Dean of Graduate Studies
Al-Quds University

A Framework for Small Software Development Workgroups

Sami K. M. Qasem

M.Sc. Thesis

Jerusalem-Palestine

1436 / 2015

A Framework for Small Software Development Workgroups

Prepared By:
Sami K. M. Qasem

B.Sc.: Computer Science-Al-Quds University-Palestine

Supervisor: Dr. RaedZaghal

A thesis Submitted in Partial fulfillment of requirements for the degree of Master of Computer Science / Department of Computer Science / Faculty of Graduate Studies – Al-Quds University.

1436 / 2015

Al-Quds University
Deanship of Graduate Studies
Computer Science Department

Thesis Approval

A Framework for Small Software Development Workgroups

Prepared By: Sami K. M. Qasem
Registration No: 20913400

Supervisor: Dr. Raid Zaghal

Master thesis submitted and accepted. Date: / /
The names and signatures of the examining committee members are as follows:

1- Head of Committee: Dr. Raid Zaghal	Signature: ……..…..
2- Internal Examiner: Dr. Rashid Jayousi	Signature: ……..……..
3- External Examiner: Dr. YousefAbuzir	Signature: ……..……..

Jerusalem – Palestine

1436 / 2015

DEDICATION

This work is dedicated…

First Of All To my parents, for all the support and love that they provided

To my family, brothers and sisters

To friends and colleagues

To all those that my success is their concern

Sami K. M. Qasem

DECLARATION

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed……..

Sami K. M. Qasem

Date: / /

ACKNOWLEDGMENTS

First and foremost Praise be to the Almighty Allah, the Most Merciful, The Most Compassionate for His graces and blessings throughout all my life. Without His consent and acceptance this research as well as any other thing in the life wouldn't come to reality, and wouldn't become a fact.

My sincere thanks are provided to my respectful supervisor Dr. Raid Zaghal, for his effort and patience that passed the limits of an academic supervisor and reached the area of friendship and brotherhood, his follow-up was the fuel that that this research needs in order to proceed .

Dr. Raid is a member of a great team of professors at Al-Quds University-Computer Science department, I am very grateful to all of them for the time and concern that they provided, every one of them was a supporting factor in a way or another, their advices and recommendations are fully appreciated.

I would like also to thank my Parents, Brothers and sisters, Wife and sons for their support that I needed and for the time that I took from their share to give to this research work.

ABSTRACT

The “Software Process”defined as a “*set of activities to produce a software product*” has been an important area of research for decades, improving this process has been a major concern of the researches in different fields of software engineering; this involves the commitment to several practices that has been organized into what became known as *Software Process Improvement Methodologies* (e.g. CMMI, Six Sigma, and ITIL). Software can be developed for large scale or small-scale companies and in small workgroups as well. Small software development workgroups usually do not benefit from these large-scale methodologies, They cannot adopt such methodologies since they cost a lot in terms of money, time and effort;This causes these workgroups or teams to be working in a more casual and non-systematic environment. Employees within these complicated methodologies usually involve working in a disciplined frame and familiarity with their terminology. This research aims at creating a new framework based on the CMMI model that is simple enough to be used by a small workgroup while maintaining the benefits of the sophisticated model. Our proposed frameworkis organized in a disciplined frame with the same terminologies of KPAs of CMMI level 2. The framework has been evaluated and tested by several software engineers in real development and their feedback has been collected. Our results show that such framework is positively accepted and used by SoftwareEngineers in Small Workgroups.

إطار منظم لمجموعات العمل الصغيرة لتطوير البرمجيات

إعدا : سامي كامل موسى قاسم

إشراف : ا . رائد الزغل

ملخص :

عملية صناعة البرمجيات والتي يمكن تعريفها بأنها مجموعة من الأنشطة لإنتاج برنامج معيز" تعتبر احد اهم مجالات البحوث التقنية منذ عدة عقود، كان تحسين هذه العملية مصدر قلق كبير في كثير من الأبحاث في مجالات هندسة البرمجيات، وهذا يقتضيالالتزام بعدة ضوابط للأنشطة والمهام التي نظمت في ما يعرف ب (Software process improvement methodologies) أي منهجيات لتحسين عملية صناعة البرمجيات مثل CMMI ، وSix Sigma ، و TIL . و كما أنه يمكن تطوير البرمجيات في نطاق واسع وفي شركات كبيرة فانه من الممكن أيضا تطويرها في شركات صغيرة و مجموعات عمل صغير . مجموعات العمل الصغيرة هذه لا يمكنها اعتماد هذه المنهجيات نظرا لأنها تكلف الكثير من المال والوقت والجهد؛ مما يترك مجموعات العمل أو فرق العمل الصغيرة في بيئة عمل فوضوية وارتجالية في أغلب الأحيان . العمل ضمن هذه المنهجيات يحتاج إطار منظم ومنضبط بالمصطلحات والآليات الخاصة ب ، و هذا البحث يهدف الى إيجاد إطار عمل (Framework) يعتمد على تاسيط منهجية CMMI المعروفة بحيث يكون بسيطا وسهلا بما يكفي لاستخدامه من قبل مجموعة عمل صغيرة ومع حفظ كونه منظم في إطار منضبط مع نفس المصطلحات من KPAs الموجودةفيCMMI المستوى الثاني . تم تقديم إطار العمل هذا لعدة مهندسي برمجيات و تم جمع ملاحظاتهم، وتبين من هذا البحث أن هذا الإطار لاقى قبولا إيجابيا و من الممكن أن يستخدم بفعالية من قبل مهندسي البرمجيات .

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER ONE

INTRODUCTION

Software development companies usually facemany challenges whiledeveloping their software products. These challenges may be commercial, technical, managerial …etc. In order for a company to improve its capabilities to overcome these challenges it should improve its performance in areas where shortage exist; in terms of software development the company should improve its ”software development life-cycle” through commitment to several practices organized in what is known as software development process improvement (SPI) methodologies. Several factors are involved in such improvement including: software quality control, project management, meeting schedules, testing, and documentation and so on. Examples on the SPI methodologies include: CMMI [36], Six Sigma [38], and ITIL [30].

Software can also be developed within small workgroups or teams. Few members may form a team to produce software and implement an idea. An IT unit or department in an institute may be asked to design and build a specific software for that institute; these teams and workgroups are usually asked to produce a good software in terms of quality, cost, documentation and time like the large software houses, yet these small workgroups are also asked to use SPI methodologies to improve the quality of their software product.

Employing SPI methodologies during software development is very costly in terms of money; time and effort; this is proved in Chapter 3 – Needs Assessments. A Large company may invest in improving its software process methodologies; It may spend money and time in order to get such benefit;but small software workgroups do not have the capacity to do so, small software workgroup is up to 5 members as discussed in chapter 3-; The idea of this research is to propose and create a new framework that is suitable for small software development workgroups. These usually form a big sector in the software industry in general. We believe that our proposed framework will fill a gap that has been ignored so far by researchers in this field. According to our findings, in addition to enabling these small teams to produce quality software, our framework will help them manage and reduce three important resources: effort, time, and money.

1.1 SOFTWARE PROCESS AND MODELS

Software process has several definitions and can be presented in several ways, most of these definitions and explanations share a common concept and understanding, and in the simplest words, the software process can be defined as a "set of activities to produce a software product". A software process model is an implementation of these activities in a specific manner that differentiates each model from other models.

1.2SOFTWARE PROCESS IMPROVEMENT (SPI)

Software process has been an important area of research for decades. Improving this process has been a major concern of the researches in the fields of software engineering; this involves commitment to several practices that has been organized into what is known today as software process improvement methodologies. Improvementsare achieved in terms of quality and reliability of the software, the time

required to produce the system, the documentation process and so on. Examples on the SPI best methodologies include CMMI, SIX Sigma, and ITIL.

1.3Capability Maturity Model - CMMI:

Because software applications through several years became a very important aspect in our life, and its importance is still increasing as time goes on, Several Methodologies – known as software process improvement methodologies (SPI) – have been used in order to improve the process of producing software, each of these methodologies has its own features and characteristics and way of implementation, Among these methodologies CMMI [36] is widely used and considered as one of the most important SPI methodologies.

CMMI, Capability Maturity Model Integration, is an approach or model used by organizations in order to improve their performance and their process of doing their tasks, it is used not only for software development process but for other processes as well, it was originally initiated or created as a military demand for the united states government with the cooperation of SEI (Software Engineering Institute, Carnegie Mellon University), the main objective of this project in 1980s was to create a reference model to be used to evaluate the software subcontractors, For more details about CMMI see appendix 1.

1.4 BACKGROUND

1.4.1 SOFTWARE COMPANIES:

As mentioned earlier, SPI methodologies are designed for software development companies, these companies can deliver better products in less time or at least less deviation from schedule. This increases the company's credibility among customers and improves its reputation and hence increases its opportunities tocompete

in the market. Despite all of this, several companies do not go in the direction of implementing an SPI methodology [Discussed in Chapter 3 – Needs Assessment], The following factors provide enough reasons for a company not to adapt such methodologies: [35,23]:

1. **Time**: Implementing CMMI takes a lot of time and this has been addressed in researches and white papers [35,23]

2. **Cost**: The process of implementing SPI methodologies costs a lot of money due to several factors, One of them is the already mentioned – which is time – especially when remembering that "*time is money*". Another factor is the need for manpower to handle several tasks and activities within different roles [35], Also, training is a major cost factor, CMMI needs a lot of training and the employees need to be qualified to deal with such a model.

3. **Effort**: Implementing SPI methodologies incurs huge overhead including follow-up, training, commitment to certain forms and scenarios and so on.

Small scale companies sometimes wouldn't suffer and provide the requirements and costs mentioned in order to get a CMMI level, one of the researches [23] Analyzed reasons for not adopting CMMI, It was shown that 43% of the companies say that they do not adopt CMMI simply because they are small organizations, 35% because CMMI adoption costs a lot, 25% because of time reasons, and other for different other reasons, in the same research, when reasons were classified into "could not" and "should not" reasons, 58% of the companies reasons where within the "could not" category only while 20% were within "should not"

category only, and this shows that there is a need to adopt a model or discipline, but the problem with the challenges faced and represented by the factors mentioned earlier and discussed in chapter 3.

1.4.2 WORKGROUPS*:*

Software can be developed in large scale or small scale companies, but it can be developed by a small workgroup as well. This workgroup is not even classified as a company although the software produced may be considered as a large system, important system in terms of decision support, a repository reference system when considering the amount and type of data it contains and so on. The best example on such workgroups is the IT units in most large institutes all around the world, Governmental and non-governmental organizations, educational institutes like Universities, health institutes like hospitals all of these are examples of institutes that includes an IT department whose responsibilities include providing maintenance and technical support of the computer systems (hardware and software) and occasionally develop certain software.

1.5 PROBLEM AND MOTIVATION:

Workgroups that are composed of only few developers or a small team of programmers usually work without any methodology or any unified process for development; Chapter 3 shows that this is correct for most of them. Employing these methodologies in companies improved their production on several levels including quality, development time, adherence to requirements, etc. [35, 18]. Small workgroups did not implement a specific model/methodology because they felt it is not a priority. The management is not willing to sacrifice time, money and efforts for extra things that they believe are not within their core scope of work. A hospital may spend millions on its medical development but won't invest relevant amounts to

develop its IT infrastructure. Even stand-alone teams or workgroups would prefer to spend every dollar they get on initiating a business or a company rather than investing part of their revenues on improving their Software Engineering and development skills. Several studies have stated that clearly:

- *"Small software organizations are often unorganized in SPI." [6]*
- *"A frequent conception about adopting Capability Maturity Model Integration (CMMI) is that it works only for large organizations (its cost and complexity appear to make it impractical for smaller organizations to implement)."[10]*
- *"Small organizations often do not adopt software process models and metrics ... money, manpower, scale and time ..." [35]*

1.6 SUGGESTED SOLUTION:

Our proposed solution in this research is to build a framework: a collection of templates, forms, guidelines, and clear procedures based on CMMI level 2 and its key process areas and sub processes. From the needs assessment handled in chapter 3 regarding the reasons for not adopting SPI methodologies, we have identified the following requirements for our framework:

1. It should support the developers (workgroup) to focus on key practices that they would inevitably face during development and are considered a major area of concern and influence on the software production (e.g. requirement management practices), and at the same time allow them to skip non-relevant practices and activities.

2. It should be short enough to be convincing to workgroups to adopt for their work – as if it is a collection of guidelines or a booklet manual that supports development.

3. It should be simple and clear enough to be followed and used at least by getting directly to the practical point to be done, or form to be filled or something to be handled.

Adapting such a framework is supposed to take less time, cost, etc.Small development teams usually skip standards when they already know that referring to this standard will cause some delay (or extra effort) in the development process. The activities in the suggested solution should not force the developer to leave the development itself for a long time. Major delays may cause the framework to be ignored and inconvenient but still minor delays can be tolerated.

1.7 METHODOLOGY:

The objective of our research is to build a framework that is considered simple, direct, short and easy to be used for software development by a small workgroup. It is assumed that using this framework is going to improve the development process and produce better quality software. In order to design this framework, we followed the steps described herein:

1. Collect required practices and categorize them as a theoretical basic for the whole research. This involves creating or using already existing templates and forms and merging them together within the framework. We will cover the major phases of software development life cycle. For example:

a. General project planning practices (tasks, participants, schedule … etc).

b. System analysis and design (priorities, requirements, documentation … etc).

c. System development and coding.

d. Software evaluation (QA practices, user satisfaction …etc).

2. Package these activities in a well-organized manual (framework) to be followed in development phases.

3. Build an evaluation criteria, questionnaire, participation list and analysis methodology; Use (test) the framework created in step 2 in a software production environment. The framework is to be tested on an actual case. A team consisting of 2 - 3 developers is asked to create a new module and add it to a system that is already running and stable, the team is supposed to provide an offer including a deadline for delivering the module and integrating it with the system (financial issues are of course out of scope of the research), the team will do the following:

    a. Make an initial study of the model in order to get estimates of timetable entries.

    b. Use the framework to write down clearly all requirements and analysis related issues and document them in the way and order specified in the framework.

    c. Approve the detailed analysis and requirements by the customer.

d. Develop the model based on the Software development life cycle and in accordance with our framework.

e. Test the system referring to the initial documented requirements and needs.

f. Deliver the final model and integrate it with the existing and running system.

4. Extract results (readings, notes, summaries and conclusions) from the whole implementation process and create comparisons with previously created modules without using the framework. These previous modules are already done and notes and readings about them are already being written down (e.g.: deadlines are missed, some features are not done as the user require)

5. Present results in a clear and precise tools.

1.9 CONTRIBUTION:

To create a framework (SPI framework) that improves the capabilities of a small software development workgroup to improve the development cycle and produce better quality software. The framework will focus on the following (Requirement specification, Quality assurance based on requirements and project documentation).

1.10 SCOPE OF RESEARCH:

1. The research will consider and target Palestine during research and testing phases, it is not possible within the research timeframe to communicate with teams outside the country especially that size and type of work of each team to be investigated. But we can generalize our research findings to include countries in the Middle East with similar capabilities and infrastructure.

2. Small workgroup is up to 5 members: Programming teams in Palestine consist of up to 5 members in general, Chapter 3 – needs assessment investigated this issue of workgroup size.Even in companies with larger number of employees and developers when they work on specific projects they specify a team for the project that may not even reach 5 members. That's why the team size considered within the research is 5 members maximum.

CHAPTER TWO

RELATED WORK

Since Software applications are significant issues in our life, and this significance is increasing year after year, the improvement of these software applications, and their development process - software lifecycle process- have been also an important concern of experts working in the field in order to avoid failure of such applications, this failure may cause major damages and losses that are harmful enough to be worried about. One of the articles [7] says that software application failure may sometimes cause loss of life: "*Software artifacts, even small programs, are among the most complex man-made products, and software development projects are among our most complex undertakings, Nowadays, our lives is governed by computers, communications and computer based systems, Failures of such systems lead to considerable economic losses or even to the loss of life.*". The same article discusses how the software development process has been improved through history of programming and software engineering, it states that at the early stages of software development software engineers depended on the end of cycle quality inception, this means that after the software application has been produced it is evaluated to check whether or not the application is appropriate to use. And after that the software process also became an important concern and then lifecycle and process models like CMM had been designed and used and became involved in the development process. The process of development the software is the concern of the framework included within our research, it is designed to help software development teams to improve their process of development their software products.

Another research [13] has proposed and investigated the various models of Software process improvement. Each of these models has its own features and

characteristics, Adopting any model in any small and medium scale enterprise is up to the enterprise itself and how the features of that model is compatible with the enterprise strategies, goals and objectives. However it was stated in the end of the research that adopting any of these models is so hard in small and medium enterprises: *"Although numerous SPI standards and models have been proposed, their adoption among small organizations is hard due to some size mismatches and to lack of experienced process engineers*". Our framework is to bypass this issue of being hard to adopt, simplification is the core issue in building the framework components.

Researchers have worked in different countries and areas of the world; sometimes the region focused in a research affects the results of the experiments handled. This means that if the experiment is applied somewhere else different results may be recorded, This makes it inappropriate to generalize the results, one of the papers [25] discussed the implementation of a project in a specific region (Walloon region - the French speaking part of Belgium) and discussed also characteristics of that region that makes the results of the experiments only focused to that region *"slow conversion to modern industrial structures, the persistence of some old-fashioned bureaucratic management style, region surrounded by rapid growing dynamic regions* ". Another study regarding application of software process improvement in small enterprises has been published [4]. This study presented a framework and its comparison with CMMI and targets the same previously mentioned region (Walloon or Wallonia), and assumes that number of members in small enterprises is up to 60; 60 employees may be considered a large enterprise in other regions. A paper that was created in Chili [33] focuses on Chili and assumes that results may apply for other Latin American countries having the same scenarios. Its definition of the size of the project is based on the duration of the project and not the number of employees in the

team (1 - 2 months: small project,3 - 6 months: medium project). Also the process explained in the study demands six roles within the development group (manager, analyst, designer, programmer, tester and user/client).

Also there was a research regarding Software ProcessImprovement for Small and Medium Companies that focuses on web development companies [24]; it includes a discussion of differences between Web development and traditional development. The designed framework is somehow generic in terms of web or traditional development; a later phase may distinguish between both tracks if it is found that there is a need to do so.

Several researches have discussed the idea of implementing and applying SPI models in big companies and large size enterprises and all related issues. Some of these include [19, 5, 12]. In our research, we focus on small software development workgroups, these workgroups may be found in it departments in different institutes, or in small companies or in other different locations and aspects.

Another area of research is accommodating the SPI models into small companies or organizations. Most of these works have pointed out that SPI models like CMMI, six sigma, etc., are not suitable for small companies and need some kind of manipulation in order to be adopted. Still when they refer to small companies they include those of a large number of employees (up to 100). But compared to small workgroups in Palestine (or similar developing countries), we consider a workgroups of up to 5 people as a “small” workgroup.

Other researches discussed implementing SPI frameworks for small workgroups but did not discuss the relevant details of their framework [18, 8]; only results of implementation were listed. Positive impact shown in the results motivates

and encourages the researchers to create similar SPI frameworks and implement them in their different environments, and this is what we have done in our research.

In 2009 a Turkish research [9] involved the creation of a questionnaire-based method that helps a software company to adopt CMMI level 2 and facilitates its implementation. The companies targeted in this research are companies that are in their way of implementing CMMI level2 and this method - in the research – is used to assist in this, the research also included a discussion of the results of this assessment method.Our framework is not an assisting factor in implementing CMMI neither it is an alternative to CMMI, If a development team is approaching towards adopting CMMI, then this is fine and no need for our framework, Our framework is designed for those not intending - at least currently – to implement any SPI.

Other researches discussed the usage of SCRUM [21] as an approach within the SPI area, and there has been a focus on the relation between Scrum and CMMI and the presentation of a mapping between CMMI and scrum [16, 1]; also there has been studies regarding implementing CMMI and scrum together [26] and how implementing scrum with CMMI affects the implementation of CMMI [2]. Also there has been an investigation of the assessment of compatibility of scrum with CMMI [34].Scrum focuses on the human resources of the development team and on team members and individuals, our framework components are related to the process of development software and not the team or members, although sometimes the relation between both process and members is addressed in approvals or consultation or wherever necessary.

