

Software Development Process Improvement Framework (SDPIF) for Small Software Development Firms (SSDFs)

Mejhem Yousef Al-Tarawneh¹, Mohd Syazwan Abdullah², and Jasem Alostad³

^{1,2} College of Arts and Sciences, School of Computing, Universiti Utara Malaysia
06010 UUM Sintok, Kedah, Malaysia

³The Public Authority for Applied Education and Training (PAAET), College of Business
P.O. Box.23167, Safat 13092, Kuwait

Abstract

Most of the software development organizations all over the world are Small Software Development Firms (SSDFs). These firms have realized that it is necessary to organize and improve their software development and management activities. Traditional software process improvement (SPI) models and standards are generally not possible to be implemented directly by SSDFs, as these firms are not capable of investing the cost of implementing these programs due to limited resources and strict deadlines to complete the projects. In addition, the existing regional SPI models which were developed for SSDFs are not suitable to be implemented by SSDFs all over the world. Furthermore, SSDFs also have ignored the software development practices to explain “how to do the improvement”; as they only focus on “what to do for improvement”. This paper presents a new software development process improvement framework (SDPIF) for SSDFs based on eXtreme programming (XP) as the software development method and Capability Maturity Model Integration version 1.2 for Development (CMMI-Dev1.2) as the SPI model.

Keywords: *CMMI-Dev1.2, XP Method, Software Development Process Improvement Framework.*

1. Introduction

Technological advancements affect our life in many ways, and control our way of living in all sectors. In the software development field, we can see the high spread of SSDFs all over the world [1]. These firms play a crucial role in the economy of many countries, where they develop a large portion of the required software applications, offer many job opportunities, and exploit new technologies [2]. Unfortunately, these firms are suffering from problems related to developing their software products as they are unaware of the basic software development best practices. The main reason to

this is that most of them are using ad-hoc manner for the software development [3].

In addition, these firms have lack of understanding of the success factors of SPI and do not have enough people to perform all the SPI activities. Therefore, they find themselves to be very far from implementing formal SPI traditional models and standards, such as ISO 9001 Series, ISO/IEC 15504 (SPICE), ISO/IEC 12207 and BOOTSTRAP, where these models and standards were developed for large and very large firms, very complicated and too expensive to be implemented by SSDFs [4].

Even though the SPI traditional models and standards are difficult to be implemented directly by SSDFs, SPI in these firms is still possible through simplification of these models and standards [1] [5]. There are some regional initiatives of SPI which were developed for SSDFs, such as: OWPL in Belgium; ASPE-MSD in Brazil; PRISMS in Britain; iFLAP in Sweden; MESOPYME in Spain, MoProSoft in Mexico; and MPS in Brazil [6][7]. However, these initiatives are not suitable for SSDFs all over the world, as they were developed based on the characteristics, environments, and infrastructures of firms in these specific countries where the models originated. In addition, the development of these initiatives were based on simplifying the SPI traditional models or standards by selecting the suitable Key Process Areas (KPAs) of SPI traditional models or standards which are suitable for SSDFs in the specific country, without identifying the suitable software development practices that would achieve global quality level [6][8].

Pikkarainen [9], and Lina and Dan [10] indicated the need for a suitable SDPIF for SSDFs. This improvement framework should specify how to carry out the tasks of improving the software processes [4]. In addition, Richardson [11] stressed the need of these firms to have SDPIF, which focuses on the software processes activities, and providing faster return on investment, flexible, and easy-to-use. The framework will allow these firms in knowing “what to do for improvement” by the SPI model and “how to do the improvement” by software development best practices [9].

This paper highlights the reasons of selecting XP as a software development method and CMMI-Dev1.2 as a SPI model in developing the new SDPIF for SSDFs as discusses in Section 2. Section 3 details out the establishment of the SDPIF and the findings of the stages used in developing the framework. The new framework and how it can be used is discussed in section 4. Section 5 concludes the paper and discusses future direction in this area.

2. Why XP Method & CMMI-Dev1.2 Model?

Lightweight software development methods are more suitable for SSDFs such as agile methods that are more applicable for these firms [12]. Agile development methods [13] have been designed to solve the problem of delivering high-quality software on time under constantly and rapidly changing requirements. The XP method is considered as the most popular and effective method compared to other agile development methods for software development processes in SSDFs. Due to the flexibility and agility of XP method, this method is reflect to as extreme, as it take good aspects in developing the software and applies these aspects extremely.

XP method has been used in this study as a baseline development method in developing the SDPIF for SSDFs these several reasons such as: (1) XP is more applicable for small, medium-scale and less complex projects and it is the most widely used agile methods as well as one of the more prominent approaches that adheres to agile principles; (2) XP is easy to use; (3) XP could be easily adapted with changing requirements; (4) XP achieves SPI better than agile methods; it conforms to level two in CMMI-Dev1.2; and (5) XP practices work tightly together by carefully applying different practice at a time that will eventually lead to SPI [9][14][15].

As for CMMI, this model is the comprehensive software improvement model of the Software Engineering Institute (SEI) based on some emerging CMM models which are Capability Maturity Model for Software (SW-CMM) v2.0 draft, Systems Engineering Capability Model (SECM), and Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98. Furthermore, the CMMI model was represented in eight versions which are: CMMI for Development v1.02 (2000), CMMI for Development v1.1 (2002), CMMI for Development v1.2 (2006), CMMI for Acquisition v1.2 (2007) and CMMI for Services v1.2 (2007), CMMI for Development v1.3 (2010), CMMI for Acquisition v1.3 (2010) and CMMI for Services v1.3 (2007) [16].

