



# HYBRID WEB ENGINEERING PROCESS MODEL FOR THE DEVELOPMENT OF LARGE SCALE WEB APPLICATIONS

OMAIMA N. A. AL-ALLAF

Department of CIS, Faculty of Sciences and Information Technology,  
AL-Zaytoonah University of Jordan, P.O. Box 130, Amman (11733), Jordan  
E-mail: [omaimaalallaf@zuj.edu.jo](mailto:omaimaalallaf@zuj.edu.jo)

## ABSTRACT

Many web development models had been proposed in the literature and there are few studies related to proposing models for developing large web applications. It is important to construct a model that lead to large web applications development with minimum likelihood of failure. A Hybrid web engineering process model for large web development enterprises has been proposed in this research. This model focuses on overall development process phases and consists of many activities: division of large web application into many small sub applications; division of large number of developers into many sub teams; identify a management team to control the development process; Stakeholders' and customers' inclusion and feedback during development process; requirement analysis and management; adopting the Spiral model by management team; adopting the Throwaway Prototype, XP agile method, Web Engineering process model by each sub team; conducting CMMI levels key process areas; conducting web engineering practices and SQA activities; and training developers on CMMI and web engineering practices. An evaluation of this model according to CMMI process areas has been conducted in this research. This evaluation has been carried out by many professional developers currently working in many large web-development Jordanian enterprises. The evaluation results shows that the Hybrid model highly satisfied each of CMMI level2 and CMMI level3, medium satisfaction of CMMI level4, and finally low satisfaction of CMMI level5.

**Keywords:** *Large Web Applications, Web Engineering, Web Engineering Process Model, Software Process Improvement (SPI), Capability Maturity Model Integration (CMMI)*

## 1. INTRODUCTION

Large web application is a software that includes large quantities of source code and functions and developed by more than 50 developers using many programming languages [1][2][3]. Many web development methodologies focused on user interface design but failed to address the overall development process of web applications [4]. At the same time, the traditional software process models have challenges to accommodate web specific aspects into their techniques and practices. Companies often implement agile development methods because of their lightweight nature and to solve problems they encountered with traditional processes. These problems are: incapable to face requirements change; lack of predictability in development process; late feedback; and long release cycles [5].

Agility can be applied to any software process for: effective response to changing requirements during development process; effective communication; and organizing a team to control the work to develop the software iteratively and deliver multiple software increments. The benefits of agile processes are as follows [6]: shortened development cycle-time by 75%; higher stability of work-loads; higher flexibility to change of development plans; higher quality by earlier feedback from the customers; and higher utilization of work-load by developing applications with a fixed number of developers. At the same time, many studies pointed out the shortcomings of agile processes such as limited support for: distributed development environments; involving large teams; developing safety-critical software; and developing large applications. Extreme Programming (XP) is the most widely used agile methodology [7]. According to above, web applications' development



requires a mix of web site development techniques together with properties of traditional software process models [8].

Many researchers addressed the main problems in developing the large scale web applications. McDonald and Welland [9] highlighted in a survey the serious problems that affect the development of large web projects in organizations. These problems are: poor project management and estimation; problems with requirement analysis phase; and focus on implementation with confusing in requirements capturing and designing.

Omaima Al-Allaf [10] determined in previous research using survey in five large Jordanian enterprises the: adopted process models; used agile processes; and development problems. The survey results showed that there is a weakness in adoption of agile processes in these enterprises. We noticed also that the Extreme Programming (XP) is well known agile process by the developers working in these enterprises. Also Omaima Al-Allaf [11] addressed in another research using a survey in large Jordanian enterprises the: characteristics of developers; properties of web development processes; development problems; and levels of usage of CMMI and web engineering practices. The survey results showed that, the tools and technology, and standards and procedures are partially adopted, whereas organizational issues, web metrics, and control of development process are barely used by these enterprises. While Balakrishnan and Somasundaram [12] addressed the requirements for the successful development of large projects such as: user involvement; management support; clear requirements and the ability to verify them; and proper planning.