Also, Other researches compared small and large organizations [15], small organizations according to them are organizations fewer than 50 members, their research included the similarities between both of them like goals and objectives and other issues, and also includes the differences, the research says that they differ in their structure and the management style and sometimes small organizations adopt tools that Assist in implementation of large SPI methodologies like CMMI. Our framework is not a helping tool to another SPI methodologies and it is targeting teams with 5 or less members, and it doesn't discuss differences and similarities between small and large organization as a core subject of the research, differentiating between small and large organizations was an entrance to our research with SPI as its focal issue.

In a related research [32] the authors investigated if CMMI is useful and usable in small settings. The paper shows that it was tested in 2 pilot sites (10s of employees in each) and that several challenges were seen during implementation, this includes the financial issue and human resources, our research mentioned such challenges as reasons for creating a new framework that is suitable for small organizations and implementing it in more than 2 pilot sites with fewer employees.

In 2005 in Pittsburg, there was an international research workshop for process improvement in small organizationsone of papers discussed a software process model designed for small enterprises (less than 100 employees) and called MoProSoft [14], this model was built in in Mexico, the paper mentions that before defining the structure of the process model the structure of the enterprises was analyzed, the processes in the Software process improvement model were categorized into three groups, top management, management and operations. The framework we worked on is not built on the same criteria, categorizing processes and roles and human resources

into three management levels do not apply for teams of few members only, our framework categorizes the processes in process areas similar to those in the second level of CMMI in a way that development process can be handled step by step without any consideration of management level

Another research [22], suggested a different system approach in software process improvement in small organizations, it includes a description of the human health and how it can differ from time to time according to person age and life phases, and also discussed the cultural side of humans, the authors suggest that it is possible to apply the human health model to software process improvement model, it is possible to map objectives of human life and symptoms and stages to corresponding areas in software development. Our research does not rely on such criteria, it doesn't map the steps to be carried over during development to any human culture or medical psychological situation, it depends on the steps to be processed in each phase and building better model implementation every time a project is implemented.

CHAPTER THREE

NEEDS ASSESSMENT

As mentioned earlier in the introduction, Software can be developed in huge software development companies and in small software development workgroups, in this research, the term big or small when describing software workgroups means number of members working within the group. Also it was mentioned that there are several challenges while employing any SPI methodology in developing software, and these challenges are the reasons for not implementing any SPI in most of the small software development workgroups (as assumed). In this chapter of the research these three factors and assumptions (Size of workgroups, SPI implementation, and reasons for not implementing any SPI) are to be investigated in order for the study to be built on a scientific base.

Also, this research includes the distribution of an evaluation form or questionnaire on different participants to provide a feedback regarding the proposed framework, the framework - as described – should be simple to understand and able to implement and at the same time should consider the methodology used as a reference in this research (CMMI Level 2) and preserve its terminology and its systematic approach and discipline. Analyzing the collected feedbacks and evaluations requires categorizing the questions of the evaluation into these key points (Simplicity, Implementation, Terminology, and Systematic Approach). This creates another factor to investigate which is mapping questions of the evaluation form into these categories. And this is the fourth factor besides the three mentioned previously (Size, Implementation, Reasons for not implementing).

In order to investigate these four factors; A specific questionnaire was created and distributed on the same participants list, Feedbacks for this questionnaire were collected and arranged into excel sheet in a way to get the final results in a numeric presentation and percentage ratios, The following is a discussion of each of the factors and its measurement:

1. **Evaluation Questions Categorizing:**

The framework evaluation form – discusses later - includes 15 questions; these questions are to be classified into groups; question 10 and question 11 are excluded from this classification, question 10 is related to the participant linguistic skill and question 11 is related to whether an SPI is implemented or not in the company; both questions are not related to the suggested implementation of the framework.

Each participant was supposed to put a mark under the appropriate choice for each question referring to the four choices:

Table 3.1: Choices for Classification of evaluation questions

1	Simplicity.
2	Systematic approach and discipline.
3	Terminology and its preservation.
4	Ability to implement the framework.

The following table shows the feedback collected from the participants, for each question the percent of participants under each choice and the final column shows the class that was mostly chosen, e.g. for question 1 92.9% of the participants chose class 3 :

Table 3.2: Results of feedback collected for Classification of evaluation questions

Highest Choice	4	3	2	1	Question
3	0.0	92.9	0	7.14	1
2	21.4	0	71.4	7.14	2
4	100.0	0	0	0	3
1	0.0	0	0	100	4
4	64.3	0	0	35.7	5
4	85.7	0	14.3	0	6
1	7.1	0	0	92.9	7
2	14.3	0	85.7	0	8
4	92.9	0	0	0	9
	0.0	0	0	0	10
	0.0	0	0	0	11
3	0.0	92.9	0	0	12
2	0.0	0	92.9	0	13
2	0.0	0	85.7	7.14	14
2	7.1	0	85.7	0	15

From the table shown above, the questions of the evaluation are classified in categories as shown below.

Table 3.3: Classification of evaluation questions

No	Category	Questions
1	Simplicity.	**4,7**
2	Systematic approach and discipline.	**2,8,13,14,15**
3	Terminology and its preservation.	**1,12**
4	Ability to implement the framework.	**3,5,6,9**

2. **Size - Number Of members in the workgroup:**

Regarding number of members in the workgroup or company, also there were four choices from which the participant chose when answering the question of size of the working team (1-5, 6-10, 11-15, and more than 15). The results collected showed that the size of (1-5 members) was chosen by 57.1% of the participants while the size of (6-10 members) was chosen by 42.9% of the participants.

3. **Implementation of Any SPI:**

For this factor, the questionnaire included a direct question of whether any SPI methodology was implemented in the participants' institute or not. 85.7 % of the participants chose answered by (NO), there is no implementation of any SPI in 85.7% of the participants' institutes, only 14.3 % participants answered by (YES) and this shows that most of the software workgroups or teams included in this research do not implement any SPI methodology.

4. **Reasons for not implementing Any SPI:**

It was discussed in chapter one that there are three reasons for not adopting any SPI methodology in software development (time, effort, and cost), this section of the research investigates whether these reasons apply in our targeted institutes or not, a question was added to the questionnaire to ask what are the reasons for not adopting any SPI and 5 options were listed as shown:

Table 3.4: Reasons for not adopting any SPI methodology

No	Reason	Choice
1	Time – SPI implementation needs a lot time	**( )**
2	Effort – SPI needs huge Effort to implement	**( )**
3	Cost – SPI implementation costs a lot of money	**( )**
4	Need – Implementing SPI is not needed	**( )**
5	Other	**( )**

The results collected for this question showed that the same reasons applies for the targeted institutes by small differences in percentages as shown in the table below, answers 4 and 5 and discussing them is not within the scope of this research.

Table 3.5: Reasons for not adopting any SPI - Results

No	Reason	Result
1	Time – SPI implementation needs a lot time	**32 %**
2	Effort – SPI needs huge Effort to implement	**28 %**
3	Cost – SPI implementation costs a lot of money	**20 %**
4	Need – Implementing SPI is not needed	**12 %**
5	Other	**08 %**

CHAPTER FOUR

FRAMEWORK

Building the framework is based on the components and structure of the CMMI model, the subsections or KPAs of CMMI are to be considered while creating the framework, even the sequence of processes and steps in the CMMI are supposed to be taken into consideration. Since Processes of the CMMI model are grouped into what is known as Key Process Areas (KPAs), every step in the framework should be in turn grouped into a corresponding section or group as the KPA in CMMI.

The simplification process is not direct and clear enough to handle, there is no documented criteria for what to omit or what to keep in the new framework. That's why the best method to use is referring to professionals and experts in the field, especially those included in the participation list (the list prepared for evaluation).For each process in the CMMI, and while creating a corresponding step in the framework, there was a direct contact with these references, they were consulted and asked for guidance and recommendations, their feedback was the base on which simplification was accomplished. For example most of the consulted participants - Whether in IT departments or development groups within a software company - agreed that there is no need for any financial issues documentation, financial issues (Price estimate of the software) are not an influencing factor in the software development process for most of them, this is mainly because when the team of an IT department is asked to build a software application for the institute, there is no price to be paid for that application, the development team is employed in the institute for such task.

On the other hand, Requirement follow up was inevitable since all of the participants assured that it is an important start for all the following steps. There were some differences in the feedback in the way of dealing with requirement management and its details, when there was an equal contradiction between feedbacks regarding including a specific point, it was summarized and a new feedback was asked in order to assure that it is appropriate to include that point in its summarized way or not. The approval of a requirement for example was documented by using only a specific portion within the requirement where it is written (approved by whom and date of approval) this is instead of other documentations of approval with stakeholders.

Sometimes the difference in feedbacks existed in big sections like "planning for data management", some participants said that it is not needed and it may form an additional effort to work on it, while others said that it may be useful if used, That's why this section as a whole was described as (optional), there is no relation between this section and any other section, which means that if this section is not used then it will make no effect on other section or processes that may be built on the results of "planning for data management section". Except of course the section "monitoring data management" which is also optional since it based on the planning section of data management, or even it is better to say that monitoring section is a complementary section for the planning one.

In some portions of the CMMI there was a need for meetings for development teams, the existence of those in higher positions is also required, e.g. team leader, project manager, and so on, A lot of participants provided within their feedback that there is no need for such complications, they were only 3 or 4 developers that are always working within the same office or department, with a department manager.

This Chapter discusses step by step the components of building the framework from the CMMI model - level 2 in maturity, the purpose subsection for each section *"written in italic"* is cited from the CMMI model Itself, followed by an introduction then the steps of KPA to framework, (Key Process area to framework).

4.1 REQUIREMENT MANAGEMENT - REQM

4.1.1 PURPOSE

"The purpose of Requirements Management (REQM) is to manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products."

4.1.2 INTRODUCTION

A decision to build or buy a software system is based on the actual need to do things better in terms of several factors that are out of scope of this research, the software is never assumed as a decorative issue or a kind of accessories for the office, there are several important things that this system should do, and many functions that it should handle, each of these functions is required to work correctly and as expected and planned for, the analyst and designer of the system should do his best in order to identify these requirements correctly, developers should do their best to implement these requirements in the best way, the testers of the application should test the availability of these requirements in the application, the users will use the system to meet the requirements that are originally identified. That’s why requirements are the reference for every one in every phase in the development process of the application, managing these requirements properly will assist in producing good quality software

and vice versa. Good Requirement management includes documentation, linkage, assignment, dependency considerations, and commitment.

4.1.3 KPA TO FRAMEWORK STEPS

4.1.3.1 STEP ONE: STANDARD FORM, INITIAL AND MINIMUM

Product Requirements represent what that product should do and what functions it should perform, these functions should be stated and written down in a formal way to be followed up later on within the product development phases, formal documentation involves the specification of several attributes related to each function; these attributes include:

- **Function**: Name or title of the function, it is short and direct few words are enough, it starts with an imperative or order (e.g. change employee degree, calculate overtime … etc).
- **Description**: Explanation and elaboration of the function, the name of the function may not provide enough information about what the function does, or what it include or exclude, or who is targeted or not within the function (e.g. Assign an employee a different degree than the current one, this doesn't apply for employees under probation period.)
- **Other information**: e.g. input, source, output, destination and action.

Header of the function specification is always included it contains the project name, module name and sub-module name. in the case of a small team or workgroup working on a small to medium scale projects no need for such elaboration this framework is going to be used for one project only, no need to write the project name on every page while it is always the same project, several modules may exist in the same project that's why module is included in the header (e.g. sales, Purchases …etc),

even if sub-modules exist a sub-module may be integrated within the module itself (e.g. module local sales and module regional sales …etc). Also the function should be assigned a code, in the standard form of requirement specification this code is used for reference and archiving, in this framework this code is to be used frequently in different phases for linking, dependency, and testing.

Additional attributes may be included in the function specification according to nature of project or even the function itself; it may include for example Pre and Post Conditions, side effects, special circumstances and so on. These attributes are all stated in a standard form-like specification as shown below:

Module	**FunctionCode**
Function	Change Employee Degree
Description	Assign the employee a different degree than the current assigned one, doesn't apply for employees under probation period
Input	…
Source	…
Output	…
Destination	…
Action	line1 line2 line3

Figure 4.1 Initial Requirement

4.1.3.2 STEP TWO: SP 1.1 – OBTAIN AN UNDERSTANDING OF REQUIREMENTS

As stated before, a requirement should be identified and written properly and designed well in order to get the best results, properly or not and well or not depend on several factors and measurements, some of them are general for every project and case used and some others are specific to the project itself:

- **General**: in any project the requirement/function statement should be complete, clear, and so on.
- **Specific**: in a project the requirement/function statement may have special measurements related to that project only or even on the module level, e.g. in a financial system every requirement should not contradict with financial rules and basics.

Both general and specific types are considered to be part of the proficiency and knowledge of the analyst or designer, within the boundaries of this issue designer experience plays a bigger role than the framework used, this framework assures that its user has built a measurement criteria for the quality of the requirement specification through collecting the factors and measurements to be used; a special reference sheet is prepared containing general factors and a place for filling other specific factors, and a checklist in the requirement specification template is added in order to match the requirement written with the criteria built, see figures below:

Table 4.1: Reference sheet for requirement specification evaluation criteria

Requirement specification Characteristics evaluation criteria			
No	**G/S**	**Name**	**Description**
1	G	Complete	No …….
2	G	Clear	function shouldn't be understood in more than one way
3	G	Consistent	
4	G	Verifiable	
5	G	.	
6	G	.	
7	G	.	
8	S	GAP Met	
9	S	Gov. Rules	
10	S	.	
11	S	.	
12	S	.	

Module		**Function Code**	
Function	Change Employee Degree		
Description	Assign the employee a different degree than the current assigned one, doesn't apply for employees under probation period		
Input	…		
Source	…		
Output	…		
Destination	…		
Action	line1 line2 line3		

Quality Criteria		
1	Complete	
2	Clear	
3	Consistent	
4	.	
5	.	
6	.	
7	GAP Met	
8	Gov. Rules	
9	.	
10	.	

Figure 4.2: Requirement Specification Form with quality criteria added

4.1.3.3 STEP THREE: SOURCE AND APPROVAL.

So far 2 sub-practices of SP 1.1 are covered, but still the other 2 that are related to the person from which the requirement information was taken, A requirement may be stated correctly, clear and appropriate but the problem is that it may be taken from the wrong person, this will cause a future problem although the requirement is well understood, written and revised but from the point of view of the improper person and based on his understanding, that why the second issue to assure the correct obtaining of understanding is to clarify the source from which the requirement was taken and approved by the contact person responsible for that module or the whole. The next form (table) shows the modified form with source and approval added:

Module		**Function Code**	
Function	Change Employee Degree		
Description	Assign the employee a different degree than the current assigned one, doesn't apply for employees under probation period		
Input	…		
Source	…		
Output	…		
Destination	…		
Action	line1 line2 line3		

Quality Criteria		
1	Complete	
2	Clear	
3	Consistent	
4	.	
5	.	
6	.	
7	GAP Met	
8	Gov. Rules	
9	.	
10	.	

Req. Source		**Date**	
Approved By		**Date**	

Figure 4.3: requirement specification form with requirement source and approver info.

4.1.3.4 STEP FOUR: SP1.2 – OBTAIN COMMITMENT TO REQUIREMENTS

Once requirements or functions are clearly stated by the system analyst in the team, they should be revised with the other members of the team, each member should understand what the requirements are about and how they are going to be developed, within a small meeting, each member may say that this function can be done this way, or that function can't be done at least within the available resources and experiences, or who can be contacted in order to assist in a specific function and so on. The point is that at the end of that every function if approved should be committed to by at least one member that is capable of handling it, may be all members can do that task but one of them is considered as a specialist in such a field, committed members (staff) can be written and notes can be used. See fig (table) below, the 2 sub-practices in this practice discuss this issue, the impact of other commitments are not so significant in this framework since the group or working team is not assumed to be working on wide number of projects, or dealing with so many customers and commitments as the situation is with employees working in the large scale companies:

Module		**Function Code**	
Function	**Change Employee Degree**		
Description	Assign the employee a different degree than the current assigned one, doesn't apply for employees under probation period		
Input	…		
Source	…		
Output	…		

Quality Criteria		
1	Complete	
2	Clear	
3	Consistent	
4	.	
5	.	
6	.	
7	GAP Met	

Destination	…			8	Gov. Rules	
Action	line1 line2 line3			9	.	
				10	.	
				Committed staff		
Req. Source		**Date**		1	All	
Approved By		**Date**		2	..	
				3	..	
				4	..	
				Notes		
				1	Reference – Joe	
				2		

Figure 4.4: requirement specification form with Commitment section added

4.2 PROJECT PLANNING - PP

4.2.1 PURPOSE

"The purpose of Project Planning (PP) is to establish and maintain plans that define project activities."

4.2.2 INTRODUCTION

Project management is a core activity within the software development process; if the process is not managed correctly it usually fails even if it appears as if it succeeded, success is not measured only by answering “is the product delivered?” a lot of other questions are supposed to be answered, how it is delivered? When is it delivered? How long did it take? Is it good enough or suitable for required work? Does it meet needs and requirements? A lot of factors determine if the project failed or succeeded or how much is it considered succeeded.

Project management starts with planning, let always remember “Plan your work then work your plan”; this exactly what happens in good project management process, a good plan is constructed and then every activity falls within the frame of this plan.

Before starting the process of building the plan, the architecture and size of measurement should be clarified, The architecture of the project may be (Client server, Web application, a mobile application … etc), while the Size measurement units may be (line of code, basic form, basic report, query … etc).

4.2.3 KPA TO FRAMEWORK STEPS

4.2.3.1 STEP ONE: SP 1.1 – ESTIMATE THE SCOPE OF THE PROJECT

In project management field the term scope refers to the set of activities or tasks to be done, or work accomplished in order to produce the required product, Although this is not a formal definition but in a way or another this is in simple words what the

scope of the project means, in order to get an estimate of the scope of the project all tasks or activities or so called “work packages” are to be listed in a specific way with specific features in order to be considered as a clear reference and to be useful in their usage. The way that these work packages are represented in is the “Work Breakdown Structure – WBS”.

Sub practice 1 (WBS): When software process teams work on their projects, weather they know it or not, they start their work with designing the “Work Breakdown Structure - WBS”. WBS is a breakdown of the overall project required function to be done into smaller components that can be translated into tasks and activities later on, there is no specific limit of levels when subdividing the components of the project but 3-4 levels is usually enough. Figure 4.5 show how a WBS is built and how a project is divided into smaller tasks and every task is broken down into smaller subtasks. Figure 4.6 show an example of a WBS for an HR system and focusing on one of the tasks, HR system includes handling employment, payroll and annual increase, employment process in turn is broken down into the subtasks of preparing the parameters to be used while evaluating applications for a vacancy, then the task of filling applications, then the task of evaluating applicants according to the specified parameters and so on.

WBS, can be presented in more than one way, the well-known tree structure or form as shown in the figures 4.5 and 4.6 and also in a tabular form, the tree structure is more useful for understanding and follow up while the tabular form is more useful in manipulating issues and that's why both are important and used, start the WBS table as follows:

Table 4.2: Tasks and Subtasks

Task	Name	Sub task	Name
1	Handle …	1.1	Prepare. …
		1.2	Fill ….
2	Manage…	2.1	…

Sub practice 2 (sufficient details) – make sure that every work package or final leaf in the tree (sub task) can be assigned to a requirement.