In this study, CMMI-Dev1.2 was chosen as a SPI model in developing the SDPIF for SSDFs for several reasons such as: (1) This model was written specifically for the software industry to guide the software development improvement processes [17] and also to improve upon the best practices of other improvement models in many important ways [18]; (2) CMMI-Dev1.2 provides a comprehensive integrated solution for development and maintenance activities applied to products and services [4]; (3) CMMI-Dev1.2 is a widely-used beneficial approach for identifying the key weaknesses of a software development process which need immediate attention and improvement especially with agile development methods [9]; (4) CMMI-Dev1.2 can aid SSDFs in achieving their quality goals when used as guides for SPI [19]; and (5) Even though CMMI-Dev1.3 is the newest version of the CMMI generations, CMMI-Dev1.2 has been broadly used for assessing and improving the organizational maturity and process capability of most software development firms in the world, as this model presents extensive descriptions of how the various good practices fit together [4].

In addition, CMMI-Dev1.2 model and XP practices could be used as a combined approach to integrate the best abilities of both together, where these aspects not only can co-exist, but they even support each other [10]. Furthermore, CMMI-Dev1.2 is considered a suitable way to improve the software process of XP method [15], where high levels of CMMI-Dev1.2 would be possible to be achieved by extending XP method [20]. However, there is no extension work carried out in this respect, but it rather focuses on mapping XP method to CMMs KPAs. Accordingly, XP method is used with CMMI-Dev1.2 model in this study to develop the SDPIF for SSDFs.

3. Establishment of the SDPIF for SSDFs

Ramsin [21] used stages strategy to develop software modeling analysis methodology, where each stage has goals and tasks to achieve these goals. Stages strategy is useful and suitable to be used in this study to establish the SDPIF for SSDFs. In this study, four stages are required to be followed sequentially to establish the desired, which are [8]:

- **Stage One:** Aligning XP method to CMMI-Dev1.2.
- **Stage Two:** Developing the proposed SDPIF.
- **Stage Three:** Verifying the the proposed SDPIF.
- **Stage Four:** Validating the verified SDPIF.

Sections 3.1 to 3.4 explain these stages and discuss the results of each stage in detail.

3.1 Stage One: Aligning XP practices to the specific goals of CMMI-Dev1.2 KPAs.

This stage aimed to identify the coverage ratio of XP practices to the specific goals of CMMI-Dev1.2 KPAs. This alignment was based on the specific goals, because all the generic goals are repetitive throughout the specific goals [22]. In this stage, three scales had been used to identify the coverage ration of XP practices to the KPAs of CMMI-Dev1.2 which are: (1) Largely supported: XP practices largely support the specific goals of the KPA; (2) Partially supported: XP practices partially support the specific goals of the KPA; and (3) Not-supported: XP practices do not support or not applicable for the specific goals of the KPA.

As a result of this alignment, the CMMI-Dev1.2 KPAs are classified into there groups as follows [8]:

- **Largely supported (++):** This group consists of twelve KPAs, which are: project planning, project monitoring and control, configuration management, technical solution, product integration, verification, validation, integrated project management + IPPD, risk management, decision analysis and resolution, and quantitative project management, causal analysis and resolution.
- **Partially supported (+):** This group consists of eight KPAs, which are: requirements management, measurement and analysis, process and product quality assurance, requirements development, organizational process definition + IPPD, organizational training, organizational process performance, and organizational innovation and deployment.

- **Not-supported (-):** This group consists of two KPAs, which are: supplier agreement management and organizational process focus.

At the end of this stage, the coverage and missing specific goals of each KPA are known and used as inputs in stage two.

3.2 Stage Two: Developing the Proposed SDPIF

This stage starts by extending XP method to fulfill the partially and not-supported KPAs of CMMI-Dev1.2. In this regard, the Extension-Based Approach (EBA) of the Situational Method Engineering (SME) theory was suitable to be adapted in this study to extend the XP method, as this approach was developed for extending the existing methods to achieve specific issues [23]. Figure 1 shows the processes of the adapted EBA.

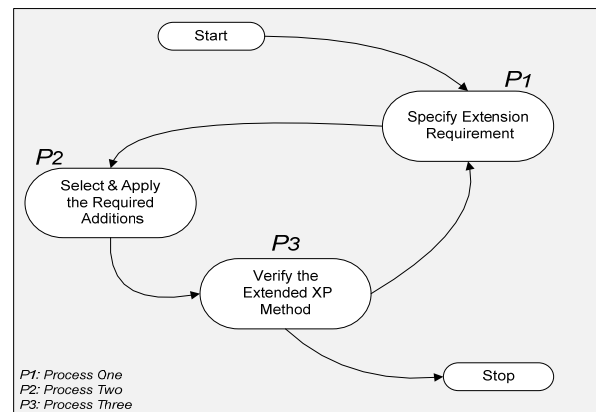


Fig. 1 EBA in to Extend XP Method (adapted from [23]).