The results of survey [11] showed that the web development enterprises faced problems related to software process improvement (SPI) and need an efficient model for implementing SPI activities in their development environment. It is essential to manage software processes efficiently to maintain the SPI implementations and increase the chances of obtaining positive results [13]. There is a need for project specific process tailoring. Many development processes for web applications had been proposed in the literature. Until now there are few studies related to web process models for developing large web applications. Therefore, this research focuses on suggesting hybrid web engineering process model to overcome as possible as the problems of web development. After that, an evaluation of this model according to capability maturity model integration (CMMI) key process areas and goals (KPA's) by the professional

developers working in large Jordanian web development enterprises. The remaining of this research is as follows: section 2 includes literature review; section 3 includes development processes in large enterprises; Section 4 describes the hybrid model in details; Section 5 includes the evaluation of the hybrid model; and finally, section 6 concludes this work.

## 2. RELATED LITERATURE

Web applications present important characteristics that must be addressed by software engineering processes. The web applications have to be delivered for the clients within the time constraints due to its strategic nature. The development of web application is similar to the development of traditional application and it is iterative in nature [14]. Pressman [15] suggested framework of web engineering process model (WebE) that consists of: customer communication; planning; modeling; construction; delivery and evaluation.

Souza and Falbo [16] presented an agile approach for development of web applications which used the concept of agile modeling. They started web engineering process with identification of business needs, followed by project planning. Next, requirements are detailed, modeled, analyses and designed. Then the application is built using tools, then, tested and delivered to end users. These activities should be applied in an iterative fashion, allowing for user feedback and system evolution.

Whitson [17] described a web development process (WebHelix) that consists of project management plan that creates a task list from a system architecture diagram and uses chart to synchronize the team workload. This method used a modified Spiral approach for development. Some parts of the design process, like the business analysis and planning, are done once at the beginning of the process. Some parts of the development process, like deployment and maintenance are done once at the end of the process. The major creation of the web application is done by using set of steps repeatedly, producing a set of more complete prototypes, with the final prototype being the completed web application.

The development of large web applications should be carried out using careful planning and systematic design methodology, and by integrating management and web technologies. While there is nothing that guarantee the success, Balasubramanian, et. al. [14] suggested that, the developers should take in their considerations



during the development process of large web applications five factors: scalability, availability, manageability, security and use proper development practices to greatly decrease the failure of large web applications.

McDonald and Welland [9] described an agile web engineering (AWE) process for development of web applications. They discussed and identified the interaction between business, domain, software and creative design within web projects in AWE process to deliver solutions that satisfy users. They outlined the AWE process life cycle detailing each stage and how the stages should be used an iterative approach.

Ginige and Murugesan [18] proposed an approach to developing large web systems that enable developers to successfully developing large web applications. They focused on web development process that assists developers in managing the changing requirements, facilitates the communication, and supports the continuous evolution, maintenance and management of the content. Their model consists of context analysis, product model, process model, sub project planning, web site design, web site maintenance, project management, documentation and quality control and assurance.

Finally Ericsson [19] suggested dividing the large system into several separate parts, each developed independently as a separate system. A super system is implemented by a set of interconnected systems communicating with each other to fulfill the work of this super system. He introduced an architectural pattern for systems of interconnected systems. Each involved system is described by its own set of models, separate from other systems' models. Each subordinate system is developed using rational unified process (RUP) as a black box considering other systems with which it communicates as actors. This construct allows recursion within one model, and it considers each subsystem a system in its own right.

### 2.1 Problems in Large Web Development

Many literature researches addressed the factors responsible for the failure of web applications and highlights problems that affecting the development of large web applications. The most important problems that caused the failure of development of large web applications are: problems in requirement analysis phase [9]; poor project management of development process [9]; poor project estimation [9]; and development of web applications that exceeds the budget [19].

According to literature review, most of related researches did not take in their considerations many problems when developing large web applications such as: did not consider the overall development process; did not consider the size of web application; did not consider the number of developers; insufficient requirements' analysis and management; inconsideration of stakeholders; lack of management methodology; lack of risk analysis; did not integrate many advantage properties of XP agile process such as pair programming, refactoring, and accepting changing requirements during the development process; and finally poor utilization of SPI CMMI and web engineering best practices.

### 3. DEVELOPMENT IN LARGE ENTERPRISES

The development organizations continue to move toward the development of large web applications and they assign such projects to small teams of qualified developers. Web applications demand faster time to market and the continual integration of new requirements. Therefore, such demands have increased the popularity of agile processes which let teams increase development productivity; maintaining software quality and flexibility; and increasing software organization's responsiveness. XP focuses on small teams and lets them replace paper-based documentation with face-to-face communication. As the development organization grows, the time spent exchanging product knowledge and training new people increases and often renders XP unsuitable [20].