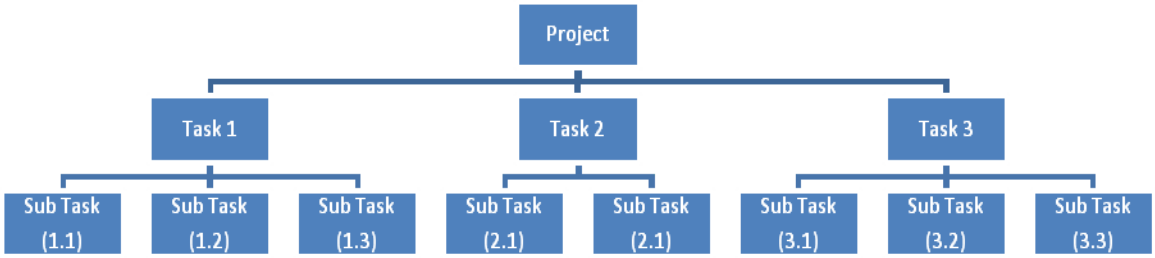


Figure 4.5: Work Breakdown Structure

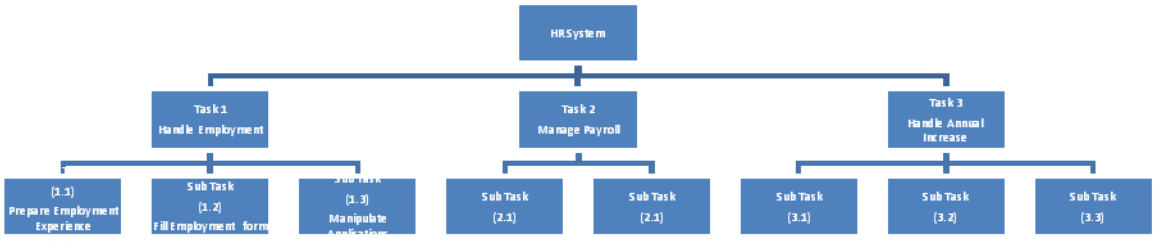


Figure 4.6: Work breakdown structure example

Sub practice 3 (externally acquired) – for every (work package) R1 is added to include weather it is externally acquired or not. Externally acquired means that it is not to be handled here only included. If it is to be handled in the same project but not within this portion of the project, it is to be acquired from another task of the project the corresponding task number is written in region 2 (example punctuation - number to words)

sub practice 4 (reused) – for every (work package) R3 is added to include weather this task when finished is to be re-used in another place of the project if yes region R4 is to include tasks that are to use this subtask.

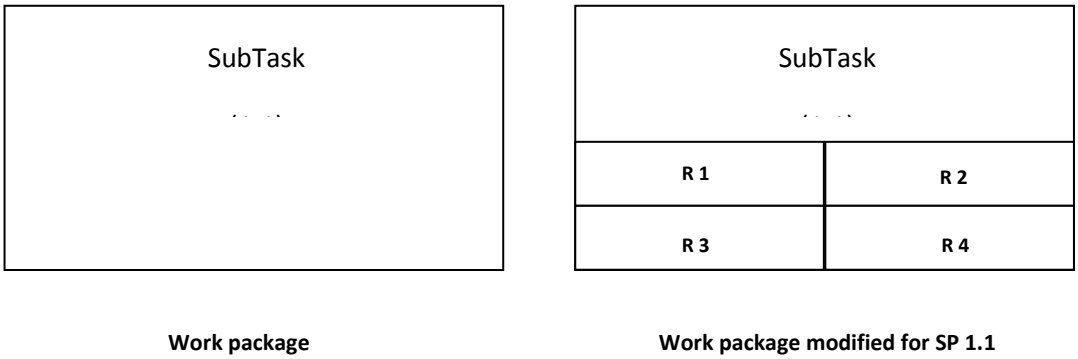


Figure 4.7: Work package for SP 1.1

Table 4.3: Work package table structure for SP 1.1

Task	Name	Sub task	Name	Is Ext Acquired	Where	To Be Reused	Where
1	Handle	1.1	Prepare. …				
		1.2	Fill ….				
2	Manage	2.1	…				

4.2.3.2 STEP TWO: SP 1.2 – ESTABLISH ESTIMATES OF WORK PRODUCTS AND TASK ATTRIBUTES

Sub practice 1 and 2: are considered in the starting phase of project planning and related to the issue of clarifying the unit of measure.

Sub practice 3 (estimate the attributes): as indicated in the graph, in region R 5 record the estimated size of the task (example 3 forms, 5 reports, 1000 LOC … etc)

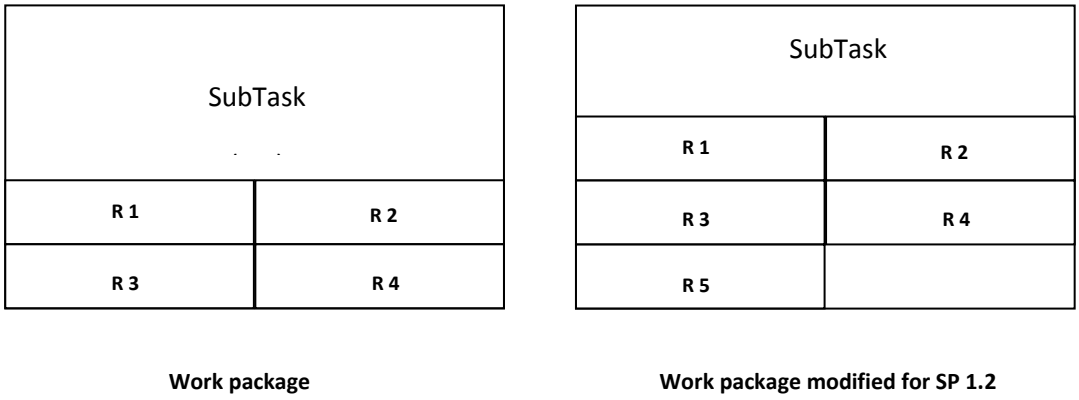


Figure 4.8: Work package for SP 1.2

Table 4.4: Work package table structure for SP 1.2

Task	Name	Sub task	Name	Is Ext Acquired	Where	To Be Reused	Where	Estimated Size
1	Handle	1.1	Prepare. …					
		1.2	Fill ….					
2	Manage	2.1	…					

4.2.3.3 STEP THREE: SP 1.3 – DEFINE PROJECT LIFE CYCLE

A Project passes several phases to be completed, these phases should be written down and explained, when working within a phase all its tasks are to be done and tasks in other phase should not be considered in current work, overlap in work if not managed well will cause a mess in progress; example: Development phase, Testing

Phase, Implementation Phase, finalizing phase and so on. These phases are to be documented in the table shown below:

Table 4.5: Project Lifecycle (Phases)

No	Name	Description	Remarks
1	Development Phase		
2	Testing Phase		

4.2.3.4 STEP FOUR: SP 1.4 – DETERMINE ESTIMATES OF EFFORT AND COST

Sub practice 1 (Historical Data): The historical data to be collected includes similar tasks and there estimated and actual measurements in terms of size (complexity) or time. For this framework a table is to be used for the current project to fill in data to be used for later projects as historical data. The importance of this step of recording and documenting readings, measures, actual time used and so on is at least the importance of historical data since process improvement is built on historical measures and considers them a core in the improvement itself, the table to be filled is shown below:

Table 4.6: Log table (enhanced from PSP: estimated time per unit e.g. 10

	For all tasks					For tasks estimated as 1 unit				
Task	**Estimated No of units**	**Est. Time**	**Actual Time**	**Unit Act. Time**	**Unit Error**	**Single unit? (1/0)**	**Est. Time**	**Act. Time**	**Unit Error**	**Remarks**
1	**1**	**10**								
2	**1**	**10**								

Table 4.7: Historical Log table from previous projects: estimated time per unit e.g. 10

	For all tasks					For tasks estimated as 1 unit				
Task	**Estimated No of units**	**Est. Time**	**Actual Time**	**Unit Act. Time**	**Unit Error**	**Single unit? (1/0)**	**Est. Time**	**Act. Time**	**Unit Error**	**Remarks**
1	**1**	**10**	**8**	**8**	**2**	**1**	**10**	**8**	**2**	
2	**1**	**10**	**9**	**9**	**1**	**1**	**10**	**9**	**1**	
3	**2**	**20**	**19**	**9.5**	**0.5**	**0**				
4	**1**	**10**	**7**	**7**	**3**	**1**	**10**	**7**	**3**	
5	**5**	**50**	**45**	**9**	**1**	**0**				
6	**3**	**30**	**35**	**11.9**	**-1.9**	**0**				
7	**1**	**10**	**11**	**11**	**-1**	**1**	**10**	**11**	**-1**	
8	**2**	**20**	**18**	**9**	**1**	**0**				
9	**2**	**20**	**21**	**10.1**	**-0.1**	**0**				
10	**4**	**40**	**38**	**9.5**	**0.5**	**0**				
	Sum	**Sum**	**Sum**	**Avg.**	**Avg.**	**Sum**	**Sum**	**Sum**	**Avg.**	
	22	**220**	**211**	**10.4**	**0.6**	**4**	**40**	**35**	**1.25**	

- There are 4 single unit tasks so in non-single tasks there are (22-4 = 18) units Estimated: (18*10=180) units of time → Actual (211-35=176) units of time
- Accuracy (for non-single unit tasks) = 176/180 = 0.98 → error = 1-0.98 = 0.02
- Accuracy (for all tasks) = 211/220 = 0.959 = 0.96 → error = 1-0.96 = 0.04
- Accuracy (for single unit tasks) = 35/40 = 0.875 = 0.88 → error = 1-0.88 = 0.12
- Error in estimation of no of units = (0.02 – (0.02*0.12)) = 0.0175 → Accuracy = 1-0.0175 = 0.985 = 98.5%

Sub practice 2 (infrastructure needs) : infrastructure needs and time estimates are to be included in this phase, write down in region R6 the estimated time for preparing the infrastructure required in terms of hours or days according to measurement unit. And the descriptions and details stated and clarified in the work packages table described later.

Sub practice 3: Converting all size estimates into timely estimates using the formula:

Size (complexity) * unit time cost + supporting needs time cost

Example:

A master form is estimated as 4 standard forms each standard form requires 3 hours of work and supporting needs 2.5 additional hours

4 * 3 + 2.5 = 14.5 Hours

Result value should be written in Region R7

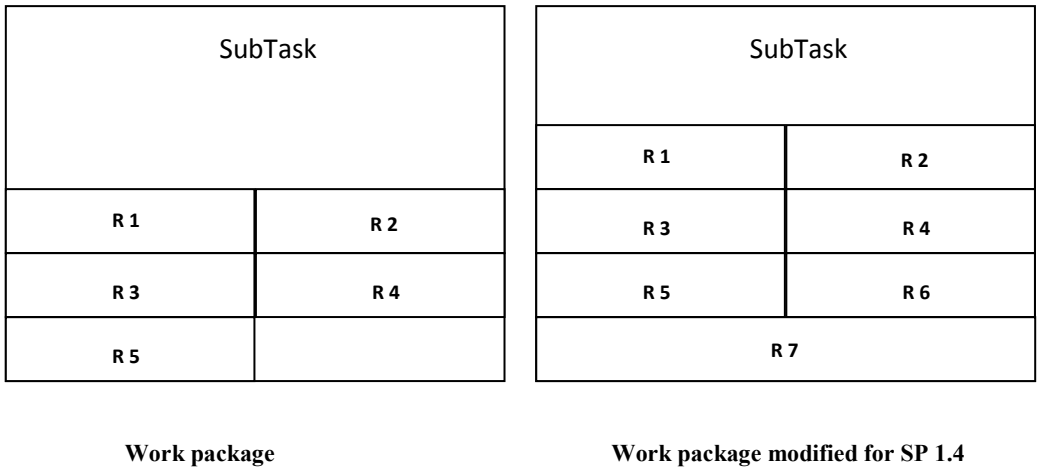


Figure 4.9:Work package for SP 1.4

Table 4.8: Work package table structure for SP 1.4

Task	Name	Sub task	Name	Is Ext Acquired	Where	To Be Reused	Where	Estimated Size	Infra Struct. Time	Total Time
1	Handle	1.1	Prepare. …							
		1.2	Fill ….							
2	Manage	2.1	…							

4.2.3.5 STEP FIVE: SP 2.1 – ESTABLISH THE BUDGET AND SCHEDULE

Sub practice 1, Major Milestones:

Milestones, Calendar based milestones are often used to organize and specify schedules at which the participant should meet and follow up what they are assigned to do separately, and discuss issues they faced, how they passed obstacles and so on, in a small team or workgroup the members are always in a continuous meeting no need to schedule a timely based meeting since they are always together, when there are things need to be discussed "loudly" this is done on the spot.

On the other hand event based milestone are important, if a member in the team is not complaining to colleagues, not discussing or arguing then things are supposed to be working fine and the proper time to meet is when an event is triggered, it may be the completion of a task, the occurrence of a risk, or even passing a risk, all of these are case dependent, milestones are filled in a special milestones table:

Table 4.9: Project Milestones

No	Name	Description	Date	Event	Remarks
1	A cycle is finished				
2	A risk occurs				

Sub practice 2, Schedule Assumptions:

Assumptions are inevitable in project planning, when analyzing a special case requires the existence of an expert that is assumed to arrive 2 months later and stay for

10 days, then this is an assumption, when connection to an equipment that is assumed to be bought and installed within 2 weeks, this is an assumption.

These assumptions are to be written down with their details, one important side of its details is the confidence value of each assumption; the arrival of the expert is an assumption but the confidence is little bit low, he/she has been notified to come but no ticket is bought, and no reservation is made yet, installing the equipment is of high confidence since it has passed the bidding phase and the price has been paid the only left step is installation and so on, Also Assumption are filled in a special table:

Table 4.10: Project Assumptions

No	Name	Description	Confidence	Remarks
1	Expert arrival	…	Low	
2	Machine installation		high	

Sub practice 3, Identify Constraints:

Sometimes handling an issue is connected to some kinds of constraints, time frames, expense limits, and so on, also these constraints should be documented in a table:

Table 4.11: Project Constraints

No	Name	Description	Remarks
1	Fiscal year	Financial module to be delivered before 01/10	
2	…		

Sub practice4, Task dependences:

It is very important to define task dependencies for many reasons; one important reason is to build the execution plan of the tasks, it doesn't make sense to start with a task that depends on other unfinished tasks and postponing the others, this should

make a big waste of time and mess in plan. So define the dependency of tasks on each other before building the development plan. Adding level column to the tabular WBS will be useful in this phase:

Table 4.12: WBS with level column added

Task	Name	Sub task	Name	Is Ext Acquired	Where	To Be Reused	Where	Estimated Size	Infra Struct. Time	Total Time	Level
1	Handle	1.1	Prepare. …								
		1.2	Fill ….								
2	Manage	2.1	…								

Start by listing the tasks that depends on no other tasks in order to be handled, the tasks that are depending on these first level tasks and so on.

Or starts by the end tasks that represent the result of the whole project then the tasks on which these tasks depend and so on.

Hint: search for the four types of dependency (start to start, start to finish, finish to start, finish to finish).

Sub practice 5, Define the budget and schedule:

From the already filled WBS tabular form and using the level column used for dependency and precedence create a schedule sheet as follows (Budget is not considered):

Table 4.13: Project tasks schedule

No	From	To	Tasks	Remarks
1				
2				

Sub practice 6, Establish corrective actions criteria:

Things may come up during the project implementation and this usually happens, some of these things may affect the overall progress of the project, corrective actions may include preplanning or rescheduling or assigning different tasks to different phases or cycles this should not be left unhandled it should be recorded and documented in a corrective action sheet:

Table 4.14: Project corrective actions

No	Case	Description	Corrective action
1	…	…	
2	…		

4.2.3.6 STEP SIX: SP 2.2 – IDENTIFY PROJECT RISKS

One important part of project planning is risk assessment and management. a risk is the possibility that something bad (in terms of planning) will happen, If Risks are not managed properly they may cause severe impact to the whole project plan, risk management wide and comprehensive, Other than identification and analysis of risks,risk management include mitigation and contingency planning that also require a lot of work. But it is worth it to perform all issues related this portion of planning, in this framework a mandatory basic part of risk management is to be considered and an optional extension of it is added for those that would like to pay more attention to risk management.

Sub practice 1, Identify Risks:

Risk handling starts with the identification of risks, the term handling is used because this framework is not going to reach the level of risk management. First

method in identification of risks is brainstorming, brainstorming has its own rules and techniques that are out of scope of this framework but to make it positive and helpful, stakeholders should be involved and important things should be considered extracting thoughts (e.g. resources, project importance, impact and so on).

Sub practice 2, Document the Risks:

As for each important step, documentation of risks is to be done,after the risks are identified the following list is to be filled:

Table 4.15: Project Risks

No	Name	Description	Remarks
1	Server performance		
2	Electricity Problem		

Sub practice 3, Review and obtain agreement with relevant stakeholders on the completeness and correctness of the documented risks.

As risks are documented in a clear list, the list is to be circulated among stakeholders for any modification or further remarks, after this follow-up the list should be approved by all stakeholders.

Sub practice 4, Revise as appropriate:Risks are not documented for archiving, the risks document should be revised from time to time, new risks may be added, a risk actually became a problem, A risk is not anymore threatening the plan and so on.

4.2.3.7 STEP SEVEN: SP 2.3 – PLAN FOR DATA MANAGEMENT (OPTIONAL)

Before development starts, some data is to be collected and organized as a reference for the project and as development goes on also data is added to the original collection and so on. Data management involves several activities including

collection, privacy, archiving and handling this data. Although Project Data Management is important in huge projects and its importance differs according to the project itself but in small scale projects and projects that are developed by small workgroups the importance of project data management decreases to a very low level. This is because scale of project determines the scale of data in it and size of workgroup determines the security and privacy aspects of handling this data. Suppose that in a specific project the data is simple and its amount is too small, then managing its storage and retrieval and building its privacy and security issues and handling its collection and distribution will be a waste of time when regarding its benefits (according to cost benefit method). But in case a project involves some complex and important data, or the data it is too sensitive to be exposed to all team members working in the project, managing data is included in the framework as an optional part including the specification of some reference rules.

Sub practice 1 Establish requirements and procedures to ensure privacy and security of the data.

Specify portions of data in the project and who has the privilege to access these portions, a member in the workgroup may have access to a specific portion while other members may not. Use the Data management Plan table for specifying data portion and use User Column to enter each user that can access that data portion.

Sub practice 2 Establish a mechanism to archive data and to access archived data.

Even for members that are allowed to access the data sometimes they are allowed to read it but not print it, some other times allowed to print but not to take out

of building and so on. These rules are to be listed and documented. Use the column description for writing down how each user is allowed to use or access each data portion.

Sub practice 3 Determine the project data to be identified, collected, and distributed.

For the project to continue it is important to specify what parts of data should be collected and what parts are to be distributed. Also this should be documented. A column titled "Collect Or dist." Is used to specify for each data portion whether it is to be collected or distributed, User and description columns are used for who is responsible for collecting or distributing that portion of data and how.

Table 4.16: Data Management Plan

No	Portion Of Data	Collect Or Dist.	User	Description	Remarks
1					
2					

4.2.3.8 STEP EIGHT: SP 2.4 – PLAN FOR PROJECT RESOURCES

As mentioned earlier a WBS should be created to identify the required work packages of the whole project, these work packages (tasks or activities) and in order to be carried out correctly and in time as planned require the availability of different resources. These resources may be equipment, staffing, training or any other requirement for the implementation of the project. Sometimes the lack of a resource may stop the progress of the project or at least a specific work package that will later makes the progress of the project stop. That why resources are to be inspected and managed well.

Sub practice 1 Determine process requirements.

The Process of developing the software and managing this process as a project may require some resources related to process management itself, a project management tool is to be bought or process training is required or any other resource related to the process itself. These process related requirements are to be specified in the resource management table, some parameters are to be specified for each resource in the table too, the category of the resource for process related resources is always "process", The start date and end date of the availability of the resource and the importance degree. Finally follow up flags for the resource if available or not and if finished with or not.

Sub practice 2 Determine staffing requirements.

While discussing a project plan a usual and frequent question is "Do we need to hire someone for this project or not?" and this may be an option depending on several factors, like more employees for less time, or employment instead of training and so on. Staffing requirements and employing someone also to be written down in the resource management plan table with the category "Staffing".

Sub practice 3 Determine facilities, equipment, and component requirements.

Finally the most clear and known category of resources, equipment or component resources, e.g. servers, laptops, software packages (not related to process), books, training and so on. All of these are documented and categorized in the resource management plan table.

Table 4.17: Resource Management Plan

No	Description	Category	Description	Start Date	End Date	Available	Remarks
1							
2							

4.2.3.9 STEP NINE: SP 2.5 – PLAN FOR NEEDED KNOWLEDGE AND SKILLS

One of the basics for performing a task is to know how to perform it if someone doesn't have the knowledge and skill to do something then it is not possible to do it even if all resources are available. Planning knowledge and skills is handled in this section.

