction and early incremental delivery of the software; small, highly motivated project teams; informal methods; minimal software engineering work products; and overall development simplicity." It puts an emphasis on delivery and communication with clients [13]. In fact, the clients are a part of the 3 stakeholders in development, along with the users and the developers. There are various methods of agile development, with two of the most popular being extreme Programming and Scrum. There are 12 principles of Agile Engineering:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for competitive advantage of customer.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter time scale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to do the work.

6. The most efficient and effective method of transmitting information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done, is essential.
11. The best architectures, requirements, and designs emerge from self–organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

These principles set the ground rules for agility. They are not always applied equally, and sometimes a few are left out completely of an agile process, but the general themes remain the same: focus on delivering small, quick iterations of the software, and focus on pleasing the customer or client.

The key features of Agile methods are continuous requirements gathering; frequent face-to-face communication; Pair Programming; refactoring; continuous integration; early expert customers Feedback; and minimal documentation [14]. The most widely used methodologies based on the agile principals are Extreme Programming (XP) and Scrum. However, other methods such as Feature-driven Development, Dynamic Systems Development Method, Crystal Clear method, and Lean development have been also used. There are many benefits of the agile approach. Organizations are claiming that agile leads to improved time-to-market, increased quality, reduced waste, better predictability and better team moral, although not all of this is supported by empirical evidence. [22]

5. Agile Methodologies in WEB Engineering

Software Engineering does not have the Web and Internet features. Web Engineering uses Internet and web technologies and software engineering principles for the systematic development of Web Apps [13].Web engineering has some challenges in applying traditional methodologies as it involves more elements to be concentrated up on. The following are some limitations of traditional web engineering processes, such as Waterfall and Spiral that led to apply agile processes in web engineering.

The following are the limitations of traditional web engineering processes:

- Lack of rigor, systematic approach.
- The complete system is not what the user wants.
- The system is not developed on time, cost increases. Lack of expandability and maintainability,
- Do not meet the performance requirements resources are wasted
- Less attention is given to development methodologies, testing and evaluation, quality assessment and control.
- Based largely on its practices of individual development.
- Lack of awareness of its life cycle.
- Needs analysis, re-design, development (including coding), the management, measurement, maintenance.
- Calls significant upgrades and system design decisions.
- It is an exercise - not an event. Legitimate concern about how they are created and their long-term quality and integrity.
- Do not rely on the initial users to debug Web App design comprehensive tests and run them before releasing the system.

Web Engineering involves some of the agile principles in its nature. All the agile principles are needed for producing efficient web application. For constructing a software application the three traditional models 1) Software model 2) Business model and 3) the domain model are efficient. However for the developing a web based application a creative design model is needed. Agile helps in developing the software more efficiently, as customer will have continuous access to the software during development, changes may happen anytime during the project development and team collaboration will be high.

A. Agile Web Engineering Process (AWE):

Agile web engineering process is one of the processes in applying agile principles that has been proposed specifically for web application development. Agile web engineering is a light weight process which helps to tackle the problems associated for the development of web based applications. AWE helps in better maintaining, implementing and continuous testing of applications [15]. This AWE solution helps in getting the solutions that will get satisfied by the end-users.

The life cycle is represented in the figure1. You can start developing a web app either in business analysis phase or evolution phase. We choose the evaluation phase for already started application [15].