The requirements often change during the software development life cycle in today's software development environment to meet business needs [21]. Agile processes allow for changing requirements during the development process and stress collaboration between developers and customers. The transition from traditional processes to agile processes affects the development team members and management. The development teams range in size from two to several individuals in development organizations. Therefore, the development process that is appropriate for very large teams will not work well for tiny teams and vice versa. Small teams can be flexible and adaptable in defining and applying agile processes. For a team with size that exceed 10, the large team can be divided into collections of smaller independent sub teams, each no larger than 10 and the interfaces well defined between sub teams [21]. Successful large projects require many developers at various levels of effort.



### 3.1 Mixing Agile with Traditional Processes

Each of traditional software process models and agile methodologies has its strengths and limitations. Khan [22] suggested that, any organization should think when to use agile and traditional methodologies. Agile methodologies are used when the: requirements are unknown and uncertain; uncertain budget for resources; and unknown risks. Traditional methodologies are used when the: requirements are well known and stable; sufficient budget for resources; and well understood risks. Finally, the compromise between agile and traditional approaches may need to be based on the nature of the project being undertaken. Projects are different in size, application domain, criticality and innovativeness. Agile approaches are more suitable for developing small projects.

The agile methodologies promise higher customer satisfaction, faster development times and a solution to rapidly changing requirements. The plan-driven approaches promise predictability, stability and high assurance. Both approaches have shortcomings that, if left unaddressed, can lead to project failure [23].

Many organizations of web applications' development are attempting to utilize both agile and traditional approaches. Although, the adoption of agile methodologies improves the productivity, quality and business satisfaction, there is also necessary need for other methodologies. Vishnu Vinekar, et. al. [24] suggested that, the agile development of systems requires a suitable organizational culture. It may be difficult to adopt all agile practices in projects that have stable requirements. New organizational structures are needed to sustain these opposing cultures so that organizations can get the full benefits of both agile and traditional systems' development. There is a need for simultaneously managing agile and traditional processes. The client's culture may be the deciding factor in using agile or traditional methods for a project [24]. The development teams need to find specific project characteristics to determine if they use agile or traditional process, or hybrid of the two to develop projects. Although many of their advocates consider the agile and traditional processes polar opposites, synthesizing the two can provide developers with a comprehensive spectrum of tools and options [25].

## 4. THE HYBRID MODEL

Hybrid model is suggested in this research to overcome the web development problems obtained from the literature studies and also problems addressed in the two researches [10][11]. This

model is based on integration of traditional Spiral, XP and WebE processes to improve the development of large web applications. Figure 1 shows the main steps of Hybrid Model.

### 4.1 Features of Hybrid Model

The features of this model are as follows:

- Possible division of large web application into many sub web applications.
- Possible division of large numbers of developers into small sub teams.
- Centralized management team to manage the development process of large web application.
- Stakeholder classifications into groups.
- Stakeholders' inclusion and feedback during the overall web development process.
- Requirement analysis, classification and management.
- Planning of large web application and also for each one of sub application.
- Risk analysis of large web application and also for each one of sub application.
- Cost, effort and time estimation of large web application and also for each sub application.
- Focusing on overall development process phases: requirement gathering and analysis, cost estimation, planning, design, coding, testing, integration, documentations and maintenance.
- Adopting the Spiral process model by management team as shown in figure 1 to manage all sub teams and the development process. The Spiral model is an incremental and evolutionary model. It maintains a Waterfall model but incorporates it into an iterative framework that more reflect the real world [26]. Whereas, in Hybrid model, we adopted WebE model and XP principles instead of Waterfall model.
- Using Throwaway Prototype model to be adopted firstly by centralized team to take initial requirements of large web application through applying Spiral steps. Secondly, to be adopted by each sub team to get more requirements about each sub application.
- Adopting many principles of XP process such as pair programming, refactoring, test-first programming, frequency integration, minimal documentation and accept changing requirements by each sub team.
- Adopting WebE model by each sub team to develop sub application.





- Configuration management of overall large web application.
- Testing, verification and validation of the large web application and for each sub application.
- Project monitoring and control for the overall large web application.
- Provide technical solutions for each sub web application.
- Integration of product, project management, teaming, supplier management and organizational environment.
- Integrating the software quality assurance (SQA) activities with web development process.
- Considering the SPI principles and web engineering best practices.
- Developers' training in large enterprises on both SPI and web engineering practices.
- Documenting and integrating all activities of the large web application.