Sub practice 1 Identify the knowledge and skills needed to perform the project.

Only a naïve person or team can start working in a project or take the responsibility to work on it before defining the needed skills to handle every part of it, It is very necessary to collect the required skills for the whole project life cycle, referring to the WBS may be a lot helpful in organizing thoughts regarding identification of required skills, All the skills are to be listed in the required skills table.

Sub practice 2 Assess the knowledge and skills available.

After identification of all required skills, the next step is to identify which one of the skills is available within the team, that what the column "available" is used for in the same table

Sub practice 3 Select mechanisms for providing needed knowledge and skills.

Those skills that are not available should be provided sooner or later but the mechanism should be specified, in-house training, external training, staffing, buying a book and so on.

Sub practice 4 incorporate selected mechanisms into the project plan.

Mechanisms to make the skills available within the project are to be considered part of the project plan itself and merged within its tasks and activities.

Table 4.18: Required Skills

No	Skill Description	Available	Mechanism	Remarks
1				
2				

4.2.3.10 STEP TEN: SP 2.6 – PLAN STAKEHOLDERS INVOLVEMENT

Stakeholders differ in their involvement in the tasks of the project while one of them is involved in a task as a reference another one is for approval and a third is not involved at all in this task. The following sheet is a matrix to show which of the stakeholders is involved in what task:

Table 4.19: Stakeholder's involvement

Stakeholder / **Task**	S.H 1	S.H 2	S.H 3
1.1	1	1	0
1.2	1	0	0
1.3	0	0	1
2.1	1	1	0

Optional:

The Previous stakeholders task involvement matrix shows weather involvement exist or not using (0/1) flag but it doesn't show what is the kind of involvement, as an

option it is possible to add a lookup sheet for kinds of involvement and considering their values in the matrix (e.g. 1 for approval, 2 for reference, 3 for both approval and reference, and so on) then the values are used in the matrix instead of 0/1 values.

4.2.3.11 STEP ELEVEN: SP 2.7 – ESTABLISH THE PROJECT PLAN

Compile all outputs of previous sections into one document or sheet, every entry in each section is to be considered in terms of dependency and schedule and then entered in this final sheet of schedule (Calendar time schedule) and called a project plan as follows:

Table 4.20: Project Plan (Schedule)

No	From	To	Activity type	Activity	Remarks
1	01/10/2001	10/01/2001	Task	1.1	
2	11/01/2001	15/01/2001	Task	1.2	
3	16/01/2001	17/01/2001	Milestone	M 1	
4	.	.	Task	2.1	
5	.	.	Task	2.2	
6	.	.	Training	T 1	
7			Task	T3.1	

4.2.3.12 STEP TWELVE: SG3 – OBTAIN COMMITMENT TO THE PLAN

Commitment is a necessary issue for the project plan; the plan means nothing without the stakeholders approve their commitment to what is in the plan related to them of course. In large projects, there are several consequences and branching in the issue of stakeholder involvement, for the scope of this framework all sub practices and related issues are to be summarized in a commitment or approval letter from stakeholders, and email is enough.

4.3 PROJECT MONITORING AND CONTROL - PMC

4.3.1 PURPOSE

"The purpose of Project Monitoring and Control (PMC) is to provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan"

4.3.2 INTRODUCTION

Back to the rule of "Plan your work then work your plan", planning the work is already done in the section of project planning (PP) now as working the plan starts there should be a step by step review of this work in other words there should be monitoring and control of the project progress and this is what this section is about. Is the project going on fine? Is it going for away from its schedule? Are risks passed over? And so on.

4.3.3 KPA TO FRAMEWORK STEPS.

4.3.3.1 STEP ONE: SP 1.1, MONITOR PROJECT PLANNING PARAMETERS

The final project plan contains entries that are assumed to be met during the project progress in this section these planned attributes are to be monitored and compared to actual values that may or may not meet the planned values.

Sub practice 1 (progress against schedule): for each entry in the final project plan an additional attribute is added and recorded for the sake of PMC, that is a flag showing that an activity is finished or not, also another entry is to be added and filled but discussed in later sections is the deviation.

Table 4.21: Monitoring Project Plan (Schedule)

No	From	To	Activity type	Activity	Remarks	Finished?	Deviation
1	01/10/2001	10/01/2001	Task	1.1			
2	11/01/2001	15/01/2001	Task	1.2			
3	16/01/2001	17/01/2001	Milestone	M 1			

4	.	.	Task	2.1			
5	.	.	Task	2.2			
6	.	.	Training	T 1			
7			Task	T3.1			

Sub practice 2 (Project cost): as mentioned earlier costs and expenses are out of scope of this research.

Sub practice 3 (Attributes of work products and tasks): monitoring project plan involves reading schedules and marking whether or not the schedules are met. Work packages in the WBS should also be monitored with a more detailed entry which is the amount of deviation form scheduled time, deviation should be recorded along each entry in the WBS tabular form and also filled in the final project plan table, it is entered (when activity is finished) in terms of time. Deviation may be a positive value, negative value or a zero.

Table 4.22: Monitoring WBS Tabular Form

Task	Name	Sub task	Name	Is Ext Acquired	Where	To Be Reused	Where	Estimated Size	Infra Struct. Time	Total Time	Deviat-ion
1	Handle	1.1	Prepare. …								
		1.2	Fill ….								
2	Manage	2.1	…								

Sub practice 4 (Resources): Also resources should be monitored especially that resources may be planned to have not already exist, additional attributes are to be added for PMC recordings, (made available or not?), (finished usage or not?) and also deviation.

Table 4.23: Monitoring Resource management plan

No	Description	Category	Description	Start Date	End Date	Available?	Made available?	Finished	Remarks
1									
2									

Sub practice 5 (knowledge and skills): As resources the same additional attributes are to be added for recording PMC reading in required skills table.

Table 4.24: Monitoring Required Skills

No	Skill Description	Available	mechanism	Made available	Remarks
1					
2					

Sub practice 6 (documenting deviations): As mentioned, for each entry in the planned schedules deviation value should be recorded during progress monitoring of the project.

4.3.3.2 STEP TWO: SP 1.2, MONITOR COMMITMENTS.

As stakeholders approve the plan and show commitment to what it contains, each of them should also provide what he/she is committed to provide, Remarks regarding commitments should be recorded in a stakeholder commitment table that includes positive remarks regarding commitment as well as negative ones.

Table 4.25: Stakeholder's commitment remarks sheet

No	Stakeholder	Commitment	Comments
1			
2			

4.3.3.3 STEP THREE: SP 1.3, MONITOR RISKS.

The importance of Risk Management in the project includes not only the documentation of risks in the sheet discussed earlier but also the follow up of these risks while the development process is in progress that's why three additional columns are added to the project risks sheet to help in monitoring and control, the first column is (Is problem?) which means when indicated as yes that the risk actually occurred and what was considered as a threat only is converted to a problem that needs to be solved. The second attribute or column is (Corrective action taken), if in any case a corrective action is taken in order to deal with one of the risks this should be registered here and written under this title, the final column is (Is passed?) is used to

indicate that a risk is not a risk any more either it became a problem and handled or the schedule passed and the risk didn't occur, the result in both cases is that the risk is not a threat any more.

Table 4.26: Monitoring Risks

No	Name	Description	Remarks	Is Problem?	Corrective Action	Is passed?
1	Server performance					
2	Electricity Problem					

4.3.3.4 STEP FOUR: SP 1.4, MONITOR DATA MANAGEMENT (OPTIONAL).

Considering Data Management an optional part of the framework means that monitoring this part is also optional, in case data management is to be monitored a (progress remarks) column is to be added to the data management plan sheet.

Table 4.27: Monitoring Data management plan

No	Portion Of Data	Collect Or Dist.	User	Description	Remarks	Progress Remarks
1						
2						

4.3.3.5 STEP FIVE: SP 1.5, MONITOR STAKEHOLDERS INVOLVEMENT.

A sheet was previously created in order to indicate the involvement of stakeholders in the project and on which task or activity, this sheet is to be used for monitoring in a different way, for each cell indicating that a specific stakeholder is involved in a specific task in any way instead of indicating involvement, involvement monitoring comments are to be written down.

Table 4.28: Stakeholders involvement progress comments

Stakeholder / **Task**	S.H 1	S.H 2	S.H 3
1.1			
1.2			
1.3			
2.1			

4.3.3.6 STEP SIX: SP 1.6, CONDUCT PROGRESS REVIEWS.

Referring to the plan as the project keeps in progress is the base of project monitoring, that's why a progress remarks column is added to the plan and updated from time to time during progress.

Table 4.29: conduct project plan (schedule)

No	From	To	Activity type	Activity	Remarks	Finished?	Deviation	Progress Remarks
1	01/10/2001	10/01/2001	Task	1.1				
2	11/01/2001	15/01/2001	Task	1.2				
3	16/01/2001	17/01/2001	Milestone	M 1				
4	.	.	Task	2.1				
5	.	.	Task	2.2				
6	.	.	Training	T 1				
7			Task	T3.1				

4.3.3.7 STEP SEVEN: SP 1.7, CONDUCT MILESTONE REVIEWS.

While tracking progress reviews as indicating the previous section, when a mile stone is encountered a report containing activities and PMC added columns is prepared and may be provided to stakeholders according to description of milestones.

4.3.3.8 STEP EIGHT: SG2 MANAGE CORRECTIVE ACTIONS TO CLOSURE.

SP 2.1: ANALYZE ISSUES:

Sub practice 1 and 2 (gather issues, analyze issues): all deviation values are to be collected and analyzed so that the project doesn't deviate from its schedule in remarkable values.

SP 2.2: TAKE CORRECTIVE ACTION:

Sub practice 1 and 2 and 3 (Determine, Review, Negotiate): After deviations are collected and analyzed which corrective action regarding these deviations is to be taken and manipulated, these actions should be reviewed with stakeholders. Negotiated with external factors if needed although this framework is designed for cases where no huge interconnection between projects is involved.

SP 2.3: MANAGE CORRECTIVE ACTION:

Sub practice 1 and 2 and 3 (Monitor, Analyze, Document): While implementing corrective actions, their implementation should be monitored and their results are analyzed and documented.

Table 4.30: Monitoring Project corrective actions

No	Case	Description	Corrective action	Result	PMC Remarks
1	…	…			
2	…				

4.4 MEASUREMENT AND ANALYSIS - MA

4.4.1 PURPOSE

"The purpose of Measurement and Analysis (MA) is to develop and sustain a measurement capability that is used to support management information needs."

4.4.2 INTRODUCTION

In terms of management, for every project there should be a clear and direct statement of the objectives and aims of that project, a software project is the same, and after that using a formal and disciplined framework is also the same, several objectives may be involved in such a process, in order to know whether or not the process meets its objectives, a strategy for Measurement and analysis should be Constructed. This section (Measurement and analysis) focuses on how to build and construct this strategy.

There are two Specific Goals in the measurement and analysis Process area, Align measurement and analysis activities and provide measurement results, practices in the first goal are considered as a preparation or planning for the practices to be executed in the second goal.

4.4.3 KPA TO FRAMEWORK STEPS.

4.4.3.1 STEP ONE: SP 1.1, ESTABLISH MEASUREMENT OBJECTIVES.

This specific practice is concentrates on why is it important for the development team to carry out the issue of measurement and analysis, it is like asking (why are you measuring this issue? For security, for improvement, for commitment …etc), this specific practice consists of five sub practices that are all related to specifying these objectives. But still there is a need in this framework for simplicity as much as possible without losing control, and since every measurement to be handled is to be

connected to an objective (if objectives are built), then a column to specify objective will be added to the table of measurements that will be discussed in next sections.

4.4.3.2 STEP TWO: SP 1.2, SPECIFY MEASURES.

The core of measurement and analysis is specifying the measures to be followed up, i.e. what to measure? These measures should all be listed in a table, the sub practices in this specific practice include, documenting each measure, describing it, operational definition (base or derives from other bas measures like equations), the following table is the starting table of the measurement and analysis items table (notice that an objective column is added to meet SP 1.1):

Table 4.31: Measurement Items Table

M_ID	Name	Description	Base/ Derived	Related objective
1	Quality	Quality before implementation	D	
2	Accuracy	Error in estimation	D	

4.4.3.3 STEP THREE: SP 1.3, SPECIFY DATA COLLECTION AND STORAGE PROCEDURES.

In simple words and without detailed explanation of sub practices this specific practice involves the data part in the process of measurement and analysis, what data to collect, how to collect it and how many times or repeat at specific stages or just once and so on, also where this data is to be stored, for this framework the storage is only within the framework itself in the specified tables and sheets:

Table 4.32: Measurement Items Table including data collection and storage

M_ID	Name	Description	Base/ Derived	Related objective	Data to collect	How to collect	Repeat
1	Quality	Quality before implementation	D		(ok) in quality check list	count	once
2	Accuracy	Error in estimation	D		Log table result calculation	Refer to log	once

4.4.3.4 STEP FOUR: SP 1.4, SPECIFY ANALYSIS PROCEDURE.

After data is collected and documented in the sheets it is important to know how to analyze this data, how it should be manipulated in order to get from it the required information, a new column is added to the measurements shoot especially for this purpose:

Table 4.33: Measurement Items Table including data collection and storage and analysis

M_ID	Name	Description	Base/ Derived	Related objective	Data to collect	How to collect	Repeat	How to analyze
1	Quality	Quality before implementation	D		(ok) in quality check list	count	once	Number / total number
2	Accuracy	Error in estimation	D		Log table result calculation	Refer to log	once	Refer To Log

4.4.3.5 STEP FIVE: SG2, PROVIDE MEASUREMENT DATA.

SP 1.1 – SP 1.4: Collect and analyze data, store and communicate results:

Specific practices and sub practices related to this specific goal are all concerned with the actual implementation of the steps specified in the previous specific Goal.Here it is required to collect data as mentioned stored in its column and manipulate analysis as clarified and store analysis results in its specified column and finally provide this sheet of results to any involved stakeholders or managers:

Table 4.34: Measurement Items Table with actual result

M_ID	Name	Description	Base/ Derived	Related objective	Data to collect	How to collect	Repeat	How to analyze	Readings / Results	Analysis Result
1	Quality	Quality before implementation	D		(ok) in quality check list	count	once	Number / total of numbers	28 Ok 40 Total	28/40 = 70%, 68% for last project, 62% one before
2	Accuracy	Error in estimation	D		Log table result calculation	Refer to log	once	Refer To Log	1.75%	1.78% for last project, 2.1% for one before

4.5 PROCESS AND PRODUCT QUALITY ASSURANCE - PPQA

4.5.1 PURPOSE

"The purpose of Process and Product Quality Assurance (PPQA) is to provide staff and management with objective insight into processes and associated work products."

4.5.2 INTRODUCTION

Software process as defined earlier is a "set of activities to produce a software product", every task or activity or even a sub task or sub activity within the software process that is known as sub practice produces what is called work product, work product may be a summary report, a detailed sheet, an approved document or even a specific value.

While the project is in progress PPQA process area (Process and Product Quality Assurance) concentrates on how much the progress is compliant with the defined process descriptions and standards e.g. templates.

4.5.3 KPA TO FRAMEWORK STEPS.

4.5.3.1 STEP ONE: PROCESS AND WORK PRODUCT.

PPQA splits its concentration into two different areas, Process and work product, in each of these areas it is required to specify four things, what to evaluate? How to evaluate? When to evaluate?and the actual evaluation. Deep details causes the differentiation between the two areas, in this framework and since the scale or size of work is small no need to go into deep details for each of the two areas, that's why there will be no need to split process from work product and both of them are used together within the framework.

4.5.3.2 STEP TWO: WHAT TO EVALUATE.

What to evaluate is the first question in PPQA section, Prepare a list of products that needs to be examined for quality (e.g. main package, all work packages – sub tasks …etc):

Table 4.35: Products to be evaluated

Product NO	Product description	Remarks
1	Main package	
2	All Subtasks	Refer to Product quality check list
3	Reports	General template checkout

4.5.3.3 STEP THREE: HOW TO EVALUATE.

For each of the Listed Products, write down how this product is to be evaluated, the criteria (parameter) and how to make the evaluation:

Table 4.36: Evaluation criteria

Product NO	Evaluation criteria No	Criteria	Criteria Description	How To Check
1	1	Accuracy	Check for accuracy of result	Different inputs, compare with manual calculation
1	2	Performance	Check if it does what it should do fast enough	Try complex cases and observe with user.
2	1	SRS	Meets the requirement specification in the requirement sheet	Check input and output and compare to Specs
2	2	Interface	Check If interface is consistent with others and accepted by user.	take continuous interface feedback from user
2	3	Integration	Check integration with other related parts	Manipulate data and check other modules
2	4	User	User Acceptance testing	User makes the testing that the programmer made

4.5.3.4 STEP FOUR: WHEN TO EVALUATE.

After the previous 2 sheets are filled then it is clear what and how to evaluate; now it is important to write down when to make the evaluations (e.g. at the prototype level, exactly before implementation … etc):

Table 4.37: Evaluation milestones or stages

Evaluation Stage No	Description	Remarks
1	Prototype	interface And business logic related issues
2	Work package finish	Tested by programmer before going on to work package.
3	Work package closure	Before marking the work package as done.
4	Final	At the final stage directly before implementation

4.5.3.5 STEP FIVE: ACTUAL EVALUATION.

For each of the Listed Products and their evaluation criteria, write down the result of evaluation in each of the stages or milestones that were listed in the previous section:

Table 4.38: Product Quality check list

		Milestones / Stages For quality evaluation										Remarks
Product NO	Evaluation criteria No	1	2	3	4	.	.	.	.			
1	1	ok	ok	ok								
1	2	ok	ok	ok								
2	1	ok	ok	201								
2	2	ok	202	ok								
2	3											
2	4											

Result No	Description
201	The work package failed to give out the expected results as indicated by requirement specification
202	The user noticed a difference in navigation method between the screen and the other screens

4.6 CONFIGURATION MANAGEMENT - CM

4.6.1 PURPOSE

"The purpose of Configuration Management (CM) is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits. "

4.6.2 INTRODUCTION

When managing a software development project, it is necessary to follow up and maintain the history of the projects the different versions of it and also the bugs and fixes of these bugs that appeared in the system, this is where Configuration Management is needed. Configuration Management as a concept includes several areas of work or points of focus like version control systems and change or bug tracking systems.

The problem with configuration management when analyzed within the scope of this research is that it includes several tasks and activities that are considered large scale and out of the frame of small scale systems or small software development workgroups, for example, it includes the establishment of a mechanism to manage multiple control levels like lower level CCB (Change Control Board) and higher level CCB and so on, and also the involvement of configuration management systems and categorizing them into dynamic systems, controlled systems and static systems, and a lot of other examples and components of configuration management.

In order to deal correctly with this problem the purpose and general idea of the sub practices is considered and organized in a simpler form presentation of these ideas, also if in some cases the system developed was very small then configuration management may be skipped, that's why it is considered an optional section of the

framework but still it is recommended, the general idea of the sub practices includes the specification of items to be followed up, the changes requested to be done, the changes actually done to follow up items and finally the versions that are produced based on these changes and modifications.