As shown in Figure 1, three main processes (P1, P2, and P3) are used in extending XP method, which are [8]:

- **P1 (Specify extension requirements):** this process aims to extract the required software development, management, and improvement addition that are needed to cover the partially and not-supported KPAs of CMMI-Dev1.2. In this regard, the required additions were extracted to fulfill the missing KPAs of CMMI-Dev1.2.
- **P2 (Select & apply the required additions):** this process aims to extract the new phases of the proposed Extended-XP method and harmonize these phases to be comprehensive for all the popular software development methodologies. In addition, it distribute the required software development, management,

and improvement additions into the new phases of the proposed Extended-XP method based on the need for these additions during the software development lifecycle.

In this regard, the popular software development models (Waterfall, Spiral, incremental, and prototyping) and the required software development, management, and improvement additions were used to extract the new phases of the Extended-XP method to be harmonized and homogeneous compared to generic activities of the popular development models.

- **P3 (Verify the Extended-XP method):** this process aims to verify the commitment of the proposed Extended-XP method to the principles of XP method. This process is very important to ensure that the Extended-XP method is still applicable as an agile method. In this respect, XP values that reflect the XP principles were used as a main question during the verification process in stage three.

In general, the SPI framework consists of four generic elements, which are [24]: (1) Software Process: a set of tools, practices, and methods to produce software products according to specific plan; (2) Assessment: this element is used to assess the current state of the software process and this can be done by implementing the suitable assessment methods; (3) Capability Determination: this element is used to know the capability level of the software process and what motivates an organization to do process improvement by identifying the capability and risks of a process; and (4) Improvement Strategy: based on the capability determination results, the improvement strategy will identify the changes which should be made to the process.

In this study, the generic elements of the SPI framework were used as a baseline in developing the desired framework. Thus, these elements had been re-arranged to be suitable for software development and improvement issues by integrating the CMMI-Dev1.2 as assessment model and the proposed Extended-XP method as a development method into the generic elements of SPI framework. Figure 2 shows the foundation of the elements in the proposed framework.

As shown in Figure 2, there are three generic processes in the new proposed software development improvement framework which are: (1) assess the current software

development processes; (2) adopt the Extended-XP method; and (3) identify the best practices of the current project firms. Section 4 explains in detail each stage of the framework after the verification process.

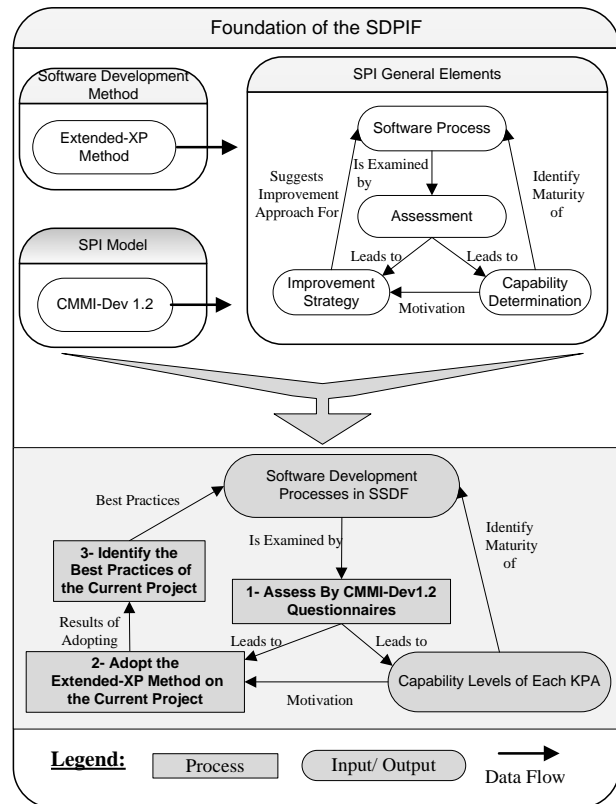


Fig. 2 Generic elements of the new framework.

At the end of this stage, the newly proposed framework goes to the third stage for the verification process.

3.3 Stage Three: Verifying the Proposed SDPIF

In this stage, the focus group method coupled with Delphi technique had been used to: (1) verify the compatibility of the proposed SDPIF to the specific goals of CMMI-Dev1.2 KPAs; (2) verify the commitment of the proposed Extended-XP method to XP values; (3) verify the suitability of the proposed framework and proposed Extended-XP roles for their related practices; and (4) verify the suitability of the proposed framework and the proposed Extended-XP structures for the software development and improvement issues.

In this regard, seven professional developers and managers with three expert researchers have participated in verifying the proposed SDPIF, where three rounds were performed in conducting this verification process. Table 1 shows these rounds.

As a result of the verification process, several modifications had been made to the proposed framework. The major modification was to remove the related activities of the organizational innovation and deployment process area from the proposed framework, as this area is not suitable to be implemented by SSDFs. Section 4 presents the verified framework (including the verified Extended-XP method).

Table 1: Focus group rounds.

Round	Session	Activities
R.1	S.1	<ul style="list-style-type: none"> - Researcher introducing himself. - Thanking the focus group members for the participation. - Presenting the research problem. - Presenting the purpose of the research.
	S.2	<ul style="list-style-type: none"> - Explaining the verification questions. - Answering the verification question individually. - Explaining if there are any inquiries about the verification questions.
	S.3	<ul style="list-style-type: none"> - Discussing the answers and suggestions of each focus group member by all the members.
R.2	S.1	<ul style="list-style-type: none"> - Modifying the proposed framework as suitable suggestions of focus group members.
R.3	S.1	<ul style="list-style-type: none"> - Viewing the verified framework to the members. - Asking if there is need for more new modifications.

