

*Master Thesis in
Software Engineering
Thesis no: MSE-2001-16
October 2002*



Evaluation of Software Projects

– A Recommendation for Implementation

The Iterating Evaluation Model

Gustav Sochacki

Department of
Software Engineering and Computer Science
Blekinge Institute of Technology
Box 520
SE - 372 25 Ronneby
Sweden

This thesis is submitted to the Department of Software Engineering and Computer Science at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:

Author:

Gustav Sochacki

Address: Hantverkaregatan 7B, SE-211 55 Malmö, Sweden

E-mail: gustav.sochacki@euromail.se

University advisor:

Bertil Rolf

Department of Business and Management

Department of
Software Engineering and Computer Science
Blekinge Institute of Technology
SE - 372 25 Ronneby
Sweden

Internet : www.ipd.bth.se
Phone : +46 457 38 50 00
Fax : +46 457 271 25

Abstract

Software process improvement (SPI) is generally associated with large organizations. Large organizations have the possibilities to fund software process improvement programs as large scale activities. Often these improvement programs do not show progress until some time has elapsed. The Capability Maturity Model can take one year to implement and not until then can measures be made to see how much quality increased.

Small organizations do not have the same funding opportunities but are still in need of software process improvement programs. Generally it is better to initiate a software process improvement program as early as possible, no matter what size of organization. Although the funding capabilities for small organizations are less compared to large organizations, the total required funding will still be smaller than in large organizations. The small organization will grow and overtime become a midsized or large organization, so by starting an improvement program at an early stage the funding overall should be minimized. This becomes more visible when the organization has grown large.

This master thesis presents the idea of implementing a software process improvement program, or at least parts of it, by evaluating the software project. By evaluating a project the specific needs that are most critical are implemented in the next project. This process is iterated for each concluded project.

The master thesis introduces the Iterating Evaluation Model based on an interview survey. This model is compared to an already existing model, the Experience Factory.

Keywords: Process Improvement, CMM, Evaluation, Software Project Process, Experience Factory

Table of Contents

Chapter 1 Preface	1
1.1 The Problem definition.....	1
1.2 The Goal	1
1.3 The Outline of the Master Thesis	1
Chapter 2 Introduction.....	2
2.1 Why do we need to evaluate our software projects?	2
2.1.1 The problem for small organizations.....	2
2.1.2 Goals and subgoals	2
2.1.3 Evaluation overview	4
Chapter 3 Software Process Improvement	5
3.1 Process defined.....	5
3.2 Process versus product oriented focus.....	5
3.3 Elements of quality.....	6
3.3.1 Defining quality.....	6
3.3.2 Software quality assurance	7
3.3.3 Software Quality Attributes.....	8
3.3.4 Quantification of Software Quality Attributes	8
3.4 Measuring the improvement.....	9
3.5 Why goals are important	9
3.6 Defining goals for evaluations.....	9
3.6.1 The interest groups	10
3.7 CMM	10
3.7.1 Short introduction to the CMM	10
3.7.2 Benefits from CMM	11
3.7.3 CMM obstacles.....	12
Chapter 4 A Field Survey for a Model	13
4.1 The Field Survey execution.....	13
4.2 The Interviews result	13
4.3 Analysis	15
4.3.1 Input/Feedback	15
4.3.2 Process.....	16
4.3.3 People	16
4.3.4 Measures.....	17

4.3.5 Time.....	17
4.3.6 Customer.....	18
4.3.7 Organization.....	18
4.3.8 Management.....	19
4.4 Summary.....	19
Chapter 5 The Iterating Evaluation Model.....	21
5.1 The software engineering model.....	21
5.2 The evaluation model.....	22
5.2.1 The action plan.....	24
Chapter 6 Organization.....	26
6.1 Introduction.....	26
6.1.1 Aspects of formal inspection — against.....	26
6.1.2 Resources.....	26
6.1.3 Aspects of formal inspections — for.....	26
6.1.4 Short introduction to formal inspection.....	27
6.2 The Organization of the Evaluation.....	27
6.2.1 Roles.....	27
6.2.2 Time management.....	27
6.2.3 Complementary roles.....	29
6.3 Training.....	29
6.4 The Change Process.....	29
6.5 Documentation.....	30
Chapter 7 Implementation.....	32
7.1 How to begin — pre-evaluation phase.....	32
7.1.1 Setting goals.....	32
7.1.2 Training.....	32
7.1.3 Documentation.....	32
7.2 How to conduct the evaluation.....	33
7.2.1 The Improvement Action Plan.....	33
7.2.2 Areas of evaluation.....	33
7.3 The evaluation of the evaluation — post-evaluation.....	34
7.4 Support and top level management.....	34
Chapter 8 Experience Factory.....	35
8.1 Introduction.....	35
8.2 The Quality Improvement Paradigm.....	35