4.6.3 KPA TO FRAMEWORK STEPS.

4.6.3.1 STEP ONE: BASELINE ITEMS.

Specification of items to be followed up: Several components may be involved in change management and version control, packages, screens, reports, plans, security issues and so on, the starting point in following up these components is putting them all in a table in order to build later steps on, the items in this table are called baseline items, if the number of baseline items is large it is good to write down the category of the item it may be helpful in giving indications for later projects, categories may be (Plans, Requirements, Packages …etc) it also may be (Code, functionality, interface … etc) :

Table 4.39: Baseline Items

Baseline Item No	Description	Category (optional)
HR-001	Manage employment Evaluation scale	Requirement
HR-002	Handle Employment Form	Requirement
Calculate-Salaries	A package including all methods for salary related calculations	Package

4.6.3.2 STEP TWO: CHANGE REQUESTS.

Changes requested to be done: Any change in the system is based on a change request; change requests are to be written in the following table, each change request must have a number. Who requested this change is also written down and the most

important for this stage is the change description that describes in details what is the required change, then an analysis is done to the requested change (what does it affect, which baseline items are affected, who should approve the request, who should handle it and follow it up, and so on) then the status of the request is updated step by step it may be (new, approved, in progress, closed … etc):

Table 4.40: Change requests

ChReqNo	Requested by	Change description	Change analysis	Status
001	Hr- Assistant	Add deductions possibility	Adding a place in salary form for deduction value	Done
002	Manager	Approval Mechanism	Any financial transaction - approved my management	Pending

4.6.3.3 STEP THREE: CHANGE FOLLOW-UP.

Changes actually done to follow up items: After a change request is analyzed and approved to be done and assigned as a task to one (or more) of the team the change is to be handled and manipulated. Any change in the system is based on a change request, and may be connected to a baseline item, one change may be related to a change request, but also in other cases several changes may be connected to one request according to effect of the change requested:

Table 4.41: Change follow-up

ChangeNo	Date	ChReqNo	Baseline Item No	Change description	Remarks
10/105					
10/106					
10/107	04/10/2011	025	HR-012	Include Deduction possibility	
10/108	04/10/2011	003	Calculate salaries	Take into consideration deduction value	

4.6.3.4 STEP FOUR: VERSION CONTROL.

Versions: Any change made to any component of the system (baseline item) transforms that item from a state to a newer state (version), each version after the first one has an old or previous and a new state, e.g. a specific form or screen (registered as a baseline item) shows the list of employee vacations but it doesn't show the remaining balance of the employee vacations, a request is issued by the HR contact person that the balance should appear on the screen, a change is done to the screen by one of the programmers, the screen is now working fine, in version control a number is assigned to the version of the screen and in the previous section written (without vacations balance), in the new portion written (with balance added):

Table 4.42: Version Control

VerCtrl No	Date	Baseline item	Version	Previous	New	User	Change no	Remarks
126								
127								
128	04/11/2011	HR-012	4	No deduction option	With deduction option added	Mem.1	10/107	
129	05/11/2011	HR-012	5	Printing not working	Calling report is fixed	Mem.1		
130	05/11/2011	Calc-sal	24	Deduction not considered	Deduction is considered	Mem.3	10/108	

4.7 SUPPLIER AGREEMENT - SA

4.7.1 PURPOSE

"The purpose of Supplier Agreement Management (SAM) is to manage the acquisition of products from suppliers"

4.7.2 USAGE AND CONSIDERATION

The purpose of this framework is to build a framework to be used for software development in small software workgroups like IT units or departments in most institutes, this framework is assumed to be useful and helpful for these work groups in all tasks and activities through the software process but in terms of software engineering topics, All of this is done in an internal basis in terms of the software development workgroup itself, Any External or financial issues are not considered within the scope of this research.

Supplier agreements, their contents, their maintenance and follow up, all of this is part of the external issues that goes beyond the boundaries of the framework. In case it is needed to manage supplier agreements for any kind of reasons, or for a specific case or a specific workgroup, this could be done separately and the framework contains nothing that contradicts with the usage of any methodology in supplier agreement management.

CHAPTER FIVE

EVALUATION AND ANALYSIS

5.1 INTRODUCTION

After going through each section and processingeach area by investigation and analysis and ending up with our framework, there should be an evaluation process of the framework in order to know how much it is close to achieve its goals. Evaluating the framework needed the participation of several qualified professionals that work in the field of software engineering and in small software development groups. It is also important to get the feedback of those that used to work on software process improvement methodologies. In order to do this and besides the framework and its filled example there was also an evaluation sheet to be filled by participants and then analyze the collected feedbacks.

5.2 PARTICIPANTS

5.2.1 CHOOSING PARTICIPANTS

Choosing participants and preparing the participants list is not an easy task, several obstacles came up and had to be managed and handled, e.g.:

1. It is not appropriate to distribute the framework and its evaluation sheet on friends and colleagues and all those that are in the email contact list, it is not a fully quantitative task, a feedback from a qualified professional may be better than tens of feedbacks from other people not working in the field or to be

more accurate, their feedbacks mean nothing, that's why this task is selective besides being quantitative.

2. Also variation of field of work is important of course not the participants field of work but the institute or company for which he/she works. That's why the targeted list of participants included mainly software development companies and NGOs, Medical institutes, training, insurance companies. And others.

3. Convincing people to cooperate is also a significant task! there should be an insisting follow up and several reminders and phone calls in order to motivate each one of the participants to follow up what has been delivered to him/her.

5.2.2 PARTICIPANT TYPES

Those that participated in the evaluation are sometimes professional individuals; years of experience especially in the filed makes an individual an excellent reference in topics related to his field, on the other hand several other participants and although they were registered as individuals but they were not, when a group is contacted through someone, then the evaluation of this person is the evaluation of a group, and the importance of his evaluation increases according to the number of those participated with him and their experience and capability. Programming teams were contacted, programming companies, it units in institutes, insurance companies, medical institutes and others. A significant participation was from an auditing company that provided besides their evaluation some recommendations for future follow up.

5.3 EVALUATION QUESTIONNAIRE:

When constructing the questionnaire, the main objectives of the framework should be put on the desk and questionnaire items should be mapped to them, simplicity, terminology, discipline and applicability are the four objectives of the framework, the main objective of the simplification of the SPI is to achieve simplicity, but not to lose any of the other three characteristics (terminology, disciple and applicability).

The questionnaire consists of two sections one of them is related to the framework itself and its usage and the second is related to comparison of framework with SPI. it includes 15 questions, excluding two questions one is regarding the linguistic skill of English (since the framework is created in English but to be used by non-English speaking users and the questionnaire was not written in English). The second question to be excluded is regarding the knowledge and familiarity of other SPI methodologies, and this question is considered while evaluating the feedbacks of the second section of the questionnaire.

The participants received three documents, the framework itself, a filled example of the framework, and the questionnaire, (note: the following questionnaire is an Englishtranslation of the one actually used; which as we mentioned earlier was in Arabic):

Evaluation Form

Please write (×) where appropriate:

Strongly Disagree	Disagree	N/A	Agree	Strongly Agree	Evaluation Item	No.
					Terms used are Clear.	1
					Sequence of steps is reasonable and matches the actual work steps.	2
					Suggested Framework is applicable	3
					Framework is easy to apply and use.	4
					Using the framework does not form an obstacle in the actual work steps	5
					Suggested framework helps in improving the software development process.	6
					Suggested framework does not need a lot of training.	7
					Considering Optional and not optional sections of the framework is reasonable.	8
					You prefer to use the framework in your work.	9
					Your English language is between very good and excellent	10
					Worked in or have knowledge or familiar with any known SPI methodology (CMMI, ITIL, …etc).	11
					Terminology used is close to that terminology that you know in the methodology you are familiar with.	12
					Sequence of steps used is close to that sequence that you know in the methodology you are familiar with.	13
					Using Suggested framework helps in going towards using a well known SPI methodology	14
					Things that exist in SPI methodology and not in the framework are not needed for small scale workgroup	15

Any comments or suggestions:

_____

_____

Figure 5.1 The evaluation Form

5.4 EVALUATION

5.4.1 Real Case Testing:

As mentioned earlier, the framework is supposed to be tested on at least one real case (i.e. actual project development), this will provide more accurate evaluation and gives a better support to the results of the feedback and the questionnaire handled. Testing in real case means a step-by-step usage of the framework in an actual development environment to produce software that is going to be actually used and implemented.

A team has been followed up in order to handle the real case testing while developing software, the team consists of three developers, two seniors and one junior; the team has a task of developing a human resources module for one of the customers.

After testing the framework and reaching the final stage of development, a meeting has been conducted to discuss the feedback regarding the framework and discussing the four major points in evaluating the framework (Simplicity, Applicability, terminology and discipline), the following was the result of their feedback and testing:

1. **Simplicity**: As expected, starting the usage of the framework for the first time was not so simple, but once the process was started it was not hard to go forward in implementation step by step, steps were clear and direct in the required data to fill, the sections of the framework vary in simplicity level but in general it was simple enough to be used, the section related to requirement documentation needs more concentration than other sections

to use, Building the breakdown structure hierarchy also needs deep thinking but once created the following steps are simple.

2. **Applicability**: using the framework is not difficult as discussed in the simplicity part of the evaluation; the team showed interest and ability to use the framework more than once, this interest that they showed is expected to increase if the framework is frequently used. But as a recommendation, computerizing the framework in a simple tool will help a lot in this part of evaluation.

3. **Terminology**: It was clear from those that are familiar with the CMMI that the terms used are very close to those used in CMMI, it was noticed that some of the terms were missing while comparing with CMMI but that was clearly justified by the fact of eliminating some portions of the CMMI for simplifications Reasons.

4. **Discipline**: the framework was well organized and not scattered, even if there is a part that is not needed it was skipped in a systematic way, assuming some parts as required and mandatory was also supporting the idea of working within a disciplined track and not left on a chaotic area.

5.4.2 FEEDBACK COLLECTED:

For each feedback collected the evaluation sheet (questionnaire) was evacuated in a total feedback sheet to get final grades for each of the evaluation items as well as for the objectives mentioned for the framework.

5.4.3 CONSIDERATIONS:

While preparing the totals sheet from the feedbacks collected, we followed a correction criterion in order to get accurate results as follows:

- For question number 11 (you have worked in or have knowledge or familiar with any known SPI methodology (CMMI, ITIL, etc).) if this question was not answered with (agree, or strongly agree), then all subsequent questions become irrelevant and were discarded (if filled).

- If it was clear that the questionnaire was filled without sufficient care or real interest, then the whole evaluation was discarded (for example, a sheet filled with “I agree” on all questions).

5.4.4 FEEDBACK EVALUATION SUMMARY:

Table 5.1: Feedback evaluation summary

No.	Strongly Agree	Agree	N/A	Disagree	Strongly Disagree
1	6	14	**0**	**0**	**0**
2	5	14	1	**0**	**0**
3	2	16	1	1	**0**
4	2	13	3	2	**0**
5	1	16	2	1	**0**
6	9	10	1	**0**	**0**
7	3	15	3	5	**0**
8	4	15	**0**	1	**0**
9	1	13	5	1	**0**
10	5	14	**0**	1	**0**
11	1	11	1	7	**0**
12	1	9	2	**0**	**0**
13	1	9	2	**0**	**0**
14	1	9	2	**0**	**0**
15	**0**	7	4	**0**	**0**

5.5 ANALYSIS

5.5.1 USAGE OF FEEDBACK:

Evaluation and analysis in researches need numbers and values to build the conclusions on, but the problem in software engineering issues and some other similar fields of specialization is that non numerical data is involved sometimes in their researches, Participants selections in the evaluation form are not numeric, averages and grades can't be derived from such selections, these non-numeric selections may give indications to something or even form a base for further studies but still they are not precise as numeric data and values.

5.5.2 MAPPING NON NUMERICAL FEEDBACK TO NUMERICAL VALUES:

As mentioned there is a problem with non-numeric readings and selections in the evaluation, hence these values must be somehow mapped to other numeric values to continue calculations, and derive precise values and numbers that can be analyzed. Because of this each one of the five selections that exists in the questionnaire is assigned a value as shown in the table below:

Table 5.2: Evaluation selections weights

Evaluation Selection	Weight
Strongly Agree	100
Agree	75
N/A	50
Disagree	25
Strongly Disagree	0

Based on the weights table above and in the total sheet (Table 5.1) that was filled from the questionnaire feedback received from each participant, totals and grades For each evaluation item (question in the questionnaire) are derived and calculated as follows:

- For each selection multiply number of participants that chose that evaluation selection by the weight assigned to it, repeat this for all selections and sum them up in the total column, for example notice evaluation item number 3:
    - = (2 X 100) + (16 X 75) + (1 X 50) + (1 X 25) + (0 X 0)
    - = 200 + 1200 + 50 + 25 + 0
    - = 1475
- Sum up number of participants in every evaluation item in the count column. For evaluation item number 3: (2 + 16 + 1 + 1 + 0 = 20)
- Grade = total / Count, For evaluation item number 3: (1475 / 20 = 73.75)

Repeat these steps for all questions of the questionnaire to get the following result table:

Table 5.3: Readings result sheet

Weight →	100	75	50	25	0			
No.	Strongly Agree	Agree	N/A	Disagree	Strongly Disagree	Total	Count	Grade
1	6	14	0	0	0	1650	20	82.5
2	5	14	1	0	0	1600	20	80.0
3	2	16	1	1	0	1475	20	73.8
4	2	13	3	2	0	1375	20	68.8
5	1	16	2	1	0	1425	20	71.3
6	9	10	1	0	0	1700	20	85.0
7	3	15	3	5	0	1700	26	65.4
8	4	15	0	1	0	1550	20	77.5
9	1	13	5	1	0	1350	20	67.5
10	5	14	0	1	0	1575	20	78.8
11	1	11	1	7	0	1150	20	57.5
12	1	9	2	0	0	875	12	72.9
13	1	9	2	0	0	875	12	72.9
14	1	9	2	0	0	875	12	72.9
15	0	7	4	0	0	725	11	65.9
Totals	36	160	26	11	0	∑	233	
Average	15.5	68.7	11.2	4.7	0.0	∑	100	
SPIRelated Totals	3	34	10	0	0	∑	47	
SPIRelated Average	6.4	72.3	21.3	0.0	0.0	∑	100	

Readings in the previous table show that throughout all the questions 36 answers were “strongly agree” while 160 answers for “Agree” and so on, the sum of all answers is 233 answers or selections. Back to the 5 choices in the evaluation sheet, percentages of selections can be calculated by dividing the count of selections of every choice by the total count which is 233:

- Strongly Agree: 36 / 233 = 15.5%
- Agree: 160 / 233 = 68.7%
- N/A: 26 /233 = 11.2 %
- Disagree: 11 / 233 = 4.7%
- Strongly Disagree: 0 / 233 = 0%

The following is a graphical representation of results:

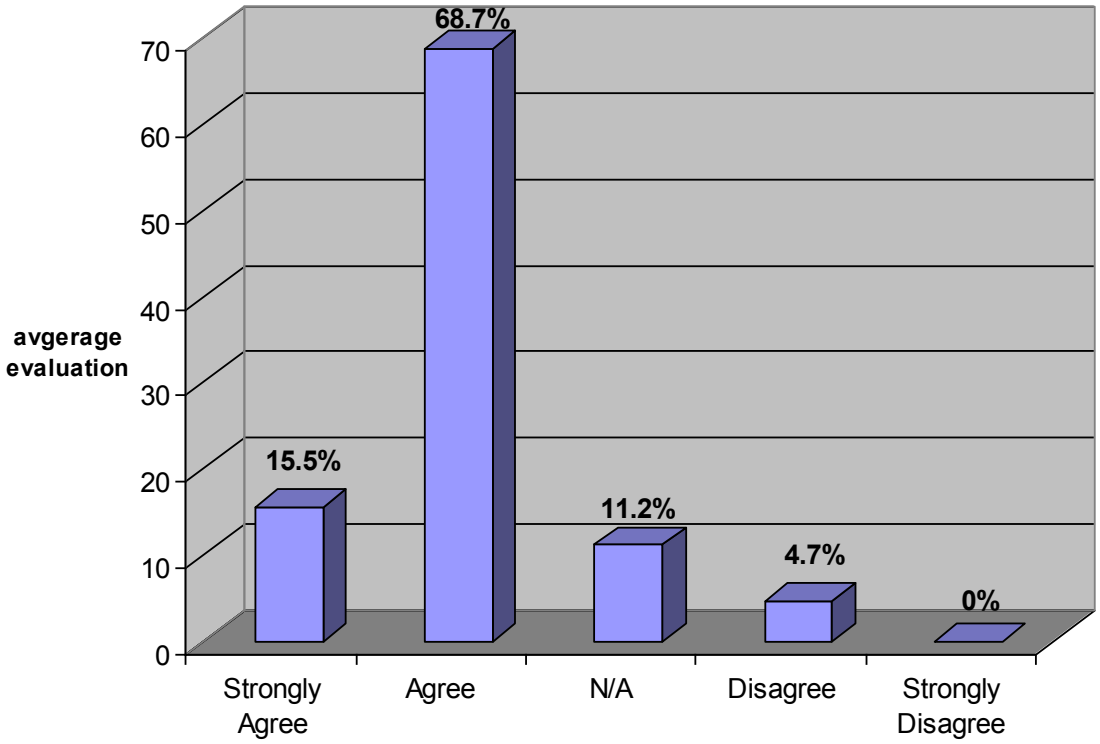


Figure 5.2: Selections of the evaluation form

5.6 DISCUSSION

Grades for each question are meaningful when investigating in details how good the framework is in that particular point of the question, but still there should be wider point of view regarding the readings and grades that are obtained so far, going back to the issue mentioned regarding writing questions in a way that serves the analysis of achieving different objectives, the questions can be categorized in several categories as in the following table:

Table 5.4: Analysis points

#	Analysis Points	Questions
1	Overall evaluation	All questions are involved except questions 10 and 11 that are related to the participant and not to the framework itself.
2	SPI Section Evaluation	Questions that are in the second section of the questionnaire and related to comparisons with SPI methodologies used.
3	Simplicity	Questions : 4,7
4	Discipline	Questions : 2,8,13,14,15
5	Terminology	Questions : 1,12
6	Applicability	Questions : 3,5,6,9

For each of the Analysis Points sum up the grades obtained for each of the questions that correspond to that analysis point and divide by their number (count questions) to get the grade for that analysis point, the following table is derived:

Table 5.5: Analysis points grades

#	Analysis Point	Total Of grades for each	Number of	Grade

		question	questions	
1	Overall evaluation	**956.3**	**13**	**73.6**
2	SPI Section Evaluation	**284.7**	**4**	**71.2**
3	Simplicity	**134.1**	**2**	**67.1**
4	Discipline	**369.2**	**5**	**73.8**
5	Terminology	**155.4**	**2**	**77.7**
6	Applicability	**297.5**	**4**	**74.4**

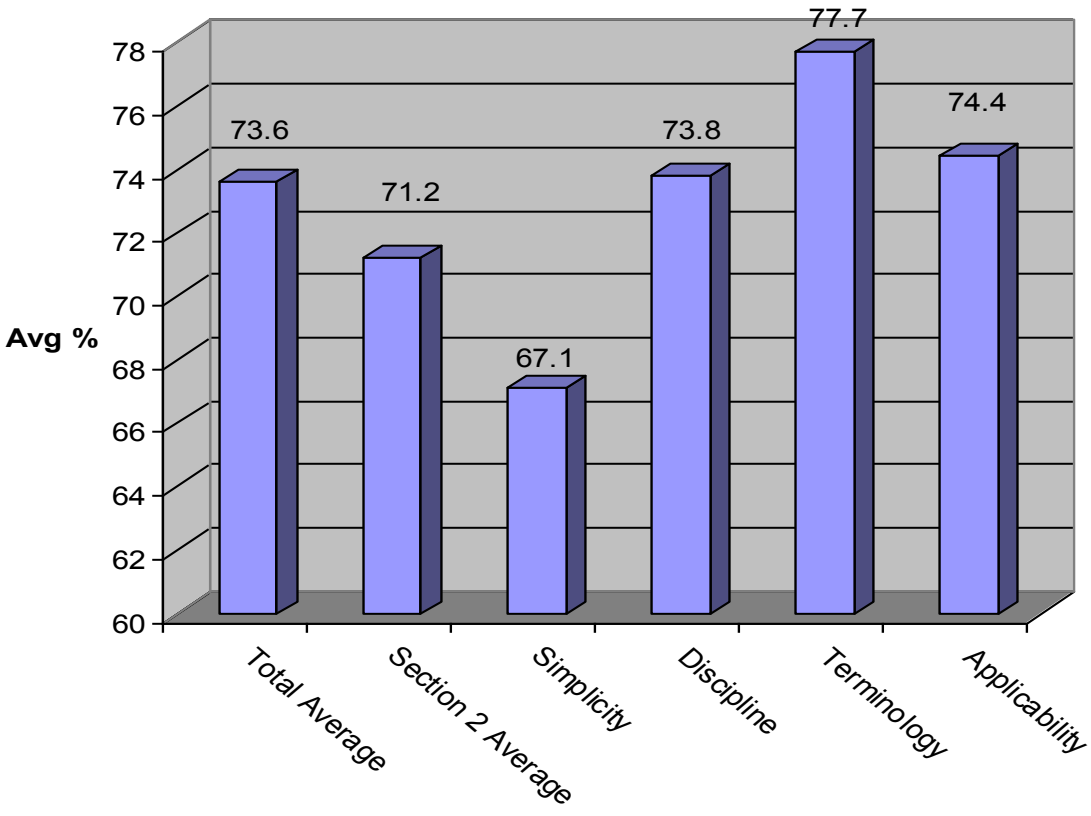


Figure 5.3: Grades or scores of analysis points

CHAPTER SIX

CONCLUSION

The main objective of this research is to build a simple and applicable framework; this framework organizes the work of software engineers within a disciplined set of practices with the maintenance of terminologies of the original CMMI model. The simplified framework focused on the following aspects (*Simplicity*, *Applicability*, *Terminology* and *Discipline*) in addition to the SPI relevancy in general. Referring to the results of the research as shown in section (5.6) it is clear that *Simplicity* was achieved by an evaluation percentage of 67.1%. Few participants were consulted regarding such a percentage, it was a description of such a percentage in this research, their feedback was in the area between the good and very good achievement, but it was clear (sometimes stated) that 67% is very good when considering a percentage of approximately 77.5% in maintaining the *Terminology*, keeping such a high percent should affect the simplicity level.