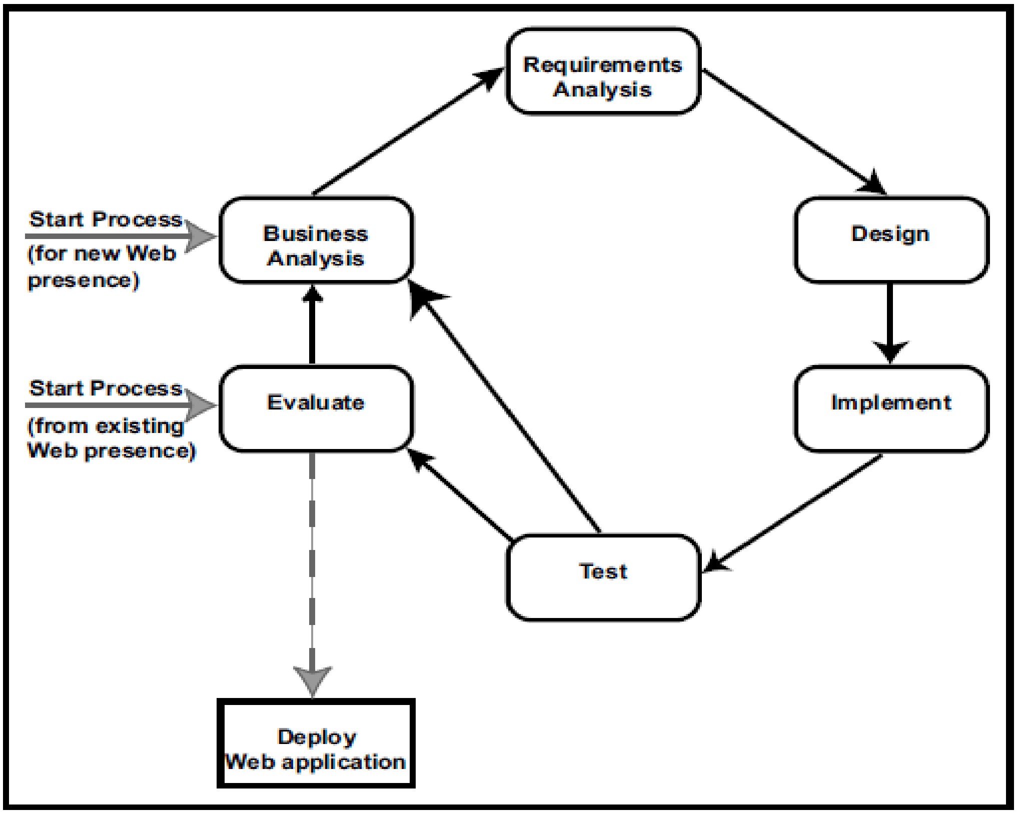

Fig.1. AWE Process Life-Cycle.

In this AWE user interaction will be high and Web engineering is multi-disciplinary, requires more reviews and collaborations are needed. However in Software engineering it mainly focuses on its functionality, there will be less reviews and take less time for developing software. Web application needs to be interacted with other applications so it needs more teams whereas software engineering needs few teams and it is very easy to focus and collaborate with each other. In web engineering the teams are inter-disciplinary in nature and most of the time they are not able to communicate. Therefore, they rely on frequent informal face-to-face communication rather than providing lengthy documentation [15]. Documentation is required and complete collaboration is not possible.

Though Agile web engineering process has certain strengths like: Flexibility, Collaboration, Simplicity, Iterative and Incremental, Strong focus on the end user, Usability. It has certain limitations and key challenges in-order to implement this agile web engineering process. However, for software engineering process usage of agile methodologies is much easier when compared to web engineering.

B. Other Agile Methodologies for Web Engineering

As applying agile principles for web applications is different from applying them to software applications, there are few other methodologies proposed in web engineering. These following methodologies are specifically designed for web applications.

1. Extreme Programing for web projects
2. Crystal Orange Web
3. Scrum for Web Development

Extreme Programing for web projects: XP for Web Projects emphases on producing executable code, automated test drives. It values more on refactoring, regression testing and continuous inspection by using pair programming. It incorporates the methods of traditional XP and also the elements to describe how graphics integrated in web applications. This process addresses only few principles. All the principles of XP are included and collaboration is done on-site development and it supports testing throughout the life cycle and new code will be continuously integrated by the developers. XP automated testing ensures a comprehensive approach to testing page layout, performance, and multiplatform operation [16]. However, it did not provided solutions for all the problems for example on-site development is impractical at some time. It also did not addressed for the projects having team more than 15.

Crystal Orange web: This is proposed specifically for web development applications by adopting some of the practices of crystal web.it can be applied for the team size of 30-40.It supports cross functional team to work together for a project. There are five categories and applied depending upon the project. The first category is Regular heartbeat with learning in which the development is done every 4 weeks for the software used by the public. Suggestions are made in this category. Another group is basic process, this group aims to organize the work piece and make sure it guarantees business initiative come to live in web project and it also organize the decisions. The other group is Maximum progress and minimum distribution and maximal defect free and a community aligned in conversation. The functional team is divided in cross functional teams for media content, business values and requirement and can focus on inter- disciplinary issues and the structure of team can be adjusted based on the project. It mainly focused on dividing teams and collaborating with each other which is the main issue in the traditional web-engineering. It is mainly helpful for short term web application projects. Crystal orange web is best suitable for projects having more teams and more elements for the software development like web application projects [17].