At the end of this stage, the proposed SDPIF has been verified as suggested by focus group members [8]. To ensure that the verified framework is suitable SSDFs, there is a need to validate the suitability of this framework with more professional managers and developers who are working in these firms. Section 3.4 explains the two approaches used in validating the verified framework.

3.4 Stage Four: Validating the Verified SDPIF

Two validation approaches were used to validate the verified SDPIF, which are: a quantitative research method that involved a survey to validate the suitability of this framework for SSDFs, and qualitative research method (descriptive statistic) that involved two case studies to validate the applicability and effectiveness of this verified SDPIF for SSDFs. Section 3.4.1 & 3.4.2 explain the results of the two validation approaches.

3.4.1 Validating Suitability of the Verified SDPIF for SSDFs

A formal validation for suitability of the verified framework by SSDFs has been undertaken by using CMMI-Dev1.2 questionnaires as the main items in this validation. The questionnaire format consists of two parts: the first part is to know the general demographic information about the respondents; while the second part is to include all the specific goals of the suitable CMMI-Dev1.2 KPAs.

In this respect, the verified framework should be clearly read and understood by the professional developers and managers in these firms to evaluate it according to the characteristics of their firms and the requirements of the specific goals of each CMMI-Dev1.2 KPAs. Therefore, a hard copy which includes the detailed description of the verified framework (included the verified Extended-XP method) and the descriptions of CMMI-Dev1.2 KPAs were attached with these questionnaires.

These questionnaires were given to 80 professional developers and managers who are working in different SSDFs in Jordan. A total of 80 questionnaires distributed, and only 37 questionnaires were returned and 7 of them returned with missing data of more than (30%) for each questionnaire. Therefore, only 30 questionnaires have been used for the validation process. Sections 3.4.1.1 & 3.4.1.2 present the results of the two parts answers.

3.4.1.1 Part One: Respondents' Profile

This part consists of four questions: current position; current work; size of firm; and software experience. As a result of the answers of this part, the majority of the respondents were members of software engineering process group (40%), while the rest are distributed: managers (26.66%), technical members constituted (20%), and project or team leaders were (13.33%). Additionally, with regard to the current work activities the highest ratio was for code and unit testing (26.66%), where software design (16.66%), software quality assurance (17%), each of configuration management and software requirement (13.33%), and finally software process improvement (6.66%). In term of CMMI training, (90%) of respondents did not receive any CMMI training, while (10%) have received training. With regard to the software experience term, (66.66%) of respondents had 6-10 years, where the other two periods (less than 5 years & 11 years and above) got the same ratio (16.66%). Concerning the firm's size,

(46.66%) of the respondents had working in firms that had 20-31 employees while (20%) of respondents had working in firms that had 41-50 employees. The firms that had 10 – 20 employees and 31 - 40 employees got the same ratio (16.66%).

3.4.1.2 Part Two: Validating the Suitability of the Verified SDPIF for SSDFs

In this part, the respondents were asked to rate the level of the suitability of the verified framework components for SSDFs. The questions of this part consisted of scaled-response from 1 to 5, 1= Strongly Unsuitable and 5= Strongly Suitable. Based on the interval scale, the appropriate interval is calculated as: appropriate interval = (number of scales -1) / (number of scales) [25]; the appropriate interval for this study is $(4/5) = 0.8$. Therefore, Table 2 shows the definitions of the intervals scales that explain the level used for each interval scales.

Table 2: Intervals scale definition of the suitability.

Interval	Degree of Suitability
From 1 To 1.80	Strongly Unsuitable
From 1.81 To 2.61	Unsuitable
From 2.62 To 3.42	Average
From 3.43 To 4.23	Suitable
From 4.24 To 5	Strongly Suitable

From the answers of the questions in this part, is can be concluded that [8]:

- **CMMI-Dev1.2 Level Two:** at this level, four KPAs achieved at the level of strongly suitable as follows: requirement management (4.66); project planning (4.50); project monitoring and control (4.56); and measurement and analysis (3.46), while the remaining three KPAs only achieved the level of suitable as a follows: supplier agreement management (3.50); process and product quality assurance (3.60); and configuration management (4.16).
- **CMMI-Dev1.2 Level Three:** at this level, just three KPAs achieved the level of strongly suitable as follows: technical solution (4.50); verification (4.30); and validation (4.36), while the remaining eight KPAs achieved the level of suitable as a follows: requirements development (3.83); product integration (4.13); organizational process focus (3.93); organizational process definition +IPPD (3.56); organizational training (3.93); integrated project management +IPPD (3.50); risk management (3.63); and decision analysis and resolution (3.60)

- **CMMI-Dev1.2 Level Four:** the two KPAs of this level achieved the level of suitable as a follows: organizational process performance (3.46); and quantitative project management (3.66).
- **CMMI-Dev1.2 Level Five:** causal analysis and resolution (3.63) is the only one KPA at this level, and achieved the level of suitable.

Therefore, it can be concluded that the verified SDPIF is suitable for SSDFs as all the related activities in the verified framework that aimed to achieve the requirements of the specific goals of the suitable CMMI-Dev1.2 KPAs are rated strongly suitable or suitable for these firms.

3.4.2 Validating the Applicability and Effectiveness of the Verified SDPIF for SSDFs

In order to validate the applicability and effectiveness of the verified framework for SSDFs, two Jordanian SSDFs implemented this framework to improve their software development processes in developing their software projects. The first firms used the verified framework in developing a computer skills online examination system, while the second firm used the verified framework in developing a brokerage online system [8].