Discipline got 73.8% which is also considered a pretty good enhancement of the teamwork in the direction of formality and systematic attitudes in higher level methodologies. *Applicability* which is also an important measure got a better evaluation percentage of 74.4%, and this shows that the framework was not only a theoretical research but also a practical product that can be used in real work environment as the results show.

REFERENCES

[1] Ana Sofia C. Marcal1, Bruno Celso C. de Freitas, Felipe S. Furtado Soares ,Arnaldo D. Belchior, "Mapping CMMI Project Management Process Areas to SCRUM Practices", IEEE – 2007

[2] CarstenRusengJakobsen, Jeff Sutherland, "Scrum and CMMI – Going from Good to Great", IEEE – 2009

[3] Claude Laporte, Simon Alexandre, Alain Renault, "The application of international software engineering standards in very small enterprises ", IGI Global, 2008

[4] Claude Y. Laporte, Jean-Marc Desharnais, Mohamed MounirAbdouelfattah,Jean-Claude Bamba, Alain Renault,Naji Habra, "Initiating Software Process Improvement in Small Enterprises: Experiments with CETIC's Micro-Evaluation Framework ", International conference on software development – university of Iceland - 2005

[5] Daniel J. Rocha, "Strengthening the validity of software process improvement measurements through statistical analysis ", University of Gothenburg University (GUPEA), 2008

[6] Daniel Karlstrom, "Increasing involvement in software process improvement", Lund University – Master thesis 2002

[7] E. Georgiadou, "Software Process and Product Improvement: A Historical Perspective", Cybernetics and Systems Analysis; Jan/Feb- 2003.

[8] Eugene G Mcguire ,"Initial Effects of software process improvement on an experienced software development team" , IEEE, 1996

[9] FatihYucalar, SenolZaferErdogan, "A questionnaire based method for cmmi level 2 maturity assessment", journal of aeronautics and space technologies - 2009.

[10] Francisco Alvarez R, Jaime Munoz A, Pedro Cardona and Alfredo Witzenfeld R., "Interpretation of CMMi for the small software industry", HifenUruguaiana, 2006

[11] Francisco Pino, Felix Garcia, Mario Piattini, " Software process improvement in small and medium software enterprises: a systematic review ", Springer Science and business media, 2007

[12] Frank Lui, Yan Sun, Gautam Kane, Yuji Kyoya, KunioNogudhi, "QFD Application in software process management and improvement Based on CMM " ACM, 2005

[13] G.K.Viju, Mohammed MerghanyAbdElsalam, Khalid Ahmed Ibrahim, Mohammed Jassim Mohammed Jassim, " The Impact of Software Process Improvements in Small and Medium Scale Enterprises ", (IJSCE) Volume 3 – 2013.

[14] Hanna Oktaba, "MoProSoft : A software process model for small enterprises " The first international research workshop for process improvement in small settings - Pittsburgh– 2005.

[15] Ita Richardson, Christiane gresse von wangenheim, "Why are small organizations different ? ", IEEE - 2007.

[16] Jessica Diaz, Juan Garbajosa, Jose A. Calvo-Manzano , "Mapping CMMI Level 2 to Scrum Practices: An Experience Report", EuroSPI - 2009

[17]K. Langre, P. Larson, M. Shaw," Software process improvement in small organizations: case study ", IEEE, 2005

[18] Karl Weigers, "Implementing software engineering in a small workgroup", Process Impact – Computer language 1993

[19] Karl weigers, "Software process improvement: 10 traps to avoid", software development magazine, 1996

[20] Keane, "Process Improvement Management", Keane White papers, 2010.

[21] Ken Schwaber, Jeff Sutherland, "The Scrum Guide", Scrum.org - 2013

[22] Kirk D. ,MacDonell S, "A systems approach to software process improvement in small organsations" Auckland university of technology - 2009

[23] Mark Staples, MahmoodNiazi, Ross Jeffery, Alan Abrahams,PaulByatt, Russell Murphy, "An exploratory study of why organizations do not adopt CMMI" , ScienceDirect, 2006.

[24] Muhammad Sulayman, "A Systematic Literature Review of Software Process Improvement for Small and Medium Web Companies ", Department of Computer Science - The University of Auckland - 2009.

[25] Naji Habra, Eustache Niyitugabira, Anne-Catherine Lamblin, Alain Renault, "Software Process Improvement in Small Organizations Using Gradual Evaluation Schema ",Information and Software Technology, Volume 50, 2008

[26] Neil Potter, Mary Sakry, "Implementing SCRUM (AGILE) And CMMI Together", The Process Group - 2009

[27] Noopur Davis, Julia Mullaney, " The team software process in practice: A summary of recent results ", SEI, 2003.

[28] Official Website, "IT Processes audit and improvement. CMMI and ITIL", Cynertia Consulting - 2013

[29] Official Website, "Software Testing Standards-CMM Levels", E-Testing Hub - 2012.

[30] Pink Elephant, "The ITIL Story White Paper ", Pink Elephant, 2004

[31] Renato Volpe, Farley nobre, marcelopessoa, Mauro Spinola, "The role of software process improvement into total quality management: an industrial experience ", IEEE, 2000

[32] Sandra cepeda, Suzanne Garcia, Jacquelyn langhout" Is CMMI useful and usable in small settings? One example ",The journal of defense software engineering - 2008

[33] Sergio F. Ochoa, José A. Pino, Luis A. Guerrero, César A. Collazos, "SSP: A Simple Software Process for Small-Size Software Development Projects", Springer US – Volume 219 - 2006.

[34] SrinivasChillara, Pete Deemer, "Scrum and CMMI: A High level assessment of compatibility", csaic - 2003

[35] Steven C Jackson, "Software Process Model and Metrics Adoption for small software organizations", scribd, 2009

[36] Watts Humphrey, CMMI, PSP and TSP Models, SEI - 2002.

[37] ZaheerHabib, "The critical success factors in implementation of software process improvement Efforts", University of Gothenburg – Master thesis, 2009

[38] Zhedan Pan, Hyuncheol Park, JongmoonBaik,Hojin Choi, " A Six Sigma Framework for Software Process Improvements and its Implementation", IEEE, 2007

Appendix 1: CMMI

Capability Maturity Model - CMMI:

Software applications are now part of almost every activity and aspect in our life, if not, it is at least part of most of them, Telecommunications, Automation and Machines, Medical Systems, Banking, Finance, Mobile, Internet … etc , in all of these, Software applications are involved, this involvement increases the risks of having a bad quality software and the benefits of having a good one, quality here means several things (working as expected, when expected, delivered on time and so on) , this in turn increased the importance of all software related issues.

Because software applications through several years became a very important aspect in our life as mentioned, and its importance is still increasing as time goes on, Several Methodologies – known as software process improvement methodologies (SPI) – have been used in order to improve the process of producing software, each of these methodologies has its own features and characteristics and way of implementation, Among these methodologies CMMI [7] is widely used and considered as one of the most important SPI methodologies.

CMMI

CMMI, Capability Maturity Model Integration, is an approach or model used by organizations in order to improve their performance and their process of doing their tasks, it is used not only for software development process but for other processes as well, it was originally initiated or created as a military demand for the united states government with the cooperation of SEI (Software Engineering Institute, Carnegie Mellon University), the main objective of this project in 1980s was to create a

reference model to be used to evaluate the software subcontractors, CMMI is an abbreviation of :

- Capability, when it is said that a process fails, then this is just saying in one word that the process doesn't have the capability to achieve its objectives, In CMMI the quality of the product depends to a large extent on the quality of the process used to develop the product.
- Maturity, If a process doesn't fail (has the capability to achieve its objectives), then the next step to consider is how much this capability is mature, in CMMI there are five levels or steps of maturity [section 1.3.2], in each level there are specific process areas and practices that need to be fulfilled in order to rank the process in that level.
- Model, CMMI is a model based on best practices and serves as a guide to improve the process of developing products.
- Integration, A capability maturity model was initially created and abbreviated (CMM), there was a CMM for software and other CMMs for hardware and systems, Later, and all different disciplines of CMM were integrated into one integrated model CMMI.

LEVELS

CMMI consists of Five levels representing the level of software process evaluation (Figure 1.1), the higher the better, each level is a pre-requisite for the following higher one, in order to start working on achieving level 3, level 2 is assumed to be already achieved and this applies for the higher levels, also to be ranked in a specific level all the KPAs (Key Process Areas) of that level must be fulfilled and implemented.

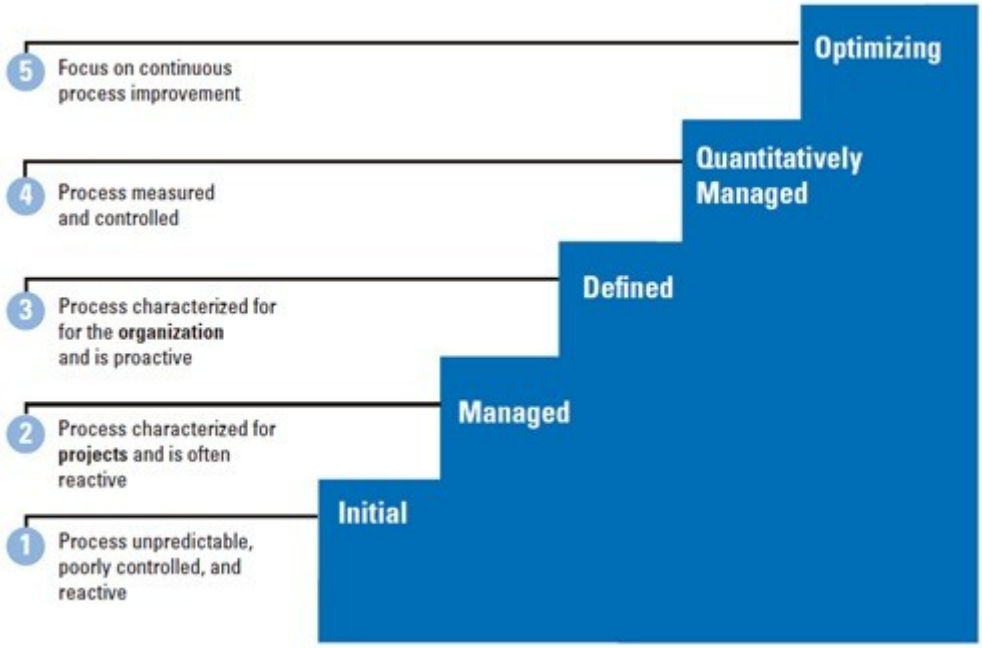


Figure 1.1: The Five levels of CMMI [28]

- **Maturity Level 1 – Initial:** in this level a process is used to be an ad hoc or chaotic process the quality is fully dependent on the individual skills and competences, the product are unpredictable and no level of quality can be expected, there is no defined process or systematic approach in the group work and progress.
- **Maturity Level 2 – Managed:** in level 2 there is a defined process, everything works as planned and according to policies, working members are also skilled and disciplined, level 2 is called (managed) since the status of the work product and the progress follow up is visible to management at several points and milestones, visibility to the management is not based on the final product or result with a success or fail evaluation.
- **Maturity Level 3 – Defined:** In level 2 the set of standards, procedures, and process descriptions may differ from a project to projects, but in the

same project they are totally linked to and consistent with each other to formulate a strong process, in Level 3 this is not the case, standards, descriptions and process related issues are not only consistent within the project but also linked to the institute standards and objectives and consistent with the institute policies and tendencies, this is a major part of being different than level 2. So once a project is initiated its process is already (Defined) and not to be defined for the project.

- **Maturity Level 4 – Quantitatively managed**: In level 4 the performance and quality of process are expressed in terms of numbers and statistical techniques, the quantitative objectives are based on requirements and needs of several things including customers, implementers, end users, organization and so on.
- **Maturity Level 5 – Optimizing:** In level 4 deviation from standards are discussed, their causes and results are expressed in numerical readings, In level 5 these causes are studied analyzed in order to make sufficient changes on the process itself so that the process is improved, improvement and innovation is the concern in this level in so that the process is in its best (optimal) case.

LEVEL2 KEY PROCESS AREAS (KPA)

CMMI consists of 22 key process areas distributed among the different levels (Figure 1.2).

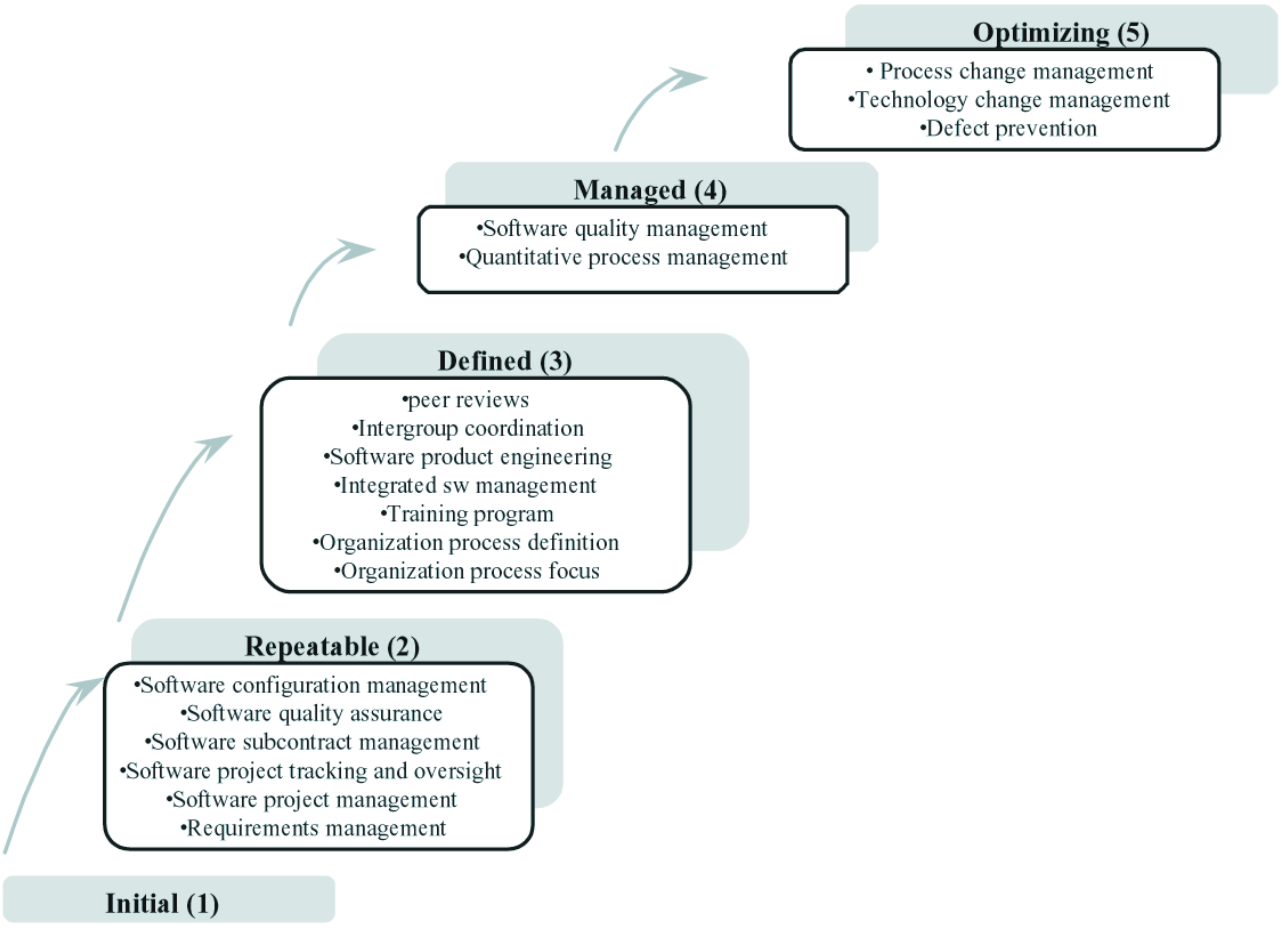


Figure 1.2: KPAs in CMMI levels [7]

Although Level 1 is the first level of CMMI but this level is the ad hoc level in any institute, no formal process, no disciplined criteria, it is based on the proficiency of those involved in development, the formal and systematic work starts at level 2 of CMMI, processes that are involved in this level (as well as any other level) are aggregated in what is known as Key Process Areas KPAs, all processes or activities that are related to requirement management are grouped together in the requirement

management KPA, and all processes or activities that are related to planning of the project are grouped together in the Project planning KPA and so on.

Level 2 consists of seven KPA :

1. Requirement Management (REQM).
2. Project Planning (PP).
3. Project Monitoring And Control (PMC).
4. Measurement And Analysis (MA).
5. Process And Product Quality Assurance (PPQA).
6. Configuration Management (CM)
7. Supplier Agreement (SA)

Appendix 2: The evaluation Form

Evaluation Form

Please write (×) where appropriate:

Strongly Disagree	Disagree	N/A	Agree	Strongly Agree	Evaluation Item	No.
					Terms used are Clear.	1
					Sequence of steps is reasonable and matches the actual work steps.	2
					Suggested Framework is applicable	3
					Framework is easy to apply and use.	4
					Using the framework does not form an obstacle in the actual work steps	5
					Suggested framework helps in improving the software development process.	6
					Suggested framework does not need a lot of training.	7
					Considering Optional and not optional sections of the framework is reasonable.	8
					You prefer to use the framework in your work.	9
					Your English language is between very good and excellent	10
					Worked in or have knowledge or familiar with any known SPI methodology (CMMI, ITIL, …etc).	11
					Terminology used is close to that that you know in the methodology you are familiar with.	12
					Sequence of steps used is close to that that you know in the methodology you are familiar with.	13
					Using Suggested framework helps in going towards using a well known SPI methodology	14
					Things that exist in SPI methodology and not in the framework are not needed for small scale workgroup	15

Any comments or suggestions:

_____

_____

Appendix 3: Evaluation Form in Arabic

نموذجتقييم

ضعإشار (؛) فيالخانةالمناسبة

رقم	بندالتقييم	أوافقبشدة	أوافق	لاادري	أعارض	أعارضبشدة
1	المصطلحاتالمستخدمةواضحةومفهومة .					
2	تسلسلالخطواتمنطقيويتناسبمعتسلسلخطواتالعملالفعلي .					
3	النموذجالمقترحقابلللتطبيق .					
4	النموذجالمقترحسهلللتطبيق .					
5	استخداماالنموذجلايشكلعائقلسير العملالفعلي .					
6	النموذجالمقترحيساعدفيتطويروتحسينعمليةإعدادالبرمجيات					
7	النموذجالمقترحلايحتاجللكثيرمنالتدريب					
8	اعتباربعضالخطواتفيالنموذجاختياريةوبعضهاإجباريةمناسبةومنطقي .					
9	تفضلاستخداماالنموذجفيعملك					
10	مستوىلغتكالانجليزيةمابينجيدجداوممتاز					
11	لديكمعرفةأواطلاعأوعملتبنموذجمعروفلتطويرعمليةإعدادالبرامج (SPI methodology) مثل (CMMI, ITIL, ... etc)					
12	المصطلحاتالمستخدمةفيالنموذجقريبةمنالمصطلحاتالمستخدمةفي (SPI nethodology) التياعرفها					
13	تسلسلالخطواتفيالنموذجقريبةمنتسلسلالخطواتفي (SPI nethodology) التياعرفها					
14	استخداماالنموذجالمقترحيساعدفيالانتقالإلى (.PI meth ؛).					
15	الأمورأوالعناصرالموجودةفي (PI methodology ؛) وليستموجودةفيالنموذجالمقترحلاتتناسبمعفرقالعملالبرمجيةالصغير .					