Scrum for Web Development: Scrum framework is project management that can be applied to web development environment. Scrum methodology mainly depends on self-organizing cross functional teams. Scrum projects make progress in a series of sprints, iterations that are temporary in case no more than one month. It can accept more change in requirements.it involves product backlog, sprint review, Most of the steps in agile software engineering is included in scrum just scrum is applicable for web applications but has many limitations. Scrum methodology is very helpful for web based projects for the following reasons.

- It is a slightly controlled method that emphasizes the frequent update of progress at work through regular meetings. Thus there will be a clear view of the project development.
- Due to short sprints and continuous feedback, it is very easy to deal with changes.
- Fast moving, cutting edge developments can be coded and tested using this method, since a mistake can be easily corrected quickly.

In compare to AWE, although XP and scrum addressed some of the issues associated with the development they did not address about the creation of structure that reflects the business needs that evolve periodically and to document it in the development process. Of all the methods we have discussed agile web engineering is well suited for web development. However, it lack some of the principles that should be applied in web engineering development process they are AWE needs proper training and expertise across the organization, currently perceived to be a problem with our process [15]. There might be issues on a large project crossing geographical boundaries." "One weakness is that AWE has not been used in a real project. This is perceived as a risk in an organization like this one. Another weakness is that AWE requires a cultural shift, which is easier to write about in paper than to happen in reality." Need to have a company that is forward thinking and willing to change. Need to have a culture where employees employed will want to change. Not suitable within the financial services sector [15]. It is based on staff are good. Poorest skilled developers could struggle more and managers may feel that they are losing control. Team members may be inadequate to the intense involvement that characterizes agile methods in we engineering and at the same time Prioritizing changes can be difficult when there are multiple stakeholders. It can be difficult to keep the interest of customers who are involved in the process in complex applications.

6. Agile Methodologies in Software Engineering

Agile software development incorporate aspects of lean software development and many helps to accept the change in requirements and mainly focusing on customer collaboration in entire Software engineering process. The development cycle is very less and team maintains lite process. Adopting light weight methodologies in software engineering give a greater advantage to change the requirements at any stage. As Software applications are not multidisciplinary and mainly focus on functionality only few team are assigned for a project that would help team to make decisions in ease. As architectural design is minimal teams can work on parallel on architectural design without any issues. Applying agile practices can reduce cost and can focuses on current requirements and no detail documentation is needed and developers can focus on requirements. Adopting agile principles in software engineering gives better results when compared with web engineering.

There are many methodologies proposed that applies agile principles to build software application very fast. All the methodologies aim to deliver the idea of agile principles. However they work in different ways. The most common methodologies are:

A. Extreme programming
B. Crystal Methods
C. Scrum
D. Feature driven development

All the above methodologies are proposed specifically for software applications and solved the problems that occur in software development process.

Extreme programing: This focuses on the development rather than managerial aspects of software projects. According to the book "Extreme programing explained" by Kent beck [18] concluded that XP mainly focus on coding, changing requirement, proceeding without knowing everything about the future, relying on other people and he also explained how XP is used in different projects and its life cycle. Unlike XP for web XP has no cross functional teams. Each team member works on the same component.

Crystal Clear: This is helpful in tailoring the needs of software develop techniques and it is not suitable for developing highly critical System and complex software application. The teams may be collocated because of the emerging needs of requirements. Crystal Clear methodology emphasis on face-to-face communication. It is one of the crystal families of methodologies that are varied based on the team size and project [19].

Scrum: This is a project management model applied to software engineering. It mainly focuses on how team should work together to produce a flexible software to accept the new change. The development cycle is 1 or 2 week reviews are conducted every day. It involves iterative and incremental cycles called sprints. The key practices in scrum are Product backlog, Sprints, Sprints planning meeting, pint backlogs. Both XP and Scrum can be used together to produce effective product.

Feature driven development: It does not address all the agile principles. It cannot be applied in entire software development, it is only applied in design phase and build phase. Though agile principles are applied in different ways in each method, some of the properties are common in all methods that include Feedback, Learning, Customer involvement, Frequent meeting, accepting a change in middle of the process focusing on business problems etc.

Adopting methodology is dependent on the types of application and environment in the software organization. Agile methodologies work best for teams with relatively small number of members as teams are very few in software engineering agile principles are very easy to apply in software engineering.