#### 4.2 Management Components in Hybrid Model

To be conducted by the centralized team to manage sub teams and overall large web development process. The management team consists of many professional software engineering members each has responsibility of management of one sub team and they meet weekly. Figure 2 shows the steps which accomplished between each sub team and the management team. The management team has the responsibility to:

- Understand functions, objectives, environment and stakeholders' needs of the large application through customers' communications.
- Analyze and classify requirement into classes.
- Divide the large web application according to its size into many sub applications.
- Divide the large number of developers into many multidisciplinary sub teams.
- Assign the responsibilities for particular tasks to these sub teams.
- Address non-technical issues such as organizational and management policies; human resources development; and legal, cultural and social aspects.
- Adopt the steps of Spiral model (planning, scheduling, cost estimation and risk analysis)

for the overall large web application components.

- Identify all stakeholders (users of web application and organization that needs this application.
- Assign each member in the management team, the role of management one sub team. Give this manager the initial requirements, plan, risk estimation of this one sub application. Later, accept from him the sub application Prototype. This process is repeated until delivery of Prototype that matches the requirements. After management team agreement on final Prototype, the sub team begins to develop this sub application.
- Integrate, test and evaluate performance of each developed sub application with overall large web application and returns feedback.

#### 4.3 Development Component in Hybrid Model

To be conducted by sub teams. The sub teams may work in parallel (independent sub systems) or in incremental fashion if sub systems are depended on each other. Each sub team members meets every day. The responsibilities of each sub team are as follows:

- The manager of each sub team takes the requirements of sub web application.
- Communications with customers for more analysis for these requirements.
- More description of sub application goals, tasks and estimated effort.
- Adopting Throwaway Prototype for requirements elicitation. Each sub team manager delivers the sub application Prototype and takes feedback from the management team until the final version of Prototype which match the user requirements is build.
- Builds the sub web application using WebE process model and many XP principles according to the final version of sub application Prototype.
- The manager of sub team delivers the sub application with its documentation to the management team to integrate it with the main web application after testing it.