أيتعليقاتأومقترحاتعلىالنموذجالمقترح:

_____

_____

_____

Appendix 4: The framework

Template Framework

For Small Software Development Workgroups

Al-Quds University

Jerusalem

Requirement Management (RM)

Requirement evaluation criteria: While writing requirements of the application, each requirement should be checked, is it a "good" presentation of the requirement or not, to know if the requirement is "good" or not in terms of quality, there should be a reference and a list of measurement parameters (e.g. is it clear?, is it consistent with auditing rules? is it legal? and so on), some general (G) rules can be found in the framework demo, you can add your specific (S) parameters, write these parameters in the following table:

Requirement specification Characteristics evaluation criteria			
No	**G/S**	**Name**	**Description**

Reference sheet for requirement specification evaluation criteria

Every requirement should be stated in the following form:

Module		**Function Code**			Quality Criteria		
Function							
Description							
Input							
Source							
Output							
Destination							

Action						
				Committed staff		
Req. Source		**Date**				
Approved By		**Date**				
				Notes		

Fig5: requirement specification form with Commitment section added

- Module: the name of the project you are working on it is better if it is coded, a human resources system may be coded, HR Ver2, or a financial system may be Fin 1.0 …etc.
- Function Code: code of each function or requirement, a sequence number (001,002 …etc) may be sufficient.

Module		**Function Code**		Quality Criteria		
Function						
Description						
Input						
Source						
Output						
Destination						
Action						
				Committed staff		
Req. Source		**Date**				

Approved By		**Date**	

Notes		

Module		**Function Code**	

Function		
Description		
Input		
Source		
Output		
Destination		
Action		

Req. Source		**Date**	
Approved By		**Date**	

Quality Criteria		
Committed staff		

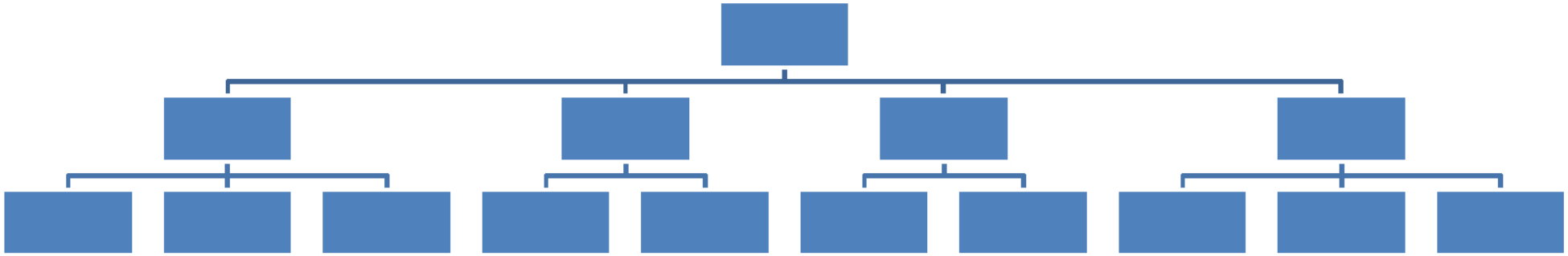
	Notes	

Project Planning (PP) and Project Monitoring And Control (PMC)

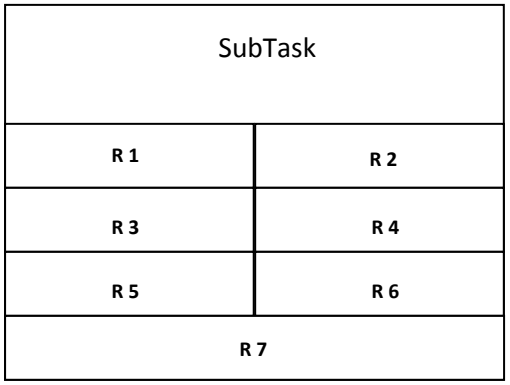
Start by filling the following tree (break down structure)

e.g. (HR: (1) handle employment (2) handle annual increase, (1.1) prepare employment sheets, (1.2) fill employment form … etc

(Don't forget to give numbers for each task and subtask – work package)



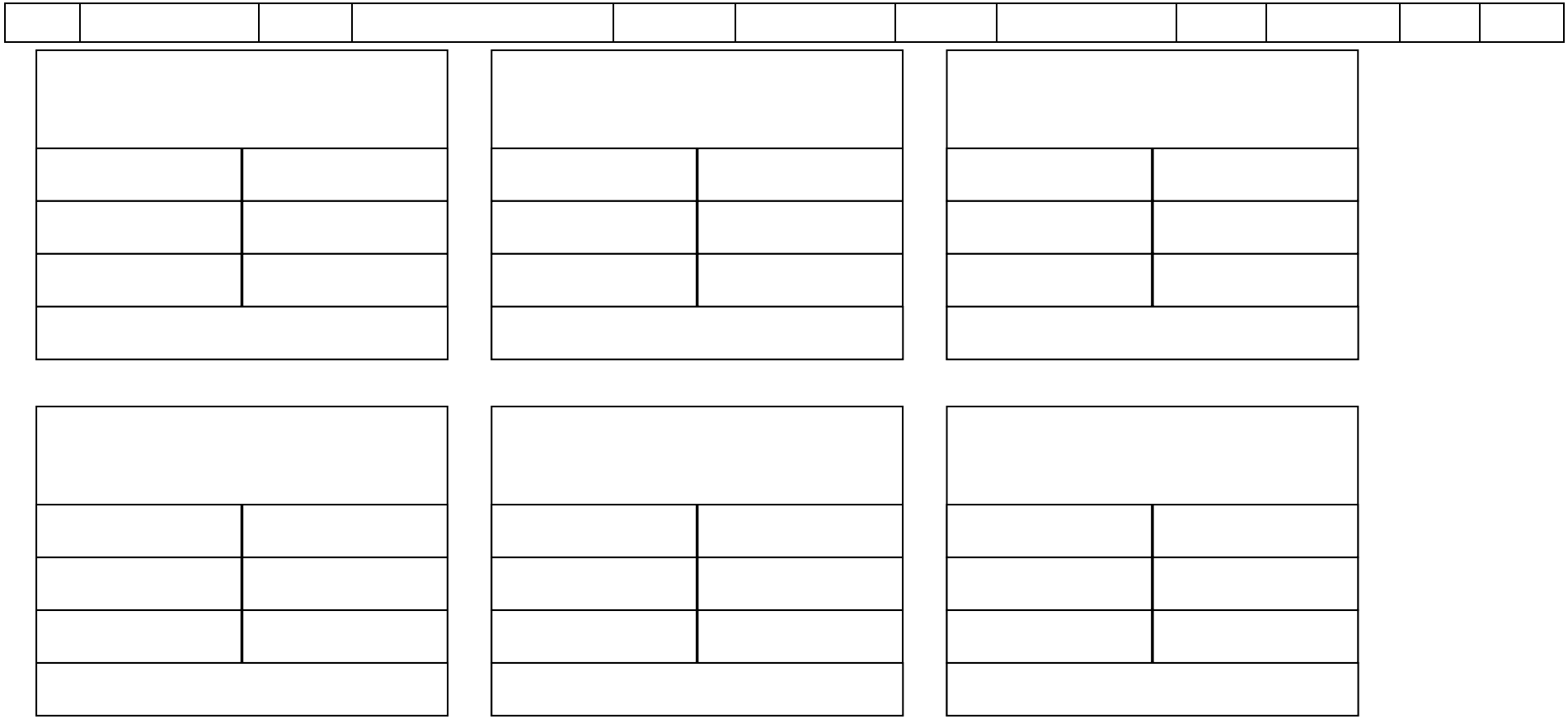
Work packages should be detailed as follows



- R1: Is externally Acquired (done somewhere else and to be used here), e.g. archive document function done in a lot of places.
- R2: If externally acquired, from where. (Which module? Which part?)
- R3: is to be used somewhere else.
- R4: Where to be used.
- R5: Estimated size, time required (5 days, 3 days, 15 days …).
- R6: Infra structure time, time required before starting the function work itself (e.g. special input device like barcode reader to be brought in order to work on the function).
- R7: Total time (R6 + R7).

Also fill the work packages in the following sheet (tabular form of the work break down structure). Use level column for the dependency and precedence creation, e.g. work packages that do not depend on any other subtasks are assigned level 0 and then those that depend on them are assigned level 1 and so on.

Task	Name	Sub task	Name	Is Ext Acquired	Where	To Be Reused	Where	Est. Size	Infra Struct. Time	Total Time	Level



Project Planning (PP) and Project monitoring and control (PMC):
Although Planning is the most important for any project, but still monitoring and controlling the plan is also important this is why there is a consideration in the following section of project monitoring and control (PMC), in the planning phase PMC (SHADED) sections are not filled but referred to in later phases while the project is in progress.

Project Phases:
Project passes several phases starting from being an idea up to a finished complete product. Write in this sheet the phases that the project will go through. While the project is in progress and for PMC reasons write down for each phase whenever it is finished and any comments regarding it.

No	Name	Description	Remarks	Pass date	PMC Remarks

Project Milestones Sheet:
When do you want to make a revision –especially with colleagues- of what is being done and what has been achieved so far and what about schedules and so on?

No	Name	Description	Date	Event	Remarks

Project Assumptions:
The project plan is based on specific assumptions (e.g. the arrival of a consultant on a specific date, the installation of a specific software or equipment and so on). These assumptions may occur and may not (write down the confidence of occurrence – low , medium, high)**.**

No	Name	Description	Confidence	Remarks

Project Constraints:
Some times there are things that are related to different constraints, e.g. financial module to be delivered before the end of fiscal year, the sales part to be finished before opening … etc). These constraints are to be written here.

No	Name	Description	Remarks

Project Risks:
Things and problems may appear during project implementation, the good planning includes taking into consideration the possibility of these problems to occur, brainstorming is the best starting point for detecting Risks within the project, list all risks in the following table with any related remarks to avoid the occurrence of the risk (e.g. the server performance may not be good enough, the electricity may not be stable one of the team members may leave the team and so on).While the project is in progress and for PMC reasons and as the risks are always monitored and controlled, write for each risk weather it has been turned into a problem (occurred) or not and what corrective action has been taken and if it has been passed (e.g. the server risk occurred, server was down, corrective action 1 was taken and backup server is used and the problem is passed)

No	Name	Description	Remarks	IS Problem?	Corrective action	Is Passed?

Project Corrective actions:
Writing down risks is not enough, it is important to know what to do in case a risk really occurs and transfers into a problem, also there are things that may happen but not considered as Risks because of their very low severity but still it is important to know what to do in each case, for a risk or non risk situation what corrective action to do (e.g. for Server problem use the backup server instead, for the lack of a specific tool say tool A use tool B instead with some customization).While the project is in progress and for PMC reasons whenever a corrective action is taken write the result of usage and any PMC remarks related to it.

No	Case	Description	Corrective Action	Result	PMC Remarks

Data management plan (OPTIONAL):

Depending on level of security of data and the amount of data and any other factor related to data, this section may be optional or mandatory, if needed to be used it only includes the listing of portions of data and who is responsible for collection or distribution of that portion of data, (e.g. for statistics about employee master data only Member A is allowed to collect data and extract statistics, for project follow up reporting only member B is assigned the task to produce feedback reports and so on). While the project is in progress and for PMC reasons write down any progress remarks concerning the data management plan entries.

No	Portion Of Data	Collect Or Dist.	User	Description	Remarks	Progress Remarks

Resource management plan:

Resources like hardware equipment or software component resources, e.g. servers, laptops, software packages (not related to process), Books, Training and so on. All of these (if related to the project) are documented and categorized in the resource management plan. For example a label printer must be available from 10/01/2011 to 25/01/2011 for testing issues and currently it is available, while a magnetic card printer is required for the same reason from 17/02/2011 to 20/02/2011 and it is not currently available. While the project is in progress and for PMC reasons write for each resource (if not available) if it has been provided and made available, also for each resource if it has been finished with or not, (e.g. Magnetic card printer has been brought but not finished with it yet, while label printing function has been tested and finished)

No	Description	Category	Description	Start Date	End Date	Available	Made Available	Finished?	Remarks

Required skills:

Some times there are some specific skills that must exist within the team at least by one of them to handle a specific job, when writing a java program by a java professional team, java is not considered a skill, while for a Delphi programming team may need the skill of writing a small java routine to do some kind of interface with a java program in this case knowing java is a skill to be taken care of. While the project is in progress and for PMC reasons write for each skill required (if not available) if it has been acquired and made available or not (e.g. one of the team members could collect all needed information for hardware interface).

No	Skill Description	Available	Made Available	mechanism	Remarks

Stockholder's involvement:

Stakeholders differ in their involvement in the tasks of the project while one of them is involved in a task as a reference another one is for approval and a third is not involved at all in this task. The following sheet is a matrix to show which of the stakeholders is involved in what task:

Stakeholder			
Task			

Optional:

The stakeholders task involvement matrix shows weather involvement exist or not using (0/1) flag but it doesn't show what is the kind of involvement, as an option it is possible to add a lookup sheet for kinds of involvement and considering their values in the matrix ex. (1 for approval, 2 for reference, 3 for both approval and reference, and so on) then the values are used in the matrix instead of 0/1 values

While the project is in progress and for PMC reasons write for each stakeholder any comments regarding his commitment in a sheet similar to that used for his involvement.

Stakeholder's involvement progress comments:

Stakeholder			
Task			

After that and also for PMC reasons write for each stakeholder his commitment mark (if correct to call it) and comments regarding SH commitment.

Stakeholder commitment remarks sheet:

No	Stakeholder	Commitment	Comments

Project Plan schedule:
Refer to the tabular sheet where every work package is assigned a level value and then fill in the following sheet from that sheet but ordered by level value so that lowest level of subtasks (in terms of dependency) are to be done first, Also refer to all what has been planned for like milestones or training and include them here according to their appropriate time of occurrence, the from – to fields are based on the starting of the project and the time required for each sub task. While the project is in progress and for PMC reasons write for each entry whenever it is finished (YES) and the amount of deviation from the scheduled time.

No	From	To	Activity Type	Activity	Activity Name	Finished?	Deviation

Measurement and analysis (MA)
(Optional But recommended)

Working on several projects increase experience and proficiency, in several things, time to deliver a product, accuracy in estimation, better functional understandings and so on. In this part of the framework, a list of assumed improvement sides is to be followed up.

Measurement Items:
Measurement items are these (improvements) that are expected to be achieved and to be measured if achieved or not or how much achieved, (e.g. better delivery time, or less quality problems, and so on.

M_ID	Name	Description	Base/ Derived	Related objective	Data to collect	How to collect	Repeat	How to analyze	Readings / Results	Analysis Result

Process and Product Quality Assurance (PPQA)

No one can say that an application or part of an application is **good** unless it is examined and checked to be good, to check a product 2 things should be known what should I test and when should I test to say that it is good. So for this section of the framework a list of products to test and these 2 things for each product are to be maintained.

Products to be evaluated:
What to evaluate is the first question in evaluation, Prepare a list of products that needs to be examined for quality (e.g. main package, all work packages – sub tasks …etc):

Product NO	Product description	Remarks

Evaluation criteria:
For each of the Listed Products, write down how this product is to be evaluated, the criteria (parameter) and how to make the evaluation:

Product NO	Evaluation criteria No	Criteria	Criteria Description	How To Check

Evaluation milestones or stages:
After the previous 2 sheets are filled then it is clear what and how to evaluate, now it is important to write down when to make the evaluations (e.g. at the prototype level, exactly before implementation … etc):

Evaluation Stage No	Description	Remarks

Product Quality check list:
For each of the Listed Products and their evaluation criteria, write down the result of evaluation in each of the stages or milestones that were listed in the previous section:

		Milestones / Stages For quality evaluation										Remarks
Product NO	Evaluation criteria No											

Result No	Description

Configuration management
Version control and Change management and tracking
(Optional But recommended)

Baseline Items:

Several components may be involved in change management and version control, packages, screens, reports, plans, security issues and so on, the starting point in following up these components is putting them all in a table in order to build later steps on, the items in this table are called baseline items, if the number of baseline items is large it is good to write down the category of the item it may be helpful in giving indications for later projects, categories may be (Plans, Requirements, Packages …etc) it also may be (Code, functionality, interface … etc) :

Baseline Item No	Description	Category (optional)

Change requests:

Any change in the system is based on a change request, change requests are to be written in the following table, each change request must have a number. Who requested this change is also written down and the most important for this stage is the change description that describes in details what is the required change, then an analysis is done to the requested change (what does it affect, which baseline items are affected, who should approve the request, who should handle it and follow it up, and so on) then the status of the request is updated step by step it may be (new, approved, in progress, closed … etc) :

ChReqNo	Requested by	Change description	Change analysis	Status

Change follow-up:
After a change request is analyzed and approved to be done and assigned as a task to one (or more) of the team the change is to be handled and manipulated, Any change in the system is based on a change request, and may be connected to a baseline item, One change may be related to a change request, but also in other cases several changes may be connected to one request according to effect of the change requested:

ChangeNo	Date	ChReqNo	Baseline Item No	Change description	Remarks

Version Control:
Any change made to any component of the system (baseline item) transforms that item from a state to a newer state (version), each version after the first one has an old or previous and a new state, e.g. a specific form or screen (registered as a baseline item) shows the list of employee vacations but it doesn't show the remaining balance of the employee vacations, a request is issued by the HR contact person that the balance should appear on the screen, a change is done to the screen by one of the programmers, the screen is now working fine, in version control a number is assigned to the version of the screen and in he previous section written (without vacations balance), in the new portion written (with balance added):

VerCtrlNo	Date	Baseline item	Version	Previous	new	User	Change no	Remarks

Log Table for task estimated and actual time
(Optional but recommended)

Example: (Estimated time per unit: 10)

	For all tasks					For tasks estimated as 1 unit				
Task	**Estimated No of units**	**Est. Time**	**Actual Time**	**Unit Act. Time**	**Unit Error**	**Single unit? (1/0)**	**Est. Time**	**Act. Time**	**Unit Error**	**Remarks**
	Sum	**Sum**	**Sum**	**Avg.**	**Avg.**	**Sum**	**Sum**	**Sum**	**Avg.**	

Appendix 5: the framework example guide

Template Framework

For Small Software Development Workgroups

"Example Data"

Al-Quds University

Jerusalem

Requirement Management (RM)

Requirement specification Characteristics evaluation criteria			
No	**G/S**	**Name**	**Description**
1	G	Complete	The requirement is described completely – no missing information.
2	G	Consistent	The information written doesn't contradict with other requirements.
3	S	GAP	Requirement meets the Accounting Generally Accepted Principles.
4	S	GOV	Requirement is consistent with governmental rules.