At the end of implementing the verified SDPIF by the two firms, three evaluation criteria had been used to ensure that the verified framework is effective for these firms, which are: gain satisfaction [26], interface satisfaction [26], and task support satisfaction [26][27].

For this, the interview method has been used as a data collection method to evaluate the modified framework. The primary purpose of the interview method is to understand the meanings that the interviewees attach to issues and situations in context that are not structured in advance by the researcher’s assumptions [28]. Therefore, it was suitable in this study to conduct interview with the projects team members of the two SSDFs to evaluate the effectiveness of the modified framework for SSDFs.

The results of evaluating this framework in terms of gain satisfaction, interface satisfaction, and task support satisfaction that are concluded from the answers of evaluating the verified framework by the two case studies can be summarized as follows:

- Gain Satisfaction Criteria

- **Perceived usefulness:** the verified framework enable the project team to execute their roles correctly and with high efficiency, because the

practices of each role are clear. Therefore, the productivity of each member of the project team was good compared to ad-hoc manner which had been used in the firm before implementing the verified framework. Furthermore, the distributions of the roles in the first stage of the verified framework are very similar the current role of each member which led them to execute their roles easily.

- **Decision support satisfaction:** project managers were responsible for decision making, where this is the main role of the managers. Furthermore, the continuous communications between team members assist in making better decisions.
- **Comparison with other guidance:** the verified framework stages were very clear for the project teams, where each member has specific roles to perform and as such, there is no overlap between their roles.
- **Cost (effectiveness):** the verified framework was cost-effective because the coach enabled the project to keep on the right path and kept the team working on the current features for the actual iteration. Also, the tracker was careful not to interrupt the project too many times. Furthermore, the verified framework guidance helped the team in ensuring that they are implementing their roles in the right ways.
- **Clarity (clear and illuminate the process):** the verified framework was very clearly for the project team by the training that was conducted in the first stage of this framework. Moreover, the roles of coach and tracker helped the project team for any inquiry during the development processes.
- **Appropriateness for task:** the verified framework stages were comprehensive, where stage one helped to motivate the firm to implement the verified Extended-XP method (stage two) as the development method and stage three helped to identify the best practices of using this method.

- Interface Satisfaction Criteria

- **Perceived ease of use:** during the training process, the verified framework was easily understood.
- **Internally consistent:** the roles of each member in the team were very clear, where these roles ensure consistent development process.
- **Organization (well organized):** the verified framework was well organized and structured, where the sequence of verified framework stages

and verified Extended-XP phases were useful to understand the activities easily.

- **Appropriate for audience:** based on developing the systems by the verified framework, the audiences were satisfied with product releases that helped them to add more features on the required products, as the verified Extended-XP method is incremental and iterative software development method.
- **Presentation (readable and useful format):** framework-SEPG members indicated that the verified framework is readable and is in the appropriate format. The project team also highlighted that the phases of the verified Extended-XP method were very clear, as the training process helped them to understand all things about this method.

- Task Support Satisfaction Criteria

- **Ability to produce expected results:** as a result of stage three for each case study, project managers and framework-SEPG members indicated that the implementation of the verified framework returned high capability levels compared to the levels before implementing this framework.
- **Ability to produce usable results:** the managers and framework-SEPG members indicated that the completed systems were usable for the end users because the customers participated in developing the systems (On-Site Customer), so the products were highly usable for the systems owners.
- **Completeness (adequate or sufficient):** managers and framework-SEPG members indicated that the verified framework was comprehensive for improving the software development and management processes in SSDFs. However, they argued it would be more sufficient when all KPAs of CMMI-Dev1.2 level five were included.
- **Ease of implementation:** framework-SEPG members indicated that the verified framework was very easy to implement, where the descriptions of each phases were clear. Therefore, it was easy to know the roles of each member in the developing process. The project teams also argued that the verified Extended-XP method was easy for implementation.
- **Understandability (simple to understand):** framework-SEPG members indicated that the framework was understandable. Project teams also argued the activities of the verified Extended-XP

method were easy to understand especially after the training processes.

4. The Verified SDPIF for SSDFs

As a result of verification process, the proposed framework had been verified. This section explains the stages of the verified SDPIF for SSDFs. In addition, section 4.1 presents the verified Extended-XP method. Figure 3 shows the verified SDPIF. The verified framework consists of three stages as a following:

- **Stage One: Assessing the Current Software Development Processes**

Prior to implementing the verified framework, the framework-SEPG members are responsible for determining a suitable simple repository to be used during the implementation of the framework. To achieve this, the Microsoft Office is suggested as a tool for data storing issues. In this stage, framework-SEPG members start to assess the current software development processes by using the questionnaires of

CMMI-Dev1.2 KPAs to determine the capability levels of these processes. Three scales can be used in this assessment. These are: (1) largely supported: the current software development processes achieve the majority of the specific goals of the KPA; (2) partially supported: the current software development processes achieve some of the specific goals of the KPA; and (3) not supported: the current software development processes can not achieve the specific goals of the KPA. As a result of this self-assessment, the firm will know the weaknesses of the current software development processes.

Then, the Framework-SEPG members are responsible for rearranging the current software development processes to be suitable with the required roles of the verified framework based on the assessment results. This can be done by distributing the new roles of the verified framework to the project team members as to commensurate with their experiences. At the end of this stage, the new roles will be known for each employee in the firm.