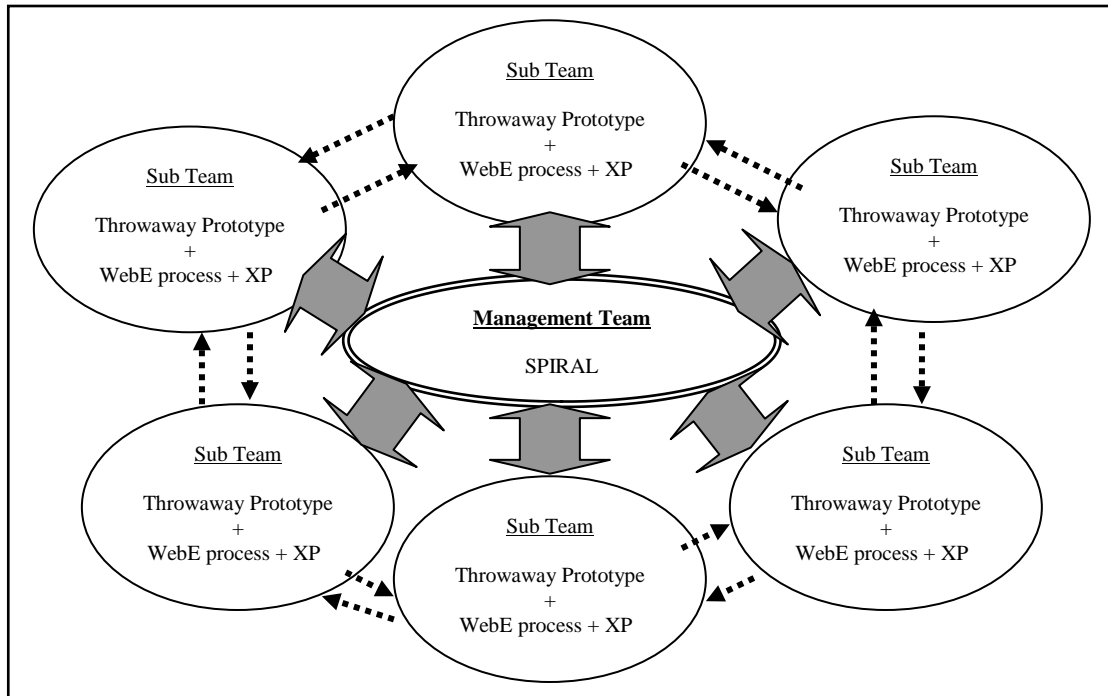


Figure 1: The Hybrid Web Engineering Process Model

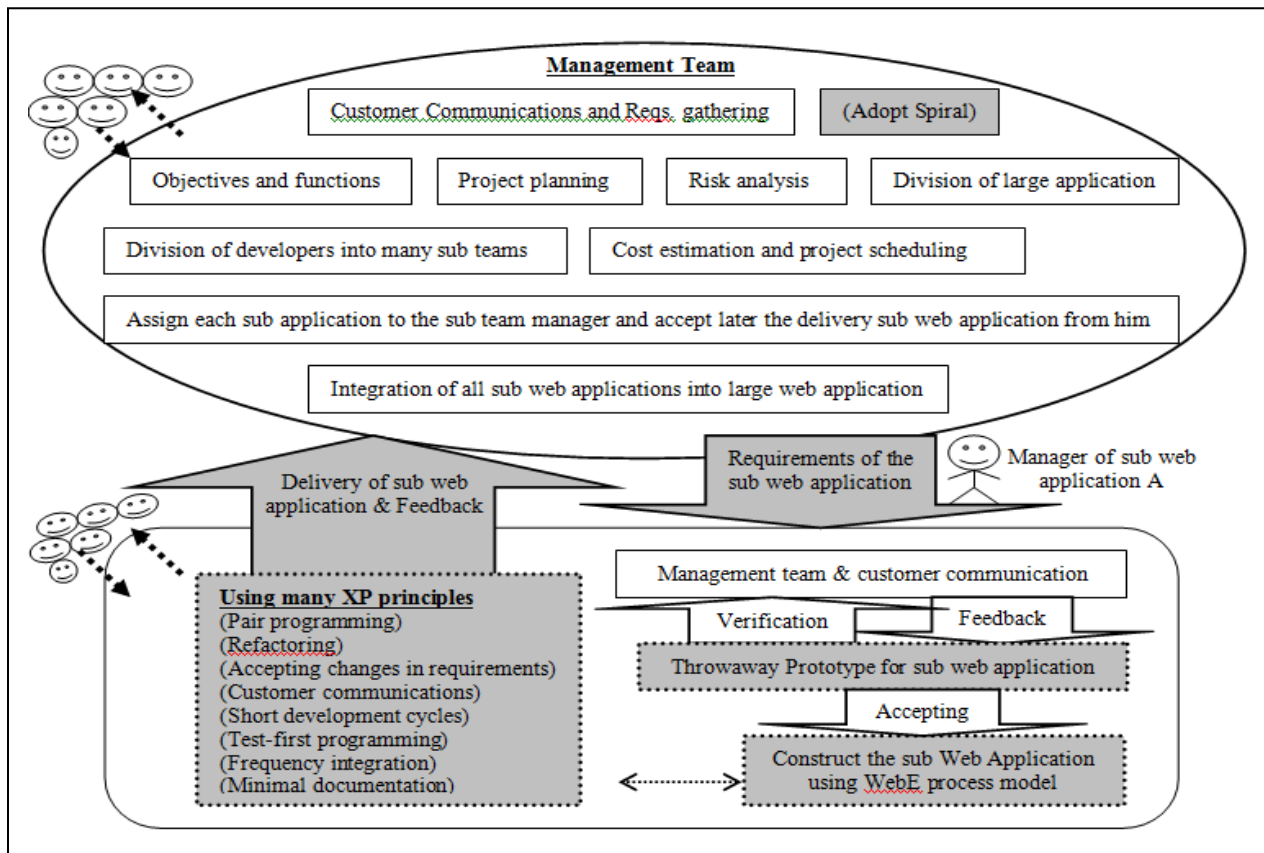


Figure 2: Roles AND Interactions BETWEEN Management AND Sub Teams



5. HYBRID MODEL EVALUATION

A formal evaluation of the use of Hybrid model by large scale Jordanian enterprises has been undertaken. A questionnaire is prepared and conducted to include all the CMMI KPAs [27]. The analysis units for this questionnaire are five large Jordanian enterprises which undertaken the development of large web applications. The questionnaire’s instrument has four parts: the first part lists CMMI level2 KPAs; the second part lists CMMI level3 KPAs; the third part lists CMMI level4 KPAs; and finally, the forth part lists CMMI level5 KPAs. This questionnaire was written in English language and reviewed and validated many times by three professional professors currently working in different Jordanian Universities and specialized in software engineering and management. The selection of these large enterprises was based on definitions of large projects mentioned in researches [1][2][3]. At the same time, the size of enterprises in Jordan is determined by the Statistical Office according to many factors such as: number of employees, budget, size of projects, and the time required for development process. The selected five enterprises have many characteristics such as they: involve 50 and more developers in development process; use more

than three programming languages in development; develop web applications to provide 50 and more functions, and more than 100 web pages to users and with more than hundreds of thousands of lines of code; project duration ranging from one to three years; finally many of these enterprises have many branches in other countries.