A. Agile Frameworks:

Frameworks are useful to define which methodologies are useful for software development [19]. Different methodologies can be combined to achieve good results. Frameworks will not specify how to carry the methods to achieve agile principles they only defines what to use to achieve agile principles.

1). Dynamic Systems Development Method
2). Adaptive Software Development (ASD)

DSDM (Dynamic System Development): This is a frame work to implement RAD development. Dynamic System Development can be used along with XP and Scrum is just a framework. Martin Fowler, one of the writers of Agile Manifesto, believes, “DSDM is notable for having much of the infrastructure of more mature traditional methodologies, while following the principles of the agile methods approach” [19]. There are nine practices to deliver the idea of DSDM .It also includes active user interaction, testing throughout the life cycle. Teams mainly focus on adaptive change and quality.

Adaptive Software Development (ASD): This is a frame work to apply agile principles in software engineering it uses some of the techniques of extreme Programing, This method is based on complex adaptive systems and it is mainly useful for applying principles in intense pressure and more adaptive change environment. Adaptive Software Development and crystal method can use any software engineering technique. Object oriented models are used to design models for development.

B. Limitations of Agile methodologies in Software engineering

This literature shows that agile methods are not completely suitable for software engineering. Perhaps these methods are based on object-oriented paradigm. The limitations we observed are the following:

- Agile methodologies mainly focus on pair programing and informal reviews to build reliable software. However, this type of technique is not yet been proven for reliable product.
- These methodologies involve the end user in the development process. Clients may be busy and involving

client is not possible some times.
- Keep it simple requires some extra work.
- Contracts can be a problem, as with other approaches to iterative development.
- The main limitation of agile practices is implementing agile practices in companies small firms require skilled personals.
- Other limitation of agile practices is regarding documentation as it mainly focuses on customer informal reviews documentation is not given a higher priority. However, documenting requirements and process flow is best practice to cope up with future problems.
- Continuous testing is needed throughout the life cycle and this may increase the cost of resources of the project.
- Agile principles given in the agile manifesto can be achieved only for certain software application.

There is still a debate whether agile principles are advantageous when compared with traditional methodologies. However, agile model has some limitations and advantages as it provides a solution for most common problems but not all.

7. Adaptability of Agile Methodologies in Web Engineering and Software Engineering

This section highlights the differences in adapting the agile methodologies to both web engineering and software engineering. Agile Development was originally developed with Software Engineering in mind, but as application development evolves into web development, these principles have been applied in different ways. Some choose not to develop using agile principles, instead choosing a different methodology, such as the Waterfall Method. Customer satisfaction remains a high priority in web development, and the low cost of quick development also supports the use of agile principles. In some ways, web development even complements agile development. For instance, the immediate release of web applications encourages user and client feedback, allowing the communication between developers and other stakeholders to remain steady.

Many frameworks have been developed under the belief that Web Engineering and Agility go hand in hand; Ruby on Rails is one of the most popular agile frameworks available. “Ruby on Rails follows the principles of “convention over configuration” and “don’t repeat yourself,” providing an agile web development framework that simplifies the development process and increases productivity for prototyping web applications [20].

Web engineering includes some of the agile principles like incremental communication but the process is rigged and needs more teams.” Agile development can also go hand in hand with popular web engineering practices such as Model-Driven web engineering [21]. On the other hand, some situations can make agile development difficult, such as when a company grows larger, and has to adjust to the newer demand. There are also some who believe that the use of agile development makes little difference in a web application's success. For example more importance is given to documentation in agile methods of web engineering and not in software engineering as the functionality of software application is limited.

Communication and testing nonfunctional requirements is very important and focus on non-functional requirements will be done at the browser level rather than functional. Agile SE methods are meant to deal with short development times, but they still suffer from the same as Software Engineering in general: Software Engineering focuses little upon content (including user interface design, graphic design, and presentation of data). In addition, the content is often created by non-software engineers (graphic artists, writers, and other experts in usability, HMI, etc.). Web engineering goes beyond SE due to the nature of the web, and due to the need for multi-disciplinary teams for Web App projects and also because of large functional teams and high scope of architectural issues. Moreover number of roles is needed in web engineering so they should work together.