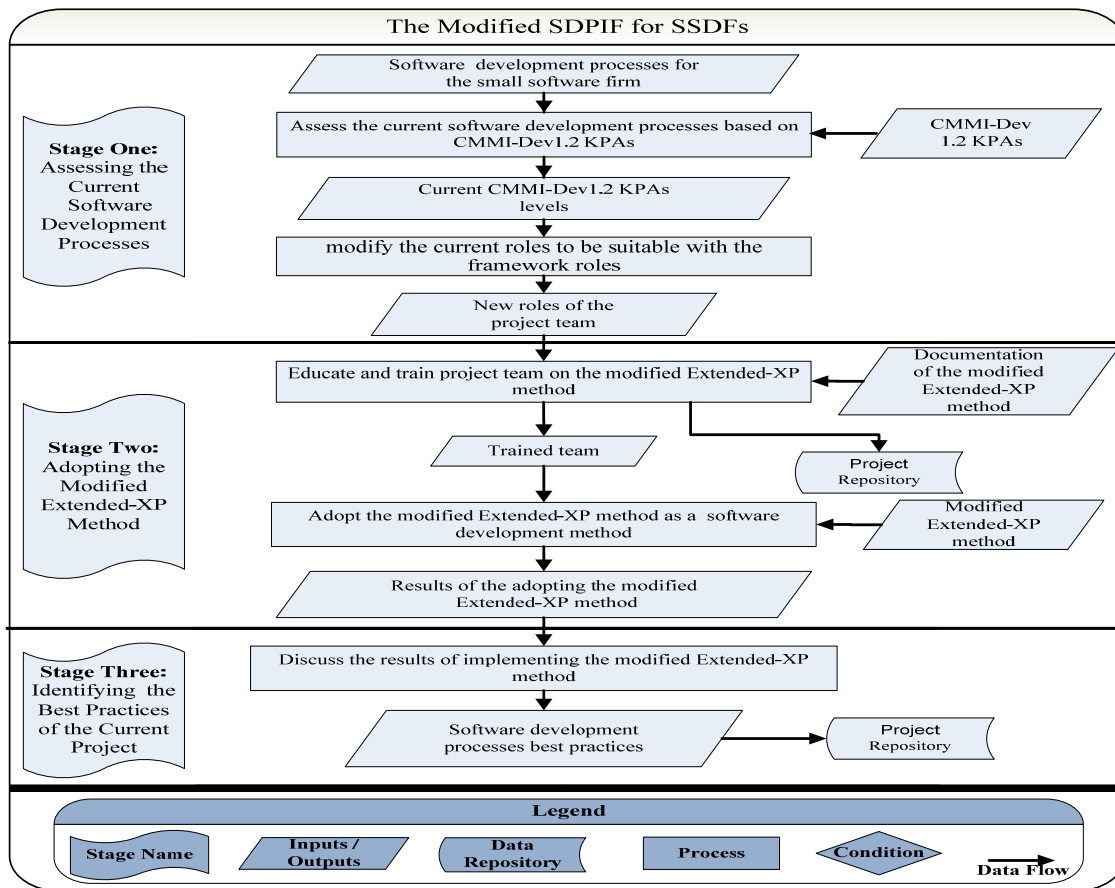


Fig. 3 SDPIF for SSDFs.

- **Stage Two: Adopting the Verified Extended-XP Method**

In order to implement the verified Extended-XP method in the right way; all the team members involved in the software development processes must have a good knowledge in their roles and must be trained. The best way to learn the verified Extended-XP method is through training courses. Furthermore, there is a need to support the team with the required XP books and with the documentation of the verified Extended-XP method during the training and the development lifecycle. Here, the framework-SEPG members are responsible for carrying out the training process prior the implementation of the verified Extended-XP method by the firm, while the members are responsible for: establish plans for training the developers, estimate the time required of training, determine if there is a need for outsourcing professional team in the training process, training the developers on how they can implement the activities of the verified Extended-XP, assessing the project teams efficiency to know if they are ready to implement the verified Extended-XP method or there is need for more training, and recording the training efficiencies in the project repository.

As a results of the training process (through the assessment of the team's efficiency), it will be known if there are needs for more training or not. Accordingly, the teams will be ready to adopt the Extended-XP method in the right way. Section 4.1 discusses the phases of the verified Extended-XP Method.

- **Stage Three: Identifying the Best Practices of the Current Project**

Referring to the results of the second stage, framework-SEPG members are responsible for meeting the project team to discuss the best practices of implementing the verified framework by using the specific practices of CMMI-Dev1.2 KPAs as the main items in this discussion. In this questionnaire; three choices have been used to answers these questions which are; "Yes" when the practice is well established and consistently performed; "Don't Know" when there are uncertainty about how to answer the question; "Does Not Apply" when the required knowledge about the project or firm and the question asked, but the question does not apply to the project; and "No" when the practice is not well established or is inconsistently performed. Through this, the best practices of

implementing the verified framework for the current project can be determined. Then, the framework-SEPG members are responsible for documenting these best practices in the project repository to be taken into account for the coming projects.