This questionnaire and a description of the Hybrid model are given to 100 professional developers in software engineering and management working in large Jordanian enterprises (as a research population). We received only 55 replies. The Hybrid model should be clearly read by the professional developers working in these enterprises to evaluate it according to CMMI KPAs. Therefore a hard copy includes the detailed description of the Hybrid model were attached with this questionnaire. Table 1 shows percentages of responses related to Hybrid model evaluation by these enterprises according to CMMI KPAs (level2, level3, level4 and level 5).

The results shows that the Hybrid model satisfied about 85.57% of CMMI level2, 86.86% of CMMI level3, 73.35% of CMMI level4, and finally about only 31.65% of CMMI level5. Figure 3 shows these percentages.

Table 1: Hybrid Model Evaluation According To CMMI

		Yes	No	Don't Know
<b>A</b>	<b>CMMI level2 Process Areas and Goals</b>			
1	Requirements Management	89.32%	4.68%	6%
2	Project Planning	85.95%	6.42%	7.62%
3	Project Monitoring and Control	89%	2%	8.99%
4	Supplier Agreement management	84.28%	2.82%	12.85%
5	Measurement and analysis	81.27%	5.425%	13.35%
6	Process and Product Quality Assurance	85.85%	4.975%	9.175%
7	Configuration Management	83.32%	5.71%	10.97%
	<b>Average</b>	85.57%	4.58%	9.85%
<b>B</b>	<b>CMMI level3 Process Areas and Goals</b>			
1	Requirements Development	89.34%	1.65%	9%
2	Technical solution	82.23%	4.42%	13.33%
3	Product Integration	90.37%	4.07%	5.54%
4	Verification	91.26%	3.72%	5.01%
5	Validation	92.68%	2.66%	4.66%
6	Organizational Process Focus	77.62%	3.78%	18.58%
7	Organizational Process Definition	80.68%	7.32%	12%
8	Organizational Training	88.55%	5.22%	6.18%
9	Integrated Project Management	92.5%	2.06%	5.41%
10	Risk Management	88.57%	4.75%	6.67%
11	Decision Analysis and resolution	87.4%	3.4%	9.2%
	<b>Average</b>	86.86%	4.17%	8.96%
<b>C</b>	<b>CMMI level4 Process Areas and Goals</b>			
1	Organizational Process Performance	80.0%	6.7%	13.3%
2	Quantitative Project Management	66.7%	6.7%	26.7%
	<b>Average</b>	73.35%	6.7%	20%

D	CMMI level5 Process Areas and Goals			
1	Organizational Innovation and Deployment	43.3%	33.3%	23.3%
2	Causal Analysis and Resolution	20.0%	30.0%	50.0%
	<b>Average</b>	31.65%	31.65%	36.65%