Reference sheet for requirement specification evaluation criteria

Module	**HR**	**Function Code**	HR-003	Quality Criteria		
				1	Complete	√
Function	**Archive Employee Documents**			2	Consistent	√
Description	Archive All Employee Documents (CV, Certificates, Work Experience, recommendations …etc).			3	GAP	√
				4	GOV	√
Input	Employee no, and scanned documents					
Source	User input and Scanner					
Output	Scanned images connected to employee					
Destination	Archive File					
Action	Read Employee number from user input. Get employee information from employee master file. Check If earlier archives exist. If earlier archives exist delete them from archive file.					
				Committed staff		
Req. Source	Mr. Joe - contact	**Date**	15/01/2012	1	All	
Approved By	Mrs. Jane - HR	**Date**	17/01/2012	2	..	
				3	..	
				4	..	
				Notes		
				1	Reference – Joe	
				2		

Module	HR	Function Code	HR-017
Function	**Change Employee Degree**		
Description	Assign the employee a different degree than the current assigned one, doesn't apply for employees under probation period		
Input	Employee no with current and new degrees		
Source	Employee Master File		
Output	Employee number with new degree		
Destination	Employee Master File		
Action	Read Employee Information From master File. Check If the employee is under probation or not If Not then assign him the new degree		

Req. Source	Mr. Joe – contact	Date	15/01/2012
Approved By	Mrs. Jane – HR	Date	17/01/2012

Quality Criteria		
1	Complete	√
2	Consistent	√
3	GAP	√
4	GOV	√

Committed staff		
1	All	
2	..	
3	..	
4	..	

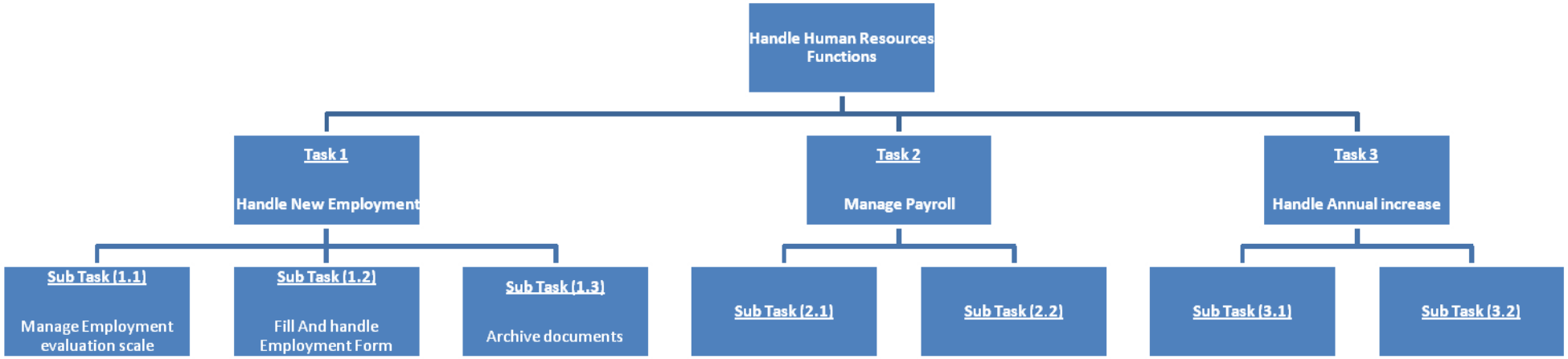
Notes	
1	Reference – Joe
2	

Project Planning (PP) and Project Monitoring And Control (PMC)

Start by filling the following tree (break down structure)

e.g. (HR: (1) handle employment (2) handle annual increase, (1.1) prepare employment sheets, (1.2) fill employment form … etc

(Don't forget to give numbers for each task and subtask – work package)



Work packages should be detailed as follows

SubTask(1.3)	
YES	ARS-001
NO	-
1 unit – 0.5 day	1 unit – 0.5 day
2 units – 1 day	

- R1: Is externally Acquired (YES).
- R2: If externally acquired, from where. (Archiving system : ARS-001)
- R3: is to be used somewhere else. (NO)
- R4: Where to be used.(-)
- R5: Estimated size, time required (1 unit of time – unit estimated 0.5 day).
- R6: Infra structure time, time required before starting the function work itself (1 unit – 0.5 day).
- R7: Total time (2 units – 1 day).

Also fill the work packages in the following sheet (tabular form of the work break down structure). Use level column for the dependency and precedence creation, e.g. work packages that do not depend on any other subtasks are assigned level 0 and then those that depend on them are assigned level 1 and so on.

Task	Name	Sub task	Name	Is Ext Acquired	Where	To Be Reused	Where	Est. Size	Infra Struct. Time	Total Time	Level
1	Handle Employment	1.1	Manage employment evaluation scale	No	-	NO	-	5	2	7	0
1	Handle Employment	1.2	Handle employment form	No	-	NO	-	6	0	6	1
1	Handle Employment	1.3	Archive docs	Yes	ARS-001	NO	-	1	1	2	0
2	Manage payroll	2.1									
2	Manage Payroll	2.2									

Project Planning (PP) and Project monitoring and control (PMC):
Although Planning is the most important for any project, but still monitoring and controlling the plan is also important this is why there is a consideration in the following section of project monitoring and control (PMC), in the planning phase PMC (SHADED) sections are not filled but referred to in later phases while the project is in progress.

Project Phases:
Project passes several phases starting from being an idea up to a finished complete product. Write in this sheet the phases that the project will go through. While the project is in progress and for PMC reasons write down for each phase whenever it is finished and any comments regarding it.

No	Name	Description	Remarks	Pass date	PMC Remarks
1	Init	Initiation Of project.		16/05/2010	Initiation documents where reserved as hard copy only
2	Analyze	Analysis of requirements			
3	Develop	Actual development			
4	Test	Testing product			

Project Milestones Sheet:
When do you want to make a revision –especially with colleagues- of what is being done and what has been achieved so far and what about schedules and so on?

No	Name	Description	Date	Event	Remarks
1	Month1	First Month Review–before mgt meeting	01/10/2011		
2	Month2	Second Month Review–before mgt meeting	01/11/2011		
3	Risk	If a risk occurs and manipulated – review		Any Risk Occurrence	

Project Assumptions:
The project plan is based on specific assumptions (e.g. the arrival of a consultant on a specific date, the installation of a specific software or equipment and so on). These assumptions may occur and may not (write down the confidence of occurrence – low , medium, high)**.**

No	Name	Description	Confidence	Remarks
1	Consultant	HR Consultant to arrive at 22/09	100%	
2	Audit Process	Auditing process not to start before 15/12	80%	During auditing process – difficult communication

Project Constraints:
Some times there are things that are related to different constraints, e.g. financial module to be delivered before the end of fiscal year, the sales part to be finished before opening … etc). These constraints are to be written here.

No	Name	Description	Remarks
1	Committee	The IT Committee holds a meeting in 17/08 a prototype must be ready for demo.	
2	Fiscal Year	The Fiscal year ends at 31/12 all modules should be delivered before 31/12	

Project Risks:

Things and problems may appear during project implementation, the good planning includes taking into consideration the possibility of these problems to occur, brainstorming is the best starting point for detecting Risks within the project, list all risks in the following table with any related remarks to avoid the occurrence of the risk (e.g. the server performance may not be good enough, the electricity may not be stable one of the team members may leave the team and so on).While the project is in progress and for PMC reasons and as the risks are always monitored and controlled, write for each risk weather it has been turned into a problem (occurred) or not and what corrective action has been taken and if it has been passed (e.g. the server risk occurred, server was down, corrective action 1 was taken and backup server is used and the problem is passed)

No	Name	Description	Remarks	IS Problem?	Corrective action	Is Passed?
1	Server	Server performance may be week mainly because of its memory	Increase server memory to decrease risk possibility	YES	1	YES
2	Member	One of team members may leave the team.				

Project Corrective actions:

Writing down risks is not enough, it is important to know what to do in case a risk really occurs and transfers into a problem, also there are things that may happen but not considered as Risks because of their very low severity but still it is important to know what to do in each case, for a risk or non risk situation what corrective action to do (e.g. for Server problem use the backup server instead, for the lack of a specific tool say tool A use tool B instead with some customization).While the project is in progress and for PMC reasons whenever a corrective action is taken write the result of usage and any PMC remarks related to it.

No	Case	Description	Corrective Action	Result	PMC Remarks
1	Server	Server performance is week, or server down	Use backup server, keep backup server maintained		
2	Tool A	Tool A may not be available any more	Use tool B instead with specific customization		

Data management plan (OPTIONAL):

Depending on level of security of data and the amount of data and any other factor related to data, this section may be optional or mandatory, if needed to be used it only includes the listing of portions of data and who is responsible for collection or distribution of that portion of data, (e.g. for statistics about employee master data only Member A is allowed to collect data and extract statistics, for project follow up reporting only member B is assigned the task to produce feedback reports and so on). While the project is in progress and for PMC reasons write down any progress remarks concerning the data management plan entries.

No	Portion Of Data	Collect Or Dist.	User	Description	Remarks	Progress Remarks
1	Master Emp Data	Collect	Member A	Data regarding master employee Files		
2	Follow up reports	Distribute	Member B	Providing feedback regarding progress	As agreed with stakeholders	

Resource management plan:

Resources like hardware equipment or software component resources, e.g. servers, laptops, software packages (not related to process), Books, Training and so on. All of these (if related to the project) are documented and categorized in the resource management plan. For example a label printer must be available from 10/01/2011 to 25/01/2011 for testing issues and currently it is available, while a magnetic card printer is required for the same reason from 17/02/2011 to 20/02/2011 and it is not currently available. While the project is in progress and for PMC reasons write for each resource (if not available) if it has been provided and made available, also for each resource if it has been finished with or not, (e.g. Magnetic card printer has been brought but not finished with it yet, while label printing function has been tested and finished)

No	Description	Category	Description	Start Date	End Date	Available	Made Available	Finished?	Remarks
1	Label Printer	PRN	Printers	10/01/2011	25/01/2011	YES	-	YES	For testing the function of printing labels
2	Magnetic card printer	PRN	Printers	17/02/2011	20/02/2011	NO	YES		For testing the function of printing magnetic cards

Required skills:

Some times there are some specific skills that must exist within the team at least by one of them to handle a specific job, when writing a java program by a java professional team, java is not considered a skill, while for a Delphi programming team may need the skill of writing a small java routine to do some kind of interface with a java program in this case knowing java is a skill to be taken care of. While the project is in progress and for PMC reasons write for each skill required (if not available) if it has been acquired and made available or not (e.g. one of the team members could collect all needed information for hardware interface).

No	Skill Description	Available	Made Available	mechanism	Remarks
1	Java programming	YES	YES		member A knows java well
2	Interface with external hardware equipment	NO	YES	Manuals and related books of equipment involved	

Stockholder's involvement:

Stakeholders differ in their involvement in the tasks of the project while one of them is involved in a task as a reference another one is for approval and a third is not involved at all in this task. The following sheet is a matrix to show which of the stakeholders is involved in what task:

Stakeholder / **Task**	S.H 1	S.H 2	S.H 3
1.1	1	1	0
1.2	1	0	0
1.3	0	0	1
2.1	1	1	0

Optional:

The stakeholders task involvement matrix shows weather involvement exist or not using (0/1) flag but it doesn't show what is the kind of involvement, as an option it is possible to add a lookup sheet for kinds of involvement and considering their values in the matrix ex. (1 for approval, 2 for reference, 3 for both approval and reference, and so on) then the values are used in the matrix instead of 0/1 values

While the project is in progress and for PMC reasons write for each stakeholder any comments regarding his commitment in a sheet similar to that used for his involvement.

Stakeholder's involvement progress comments:

Stakeholder / **Task**	S.H 1	S.H 2	S.H 3
1.1			
1.2			
1.3			
2.1			

After that and also for PMC reasons write for each stakeholder his commitment mark (if correct to call it) and comments regarding SH commitment.

Stakeholder commitment remarks sheet:

No	Stakeholder	Commitment	Comments
1	S.H 1	80%	Not always available for approval.

2	S.H 2	100%	Always prepared, ready and respond on time

Project Plan schedule:
Refer to the tabular sheet where every work package is assigned a level value and then fill in the following sheet from that sheet but ordered by level value so that lowest level of subtasks (in terms of dependency) are to be done first, Also refer to all what has been planned for like milestones or training and include them here according to their appropriate time of occurrence, the from – to fields are based on the starting of the project and the time required for each sub task. While the project is in progress and for PMC reasons write for each entry whenever it is finished (YES) and the amount of deviation from the scheduled time.

No	From	To	Activity Type	Activity	Activity Name	Finished?	Deviation
1	01/10/2001	10/01/2001	Task	1.1			
2	11/01/2001	15/01/2001	Task	1.2			
3	16/01/2001	17/01/2001	Milestone	M 1			
4	.	.	Task	2.1			
5	.	.	Task	2.2			
6	.	.	Training	T 1			
7			Task	T3.1			

Measurement and analysis (MA)
(Optional But recommended)

Working on several projects increase experience and proficiency, in several things, time to deliver a product, accuracy in estimation, better functional understandings and so on. In this part of the framework, a list of assumed improvement sides is to be followed up.

Measurement Items:
Measurement items are these (improvements) that are expected to be achieved and to be measured if achieved or not or how much achieved, (e.g. better delivery time, or less quality problems, and so on.

M_ID	Name	Description	Base/ Derived	Related objective	Data to collect	How to collect	Repeat	How to analyze	Readings / Results	Analysis Result
1	Quality	Quality before implementation	D		(ok) in quality check list	count	once	Number / total number	28 Ok 40 Total	28/40 = 70%, 68% for last project, 62% one before
2	Accuracy	Error in estimation	D		Log table result calculation	Refer to log	once	Refer To Log	1.75%	1.78% for last project, 2.1% for one before

Process and Product Quality Assurance (PPQA)

No one can say that an application or part of an application is **good** unless it is examined and checked to be good, to check a product 2 things should be known what should I test and when should I test to say that it is good. So for this section of the framework a list of products to test and these 2 things for each product are to be maintained.

Products to be evaluated:
What to evaluate is the first question in evaluation, Prepare a list of products that needs to be examined for quality (e.g. main package, all work packages – sub tasks …etc):

Product NO	Product description	Remarks
1	Main package	
2	All Subtasks	Refer to Product quality check list
3	Reports	General template checkout

Evaluation criteria:
For each of the Listed Products, write down how this product is to be evaluated, the criteria (parameter) and how to make the evaluation:

Product NO	Evaluation criteria No	Criteria	Criteria Description	How To Check
1	1	Accuracy	Check for accuracy of result	Different inputs, compare with manual calculation
1	2	Performance	Check if it does what it should do fast enough	Try complex cases and observe with user.
2	1	SRS	Meets the requirement specification in the requirement sheet	Check input and output and compare to Specs
2	2	Interface	Check If interface is consistent with others and accepted by user.	take continuous interface feedback from user
2	3	Integration	Check integration with other related parts	Manipulate data and check other modules
2	4	User	User Acceptance testing	User makes the testing that the programmer made

Evaluation milestones or stages:
After the previous 2 sheets are filled then it is clear what and how to evaluate, now it is important to write down when to make the evaluations (e.g. at the prototype level, exactly before implementation … etc):

Evaluation Stage No	Description	Remarks
1	Prototype	interface And business logic related issues
2	Work package finish	Tested by programmer before going on to work package.
3	Work package closure	Before marking the work package as done.
4	Final	At the final stage directly before implementation

Product Quality check list:
For each of the Listed Products and their evaluation criteria, write down the result of evaluation in each of the stages or milestones that were listed in the previous section:

		Milestones / Stages For quality evaluation										Remarks
Product NO	Evaluation criteria No	1	2	3	4	.	.	.	.			
1	1	ok	ok	ok								
1	2	ok	ok	ok								
2	1	ok	ok	201								
2	2	ok	202	ok								
2	3											
2	4											

Result No	Description
201	The work package failed to give out the expected results as indicated by requirement specification
202	The user noticed a difference in navigation method between the screen and the other screens

Configuration management
Version control and Change management and tracking
(Optional But recommended)

Baseline Items:
Several components may be involved in change management and version control, packages, screens, reports, plans, security issues and so on, the starting point in following up these components is putting them all in a table in order to build later steps on, the items in this table are called baseline items, if the number of baseline items is large it is good to write down the category of the item it may be helpful in giving indications for later projects, categories may be (Plans, Requirements, Packages …etc) it also may be (Code, functionality, interface … etc) :

Baseline Item No	Description	Category (optional)
HR-001	Manage employment Evaluation scale	Requirement
HR-002	Handle Employment Form	Requirement
Calculate-Salaries	A package including all methods for salary related calculations	Package

Change requests:
Any change in the system is based on a change request, change requests are to be written in the following table, each change request must have a number. Who requested this change is also written down and the most important for this stage is the change description that describes in details what is the required change, then an analysis is done to the requested change (what does it affect, which baseline items are affected, who should approve the request, who should handle it and follow it up, and so on) then the status of the request is updated step by step it may be (new, approved, in progress, closed … etc) :

ChReqNo	Requested by	Change description	Change analysis	Status
001	Hr- Assistant	Add deductions possibility	Adding a place in salary form for deduction value	Done
002	Manager	Approval Mechanism	Any financial transaction - approved my management	Pending

Change follow-up:
After a change request is analyzed and approved to be done and assigned as a task to one (or more) of the team the change is to be handled and manipulated, Any change in the system is based on a change request, and may be connected to a baseline item, One change may be related to a change request, but also in other cases several changes may be connected to one request according to effect of the change requested:

ChangeNo	Date	ChReqNo	Baseline Item No	Change description	Remarks
10/105					
10/106					
10/107	04/10/2011	025	HR-012	Include Deduction possibility	
10/108	04/10/2011	003	Calculate salaries	Take into consideration deduction value	

Version Control:
Any change made to any component of the system (baseline item) transforms that item from a state to a newer state (version), each version after the first one has an old or previous and a new state, e.g. a specific form or screen (registered as a baseline item) shows the list of employee vacations but it doesn't show the remaining balance of the employee vacations, a request is issued by the HR contact person that the balance should appear on the screen, a change is done to the screen by one of the programmers, the screen is now working fine, in version control a number is assigned to the version of the screen and in he previous section written (without vacations balance), in the new portion written (with balance added):

VerCtrlNo	Date	Baseline item	Version	Previous	new	User	Change no	Remarks
126								
127								
128	04/11/2011	HR-012	4	No deduction option	With deduction option added	Mem.1	10/107	
129	05/11/2011	HR-012	5	Printing not working	Calling report is fixed	Mem.1		
130	05/11/2011	Calc-sal	24	Deduction not considered	Deduction is considered	Mem.3	10/108	

Log Table for task estimated and actual time
(Optional but recommended)

Example: (Estimated time per unit: 10)

	For all tasks					For tasks estimated as 1 unit				
Task	**Estimated No of units**	**Est. Time**	**Actual Time**	**Unit Act. Time**	**Unit Error**	**Single unit? (1/0)**	**Est. Time**	**Act. Time**	**Unit Error**	**Remarks**
1	**1**	**10**	**8**	**8**	**2**	**1**	**10**	**8**	**2**	
2	**1**	**10**	**9**	**9**	**1**	**1**	**10**	**9**	**1**	
3	**2**	**20**	**19**	**9.5**	**0.5**	**0**				
4	**1**	**10**	**7**	**7**	**3**	**1**	**10**	**7**	**3**	
5	**5**	**50**	**45**	**9**	**1**	**0**				
6	**3**	**30**	**35**	**11.9**	**-1.9**	**0**				
7	**1**	**10**	**11**	**11**	**-1**	**1**	**10**	**11**	**-1**	
8	**2**	**20**	**18**	**9**	**1**	**0**				
9	**2**	**20**	**21**	**10.1**	**-0.1**	**0**				
10	**4**	**40**	**38**	**9.5**	**0.5**	**0**				
	Sum	**Sum**	**Sum**	**Avg.**	**Avg.**	**Sum**	**Sum**	**Sum**	**Avg.**	
	22	**220**	**211**	**10.4**	**0.6**	**4**	**40**	**35**	**1.25**	

- There are 4 single unit tasks so in non single tasks there are (22-4 = 18) units Estimated: (18*10=180) units of time → Actual (211-35=176) units of time
- Accuracy (for non single unit tasks) = 176/180 = 0.98 → error = 1-0.98 – 0.02
- Accuracy (for all tasks) = 211/220 = 0.959 = 0.96 → error = 1-0.96 = 0.04
- Accuracy (for single unit tasks) = 35/40 = 0.875 = 0.88 → error = 1-0.88 = 0.12
- Error in estimation of no of units = (0.02 – (0.02*0.12)) = 0.0175 → Accuracy = 1-0.0175 = 0.985 = 98.5%