As for the roles in the SDPIF, this framework has the same roles of XP method [13] with several additional practices to programmers, coach, and tracker. Furthermore, there are two new roles that have been added to this framework compared to XP roles which are framework-SEPG members and Extended-XP-SEPG members. The roles in the SDPIF are as follows:

- **Programmers, Customer, Tester, Coach, Tracker, Consultant, and Big Boss:** these roles have the same practices of the XP method roles [13]. In this framework, these roles are used during the implementation of the modified Extended-XP method.
- **Coach and Tracker:** together coach and tracker are responsible for implementing the required metrics to achieve the objective of process and product quality assurance and the process performance at the third phase of the proposed Extended-XP method which are: (1) calculating the difference between estimates and actual time spent on user stories or tasks; (2) calculating the velocities of the projects and the length of pair programming sessions and store these in the project repository; and (3) calculating the number of failed acceptance tests, and number of severity defects after release.
- **Programmers and Extended-XP-SEPG Members:** these members are responsible for implementing the supplying process at the first phase of Extended-XP method, where programmers are responsible for extracting the required unavailable development tools or services; while Extended-XP-SEPG members are responsible for executing the supplying process with the external suppliers.
- **Framework-SEPG Members:** these members are responsible for: (1) specifying the suitable simple project repository in the first stage of the verified framework to keep the important date during the implementing of this framework; (2) assessing the current software development processes in the first stage of the framework; (3) modifying the roles of the current software development processes to be suitable with the verified framework roles in the first stage of the verified framework; (4) arranging the required organizational training before starting to adopt the verified Extended-XP method in the

second stage of the verified framework, which are: establishing planning for training the programmers, estimating the time required of training, determining if there is need for out sources professional team in training process, training the project team on how they can implement the activities of the verified Extended-XP, recording the training results and assessing the training efficiencies in the project repository, and meeting the project team in the third stage of the verified framework to extract the best practices of the current project and storing these practices to help incoming projects in the same firm.

4.1 The Verified Extended-XP Method

The verified Extended-XP method consists of four generic phases which are:

- **Phase One: Requirement Management**

This phase consists of the same contents of the three generic processes which are: (1) exploring the customer requirements process: this process consists of the same activities of the exploration phases in XP method [13]; (2) Supplying the required development tools and services process: this process is used to support the project with the unavailable required development tools or services, where programmers are responsible to identify the unavailable required development tools and services. Then, the Extended-XP SEPG members are responsible for determining the type of acquisition that will be used for the products to be acquired, selecting suppliers, establishing and maintaining agreements with suppliers, executing the supplier agreement, monitoring selected supplier processes, evaluating selected supplier work products, accepting delivery of acquired products, and transitioning acquired products to the project; and (3) Planning process: this process consists of the same activities of the planning phases in XP method [13].

- **Phase Two: Development**

This phase has the same activities of iteration to release phase in the XP method [13].

- **Phase Three: Product Delivery and Product & Process Efficiency**

This phase consists of the same activities of productionizing phase in XP method [13] and other additions such as: (1) several metrics that could be appropriate for objectively verifying the products and the process which are: release plan adherence, percentage of test cases that are running successfully, percentage of acceptance tests that are

running successfully, length of pair programming sessions, and project velocity; (2) convey the metrics through defined channels to the affected parties and senior management. At the end of this phase, there is a need to keep the metrics results to help in the measurement of the same user requirements for the coming projects.

- **Phase Four: Maintenance & Death**

This phase consists of the same activities of maintenance and death phases of XP method [13].

5. Conclusion and Future Work

SSDFs represent a high proportion of software firms around the world. Unfortunately, most of these firms are adopting add-hoc manner in developing their software products and are unaware of the basic software best practices and the existence of SPI traditional models and standards, where all of these models and standards were developed for large firms. In addition, the regional SPI initiatives are not suitable to be implemented by SSDFs all over the world, as they were developed based on the environments of firms in these specific countries where the models originated, and they neglect the software development practices to do the suitable improvement.

This study aimed to help SSDFs in developing a suitable SDPIF to enable them for managing and improving the software development activities in a systematic way to keep these firms alive and make it more effective. In this respect XP method and CMMI-Dev1.2 model have been selected in this study to develop a new framework for several reasons as discusses in Section 2.

This paper presented the results of the research stages used in developing the new framework (aligning XP method to CMMI-Dev1.2 KPAs, developing the proposed SDPIF, verifying the proposed framework, and validating the verified framework)

Referring to the results of the first approach of validating the modified framework, all the software, development, and improvement practices that are used in developing the modified framework to achieve the twenty one KPAs of CMMI-Dev1.2 were suitable for SSDFs. Furthermore, based on the responses of the team members of the two case studies on the evolution criteria questions at the second validation approach; it can be concluded that the modified framework is useful, useable, satisfy user needs and valid for use by SSDFs.

There are several potential directions for extracting or complementing this research, such as: fulfilling the

missing KPAs and specific practices of several KPAs; using more agile practices; and applying the framework to case studies of larger scope.