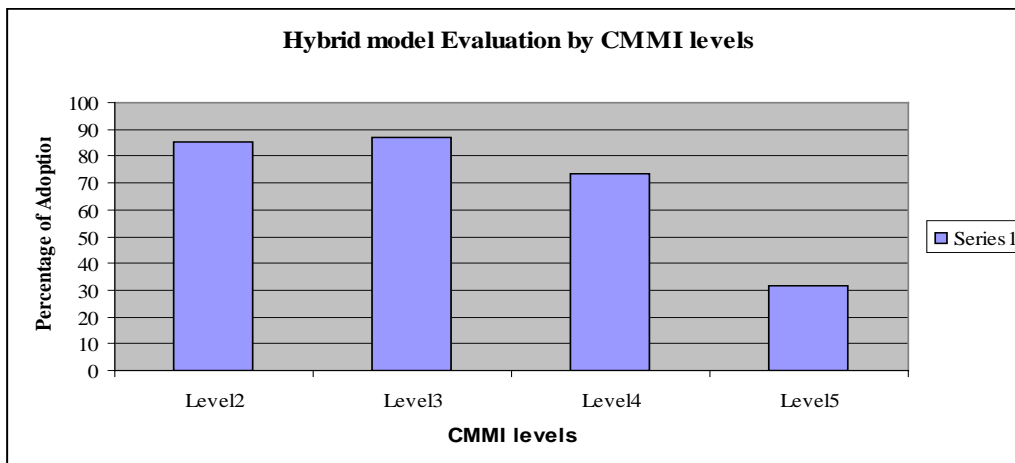


Figure 3: Hybrid Evaluation according to CMMI

## 6. CONCLUSION

Developing web applications is different from traditional software development and poses additional challenges because they differ in the nature and life cycle. The traditional software process models have limitations: the several phases in system development slow down the development process and the requirements specifications are not flexible since it is difficult to identify all customers' requirements at the beginning. These problems limit these models to adopt them in web development because the nature of these applications in rapidly changing requirements.

Many researchers recommend that these problems can be addressed by the adoption of lightweight iterative and incremental approaches such as XP to improve software development process. On the other hand, XP has many limitations: XP is used in small to medium size software projects; lack requirements management and lack of documentation of requirements; and XP is not suitable to adopt it by the large teams.

Therefore, we proposed a Hybrid model according to: problems obtained from the literature related to the development of large web applications; and problems obtained from the results of two surveys in large Jordanian web development enterprises [10][11].

We proposed this Hybrid model to: overcome the limitations of traditional software process, agile XP and WebE processes when adopting them in

large web applications' development in large enterprises; support the suitable development of large web applications within the time, budget and effort; overcome as possible as the web development problems in literature and in large web development enterprises [10][11]; and match the SPI and CMMI KPAs to improve the large enterprises' software process and software quality assurance.

In Hybrid model, we suggested to: divide the large number of developers into many small sub teams; identify a management team to manage the overall development process and all sub teams; divide the large web application into many small sub applications; adopt the Spiral model to be used by the management team as solution of poor risk identification and analysis and poor management problem; adopt XP principles to be used by each sub team; adopt the Throwaway Prototyping to be used by each sub team to overcome the XP requirement gathering, identifications and management limitations; adopt the WebE process model to be used by each sub team to match the nature of web applications' development; and monitor and control the development process of sub web applications by the management team to be under the SQA activities, SPI practices and CMMI KPAs.

The core of agile is to embrace change. To succeed, enterprises need to change their core business principles and the ways they are conducting business (change the organization culture). This means that agile methodologies could



change the way the enterprises manage their software projects. Therefore, we suggested adopting sufficient training for all developers on agile XP principles.

An evaluation of the Hybrid model has been carried out by distributing questionnaire to many professional developers and managers currently working in different large Jordanian enterprises. This questionnaire is based on CMMI KPA's and goals. The evaluation results showed that this Hybrid model is highly satisfied CMMI level2 and level3 and partially satisfied level4, and finally poorly satisfied CMMI level5.