In Agile, web teams are divided such that all teams perform same task while having different experts in team, however it requires more skilled human resources. As for the application of agile development, the many different applications of agility to software engineering parallel the many different applications of agility to web engineering: because the methods are flexible, and the rules need not be followed strictly, the agile principles can help improve almost any application's development. Table 3 outlines the difference between agile approaches in web and software engineering.

Table 3. Key Differences in Agile Approach in Web Engineering and Software Engineering.

Agile in Web Engineering	Agile in Software Engineering
Multidisciplinary in nature collaboration is not Possible between different teams.	Mainly focus on functionality so collaboration is possible between different teams.
Each sprint takes more time as more teams need to interact.	Each sprint takes less time as few teams conduct formal review.
Documentation is required as communication is very difficult between teams. XP for web solved the issues of large teams by giving more importance to documentation.	Face to face communication is required Documentation is not given high importance as communication is possible between teams.
A high chance of architectural issues arises as web applications may interact with multiple systems in real environment.	Less focus on architectural issue as standalone software works independently.
More emphasis on functional and non- functional testing.	Emphasis of functional testing rather than non-functional testing.
Need more people to maintain large teams.	Need few human resources to maintain the team.

8. Conclusion

As for the application of agile development, there are many different applications of agility to software engineering parallel the many different applications of agility to web engineering: because the methods are flexible, and the rules need not be followed strictly, the agile principles can help improve almost any application's development. There are a few very important differences between software engineering and web engineering, some of the agile principles are involved in web engineering like incremental development evolutionary model, agile methods of web engineering adopts some techniques of agile methods in software engineering however there are some issues associated in applying agile methodologies in web engineering. Different companies follow their methodologies based on the environment the methodologies that we have discussed in this paper are proposed to solve the most common problems in web engineering. All the methodologies proposed have not solved all the issues in web engineering but few. However applying agile methodologies in web engineering is very useful for small web applications. It can be said that agile methodologies in web engineering follows some of the techniques beyond agile software engineering because of its multi-disciplinary in nature and comprising more elements. The only difference and difficulty in web engineering using agile methods is communication continuous change in requirements and some architectural issues.

References

[1] Mills, Harlan D. "The management of software engineering, Part I: Principles of software engineering." IBM Systems Journal 19.4 (1980): 414-420. DOI. 10.1147/sj.194.0421

[2] Lee, Yang W., et al. "AIMQ: a methodology for information quality assessment." *Information &* management 40.2 (2002): 133-146. http://dx.doi.org/10.1016/S0378-7206(02)00043-5

[3] Mulder, Chris. "Challenges of Web Application Development: How to Optimize Client-Side Code." (2011), http://www.liacs.nl/assets/Bachelorscripties/2011-15ChrisMulder.pdf.

[4] Hussein Al-Bahadili. "Book Review - Web Engineering: A Practitioner's Approach by Roger S Pressman and David Lowe, McGraw Hill Higher Education, ISBN: 978-0071263771, 512 pages, Copyright 2008". International Journal of Information Technology and Web Engineering, Vol. 4, No. 3, pp. 78-80, July-September 2009.

[5] G. Reif, "WEESA—Web Engineering for Semantic Web applications." Ph.D. Thesis, Technische Universität Wien, 2005.

[6] Robles Luna, E., Grigera, J., & Rossi, G. (2009, June). "Bridging Test and Model-Driven Approaches in Web Engineering." In Proceedings of the 9th International Conference on Web Engineering (pp. 136-150). Springer-Verlag.

[7] Coda, F., Ghezzi, C., Vigna, G., and Garzotto, F. 1998. Towards a Software Engineering Approach to Web Site Development. In Proceedings of the 9th international Workshop on Software Specification and Design (April 16 - 18, 1998). International Workshop on Software Specifications & Design. IEEE Computer Society, Washington, DC, 8.

[8] El Sheikh, A. and Tarawneh, H. 2007. A survey of web engineering practice in small Jordanian web development firms. In Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (Dubrovnik, Croatia, September 03 - 07, 2007). ESEC-FSE '07. ACM, New York, NY, 481-490.

[9] Larry L. Constantine, Lucy A.D. Lockwood, "Usage-Centered Engineering for Web Applications," IEEE Software, vol. 19, no. 2, pp. 42-50, Mar./Apr. 2002, doi:10.1109/52.991331.

[10] Palmer, Jonathan W. "Web site usability, design, and performance metrics." *Information systems research* 13.2 (2002): 151-167.