References

- [1] A. Baruah, "Contribution of Software Process Improvement Approaches For Small and Medium Scale Enterprises", *International Journal of Computing and Corporate Research*, Vol. 2. No.2, 2012, pp.1-10.
- [2] P. Savolainen, H. M. Sihvonen, and J. Ahonen, "SPI with lightweight software process modeling in a small software company", *Lecture Notes in Computer Science*, 4764, 2007, pp. 71-81.
- [3] G. Valdes, M. Visconti, and H. Astudillo, "The Tutelkan Reference Process: A Reusable Process Model for Enabling SPI in Small Settings", in *Systems, Software and Service Process Improvement: 18th European conference (EuroSPI 2011)*, 2011, pp. 179-190, Roskilde, Denmark, 2011.
- [4] I. Garcia, C. Pacheco, and J. Calvo-Manzano, "Using a web-based tool to define and implement software process improvement initiatives in a small industrial setting", *Software, IET*, vol. 4, NO 4, 2010, pp. 237-251.
- [5] H. Altarawneh, and S. Amro, "Software Process Improvement In Small Jordanian Software Development Firms", in the *7th International Conference on Perspectives in Business Informatics Research (BIR'2008)*, 2008, pp. 175-189.
- [6] D. Mishra, and A. Mishra, "Software process improvement in SMEs: A comparative view", *Computer Science and Information Systems*, Vol. 6, No. 1, 2009, pp. 111-140.
- [7] A. B. M. Isawi, "Software Development Process Improvement for Small Palestinian Software Development", M.S. thesis, Faculty of Graduate Studies, An-Najah National University, Nablus, Palestine, 2011.
- [8] M. Y. Al-tarawneh, "Harmonizing CMMI-Dev1.2 and XP Method to Improve the Software Development Process in Small Software Development Firms", Ph.D. thesis, School of Computing, Universiti Utara Malaysia, Sintok, Malaysia, 2012.
- [9] M. Pikkarainen, and V. T. tutkimuskeskus, "Towards a framework for improving software development process mediated with CMMI goals and agile practices, Academic Dissertation, Faculty of Science, Department of Information Processing Science, University of Oulu, Finland, 2008.
- [10] Z. Lina, and S. Dan, "Research on Combining Scrum with CMMI in Small and Medium Organizations", in the *International Conference on Computer Science and Electronics Engineering (ICCSEE)*, 2012, pp. 554-557.
- [11] I. Richardson, "SPI models: what characteristics are required for small software development companies?", *Software Quality*, Vol. 10, No. 2, 2002, pp. 100-113.
- [12] D. Fernandez, "Study on the correlation between CMMI and agile practices and their application in SMEs", M.S. Thesis, Computer Faculty, university Polytechnic of Madrid, Spain, 2009.
- [13] K. Beck, *Extreme programming explained:embrace change: 3th End.Reading, Mass, addition-Wesley*. Boston, 2000.
- [14] J. A. H. Alegra and M. C. Bastarrica, "Implementing CMMI using a Combination of Agile Methods", *CLEI ELECTRONIC JOURNAL*, Vol. 9, No 1, 2006, pp. 1-15.
- [15] M. Fritzsche, and P. Keil, "Agile Methods and CMMI: Compatibility or Conflict?," *e-Informatica Software Engineering Journal*, Vol. 1, No. 1, 2007, pp. 9-26.
- [16] C. P. Team, "CMMI for Development (CMMI-DEV): Version 1.3", *Technical Report, CMU/SEI-2010-TR-033*, Software Engineering Institute, 2010.
- [17] T. Galinac, "Analysis of Quality Management in Modern European Software Development", *Electronic form only: NE Eng. Rev*, Vol. 28, No. 2, 2008, pp. 65-76.
- [18] D. Goldenson, and D. Gibson, "Demonstrating the impact and benefits of CMMI: an update and preliminary results", *CMU/SEI-2003-SR-009*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2003.
- [19] G. V. Boas, A. R. C. da Rocha, and M. Pecegueiro do Amaral, "An Approach to Implement Software Process Improvement in Small and Mid Sized Organizations", in the *Seventh International Conference on the Quality of Information and Communications Technology*, 2010, pp. 447-452.
- [20] H. Mehrfard, H. Pirzadeh, and A. Hamou-Lhadj, "Investigating the Capability of Agile Processes to Support Life-Science Regulations: The Case of XP and FDA Regulations with a Focus on Human Factor Requirements", *Software Engineering Research, Management and Applications*, Vol. 296, 2010, 241-255.
- [21] R. Ramsin, "The engineering of an object-oriented software development methodology", Ph.D Thesis, Department of Computer Science, university of York, UK, 2006.
- [22] D. Vasiljevic, and S. Skoog, "A Software Process Improvement Framework for Small Organizations", M.S. thesis. Department of Software Engineering and Computer Science Blekinge Institute of Technology, Sweden, 2003.
- [23] J. Ralyté, R. Deneckère, and C. Rolland, "Towards a Generic Method for Situational Method Engineering", in the *15th International Conference Advanced on Information Systems Engineering (CAiSE2003)*, 2003, pp. 95-110.
- [24] T. Rout, (project manager), *SPICE: Software Process Assessment-Part 1: Concepts and Introductory Guides*, 2002. Downloadable from <http://www.sqi.gu.edu.au/spice/suite/download.html>.
- [25] S. Birisci, M. Mentin, and M. "Karakas, Prospective Elementary Teacher's Attitudes Toward computer and internet use: A Sample from Turkey", *World Applied Sciences Journal*, Vol.6, No. 10, 2009, pp. 1433-1440.
- [26] B. A. Kitchenham, "Evaluating software engineering methods and tool part 1: The evaluation context and evaluation methods", *ACM SIGSOFT Software Engineering Notes*, Vol. 23, No. 5, 1998, pp. 11-14.

- [27] E. J. Garrity and G. L. Sanders, "Dimensions of information systems success," Information systems success measurement, Idea Group Publishing (IGP), Hershey, pp. 13-45, 1998.
- [28] M. Easterby-Smith, R. Thorpe, and A. Lowe, "Management Research: An Introduction," Sage Publications Ltd, London, 1991.