#### REFERENCES:

- [1] J. Garland and R. Anthony, "Large-Scale Software Architecture: A Practical Guide Using UML", *John Wiley & Sons Ltd*, England, 2003.
- [2] D. Bell, "Software Engineering: A Programming Approach", 3<sup>rd</sup> edition, *Addison Wesley, Pearson Education Limited*, 2000.
- [3] B. Curtis, H. Krasner and N. Iscoe, "A Field Study of the Software Design Process for Large Systems", *Communications of the ACM*, NY, USA, Vol.31, No.11, Nov1988, pp.1268-1287.
- [4] D. Howcroft and J. Carroll, "A Proposed Methodology for Web Development", *Proceedings of the Eighth European Conference on Information Systems (ECIS)*, Vienna, July 2002, ECIS, 2002.
- [5] K. Baxter et. al, "Extreme Usability: Adapting Research Approaches for Agile Development", *Conference on Human Factors in Computing Systems (CHI 08)*, Florence, Italy, *ACM*, NY, USA, Apr2008, pp.2269-2272.
- [6] M. Aoyama, "Web-Based Agile Software Development", *IEEE Software*, IEEE Computer Society Press Los Alamitos, CA, USA, Nov1998, Vol.15, No.6, pp.56-65.
- [7] B. Kent, "Extreme Programming Explained: Embrace Change", *Addison-Wesley Longman Publishing Co., Inc.* Boston, MA, USA, 1999, ISBN: 0-201-61641-6, Reading MA.
- [8] R. Vidgen, "Constructing a Web Information System Development Methodology", *Information Systems Journal*, Vol.12, No.3, Jul2002, Blackwell Science Ltd, pp.247-261.
- [9] A. Mcdonald and R. Welland, "Agile Web Engineering (AWE) Process", Technical Report TR-2001-98, Dept. of CS, University of Glasgow, Scotland, G12 8QQ, Dec2001.
- [10] O. Al-Allaf, "The Adoption of Agile Processes in Large Web Development Enterprises: A Survey In Jordan", *International Journal of Web Applications*, ISSN: 09747710, Vol. 2, No.3, Pages: 206-216, Sep2010.
- [11] Omaima N. A. Al-Allaf, "The Usage Of Capability Maturity Model Integration And Web Engineering Practices In Large Web Development Enterprises: An Empirical Study In Jordan", *Journal of Theoretical and Applied Information Technology (JATIT)*, ISSN 1992-8645, Vol.39, Issue.2, pp:150-166, May 2012.
- [12] S. Balakrishnan and M. Somasundaram, "Large Scale Object-Oriented Application Development in Practice", *Technology Review*, Tata Consultancy Services, Nov2001.
- [13] K. Kautz Hansen and K. Thaysen, "Applying and Adjusting a Software Process Improvement Model in Practice: The Use of the IDEAL Model in a Small Software Enterprise", *Proceedings of 2000 International Conference on SE*, ACM, ICSE, Ireland, pp.626-633.
- [14] V. Balasubramanian, A. Bashian, and P. Daniel, "A Large-Scale Hypermedia Application Using Document Management and Web Technologies", *Proceedings of the eighth ACM conference on Hypertext*, Southampton, UK, pp.134-145, ACM NY, USA, 1997.
- [15] R. Pressman, "Software Engineering: A Practitioner's Approach, 6<sup>th</sup> edition," *McGraw-Hill International Edition*, 2005.
- [16] V. E. Souza and R. Falbo, "An Agile Approach for Web Systems Engineering", *ACM International Conference Proceeding Series: Proceedings of 11<sup>th</sup> Brazilian Symposium on Multimedia and the Web*, Brazil, Vol.125, pp.1-3, ACM Press NY, USA, Dec2005.
- [17] G. Whitson, "Webhelix: Another Web Engineering Process", *Journal of Computing Sciences in Colleges*, Vol.21, Issue.5, pp.21-27, USA, May2006.
- [18] A. Ginige and S. Murugesan, "Web Engineering: A Methodology for Developing Scalable, Maintainable Web Applications", *Cutter Journal*, Vol.14, No.7, Jul 2001.
- [19] M. Ericsson, "Developing Large-Scale Systems with the Rational Unified Process", *Rational Software White Paper*, Rational, the e-



- development company*, USA, Rational Software Corporation, 2000.
- [20] F. Maurer and S. Martel, "Extreme Programming Rapid Development for Web-Based Applications", *IEEE Internet Computing Magazine*, ISSN: 1089-7801, Vol.6, No.1, pp.86-90, , Jan 2002.
- [21] L. Rising and N. Janoff, "The Scrum Software Development Process for Small Teams", *IEEE Software, AG Communication Systems*, Vol.17, No.4, pp.26-32, August 2000.
- [22] A. Khan, "A Tale of Two Methodologies for Web Development: Heavyweight vs. Agile", *Report, Minor research project in IS*, Dept. of IS, Univ. of Melbourne, Australia, pp.615-690, June 2004.
- [23] B. Boehm and R. Turner, " Balancing Agility and Discipline: A Guide for the Perplexed", Boston, Addison-Wesley, 2003.
- [24]V. Vinekar, W. Craig, Slinkman and Sridhar, "Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View", *Contemporary Practices in Systems Development*, ISM, Vol.23 Issue.1, pp31-42, 2006.
- [25] B. Boehm, "Get Ready for Agile Methods with Care", *IEEE 64, Computer*, Vol.35, Issue.1, pp.64-69, Jan 2002.
- [26] B. Boehm, "TRW Defense Systems Group, A Spiral Model of Software Development and Enhancement", *Computer: The flagship publication of the IEEE Computer Society*, May1988, Vol.21, No.5, pp.61-72.
- [27] Software Engineering Institute, Capability Maturity Model® Integration (CMMI®) Overview, Pittsburgh, PA 15213-3890, Carnegie Mellon University, 2005  
<http://www.sei.cmu.edu/cmmi>