[11] Constantine, Larry L., and Lucy AD Lockwood. "Usage-centered engineering for Web applications." *IEEE software* 19.2 (2002): 42-50.

[12] Eldai, O., Ali, A., & Raviraja, S. (2008, December). Towards a New Methodology for Developing Web-Based Systems. Proceedings of World Academy of Science: Engineering & Technology, 36, 190-195. Retrieved March 4, 2009, from Academic Search Complete database.

[13] Pressman R. S. "Software Engineering: a Practitioner's Approach" (2010) 7th ed.(McGraw-Hill, New York).

[14] Paulk, Mark C. "Agile methodologies and process discipline." *Institute for Software Research* (2002).

[15] Andrew Mcdonald,Ray Welland, 'Agile Web Engineering (AWE) Process: Multidisciplinary Stakeholders and Team Communication', International Conference, ICWE 2003 Oviedo, Spain, July 14–18, 2003 Proceedings,ISBN: 978-3-540-40522-1.

[16] Wallace, Doug, Isobel Raggett, and Joel Aufgang. *Extreme programming for Web projects*. Addison-Wesley Professional, 2003.

[17] Boehm, Barry. "A Survey of Agile Development Methodologies." (2007). http://agile.csc.ncsu.edu/SEMaterials/AgileMethods.pdf

[18] Beck, Kent. *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2000.

[19] Strode, D. E. "Agile methods: a comparative analysis." *Proc. 19th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2006), Wellington, New Zealand* 2006.

[20] Jose Ignacio Fernandez-Villamor, Laura Diaz-Casillas, and Carlos A. Iglesias. 2008. “A Comparison Model for Agile Web Frameworks.” In Proceedings of the 2008 Euro American Conference on Telematics and Information Systems (EATIS '08). ACM, New York, NY, USA, Article 14, 8 pages. DOI=10.1145/1621087.1621101 http://doi.acm.org/10.1145/1621087.1621101.

[21] Mart ńez, Yulkeidi, Cristina Cachero, and Santiago Meli á "Evaluating the impact of a model-driven web engineering approach on the productivity and the satisfaction of software development teams." *Web Engineering*. Springer Berlin Heidelberg, 2012. 223-237.

[22] Cubric, Marija. "An agile method for teaching agile in business schools." *The International Journal of Management Education* 11.3 (2013): 119-131.

[23] Erande, Ameya S., and Alok K. Verma. "Measuring agility of organizations-a comprehensive agility measurement tool (CAMT)." *International Journal of Applied Management and Technology* 6.3 (2008).

Authors’ Profiles

Arun Kumar Kamepally, graduated from Kennesaw state university and working as a Ruby on rails developer in Dell Private Limited- One of the world’s biggest IT Company. He was born in India on 8th May 1992.Arun received his Master’s degree in the department Computer Science Department at Kennesaw State University, Kennesaw, GA, USA in December 2015.Arun has done Bachelors of Technology in field of Computer Science at Jawaharlal Nehru Technological University, Hyderabad, India, 2013. He worked as a Graduate Research Assistant department of CS, Kennesaw State University. He worked as an ERP Trainee Engineer at CCL Products India Limited in his under graduation. He had presented various papers on Cloud Computing, Big data, Mobile Application Development in Graduation seminars. His research interests are Internet of Things, Big Data, Cloud Computing, Programming and Mobile Computing.

Tejaswini Nalamothu, born in India on 13th June 1993. Tejaswini received her Masters of Science in Computer Science degree from Kennesaw State University, Kennesaw, GA, USA with graduation in December 2015. Tejaswini has done Bachelors of Technology in field of computer science at KL University, Vijayawada, India, 2014. She worked as a Graduate Research Assistant during her entire Masters study, contributed and helped for finishing the research of several faculty. Several papers and posters on her research work were presented by her in Graduation seminars. Her research interests include Software Engineering, Project Management, Cloud, Information Systems, Data analysis and modeling and High level Programming. Ms. Tejaswini was a member of ACM Student Chapter. Her android project “Expense Tracker Application” was awarded with merit of Excellence on poster day at Kennesaw, ACM Student Chapter.

How to cite this paper: Arun Kumar Kamepally, Tejaswini Nalamothu,"Agile Methodologies in Software Engineering and Web Engineering", International Journal of Education and Management Engineering(IJEME), Vol.6, No.5, pp.1-14, 2016.DOI: 10.5815/ijeme.2016.05.01