8.3 The Experience Factory.....	37
8.3.1 Examples of Packaged Experience.....	38
8.3.2 Examples of Experience Packages	38
8.3.3 Summary	39
Chapter 9 Evaluation of the Iterating Evaluation Model and the Experience Factory vs. the interview survey	40
9.1 Experience Factory vs. Iteration Evaluation Model	41
Chapter 10 Summary.....	43
10.1 Conclusion.....	43
10.2 Strength and weaknesses	43
10.3 Future work	44

List of Figures

Figure 2-1. The Evaluation of Software Projects	3
Figure 2-2. The bond between the software process and the evaluation process.	3
Figure 3-1. The aspects of the Process [Zahran]	5
Figure 3-2. The software quality assurance resource allocation in a large and small organization .	7
Figure 3-3. Three groups of interest groups in the evaluation process.....	10
Figure 5-1. The Software Engineering Model.....	21
Figure 5-2. The evaluation activity in the SE-model.....	22
Figure 5-3. Data collection in the pre-study of the evaluation	23
Figure 5-4. Goal of the evaluation phase, the action plan	24
Figure 5-5. The action plan	24
Figure 6-1. Overview of the organization of the formal inspection	27
Figure 6-2. The process of creating the IAP-document.....	28
Figure 7-1. The iterating evaluation model	33
Figure 8-1 The Quality Improvement Paradigm, QIP	36
Figure 8-2 The Experience Factory and the Project Organization	38
Figure 10-1. The Boehm software quality model.....	45
Figure 10-2. McCall software quality model.....	46

List of Tables

Table 3-1 Software Quality Attributes	8
Table 4-1 Interview survey results	14
Table 4-2 Company data	15
Table 9-1 Data analyzed for the Iterating Evaluation Model and the EF	41

List of abbreviations

CMM	Capability Maturity Model
EAP	Evaluation Action Plan
EF	Experience Factory
EFO	Experience Factory Organization
E-protocol	Evaluation protocol
IAP	Improvement Action Plan
IEM	Iterating Evaluation Model
In housing	A project that is carried out in a local location by the organization that is responsible for the project.
QIP	Quality Improvement Paradigm
SPI	Software Process Improvement
SPICE	Software Process Improvement and Capability dEtermination
SQA	Software Quality Assurance

Chapter 1

Preface

1.1 The Problem definition

Evaluating a project is a fundamental step towards improving the organization. Today we have the CMM and SPICE that are based on best practice and give a method of what to do to improve the organization. However, these models are not so appropriate for small or medium sized organizations. The CMM and SPICE are very effort intense and costly to perform and the measured benefit is often not seen in a short time perspective.

One natural step would be to evaluate each specific project after it has finished and use this information to improve the oncoming project(s) and the organization. This evaluation will not only give feedback for improvement but is also more specific for the organization using it. In addition, a deeper understanding of the organization will create a closer approach to the different maturity levels described in the CMM.

1.2 The Goal

This research will result in a recommendation of how to perform an evaluation that the small organization can use to perform improvements, and see the benefits early.

1.3 The Outline of the Master Thesis

This master thesis has three major parts:

- the interview survey
- the creation of an evaluation model
- a description of an existing model, the Experience Factory

Analysis is made to find an appropriate way to evaluate a software project and improve that organizations software process by evaluation.

The aim is small organizations.

Chapter 2

Introduction

2.1 Why do we need to evaluate our software projects?

The software industry has had problems during a long time with projects over the budget or/and products with wrong functionality. During the 1990's, new models and techniques have been developed for improving the software development process. Two of the more known is the *Capability Maturity Model*, referred to as the CMM and the *Software Process Improvement and Capability dEtermination* model also referred to as the SPICE model. These models have become a de facto standard for improving and measuring the software process and enabling organizations to control many of the problems associated with software development.

2.1.1 The problem for small organizations

The software process improvement models are comprehensive and resource demanding when implemented. This is one of the major obstacles for small organizations; they do not have the possibility to devote all the resources and funding needed to implement a software process improvement program to its full extent.

Implementing the models of CMM or SPICE into a large organization is more demanding than implementing parts of the models into a small growing organization. This can be a problem for large organizations since the implementation requires more time, resource and funding. For the small organization this is a problem, therefore implementing parts of a SPI-model is the most appropriate way for small organizations.

The question is what parts should be chosen? Alternatively, maybe rather ask the question how to choose which parts.

This is the key issue of the evaluation; to find prioritized parts from a SPI model to improve the software process by evaluating software projects.

2.1.2 Goals and subgoals

By using evaluations that can show defects inside the software process and then perform improvements specified by a SPI model removing the defects, one attains a stable growth of maturity inside the organization. This is illustrated in figure 2-1. In the CMM there are five levels defined, by which the first one is total chaos. It is all about avoiding that chaos.

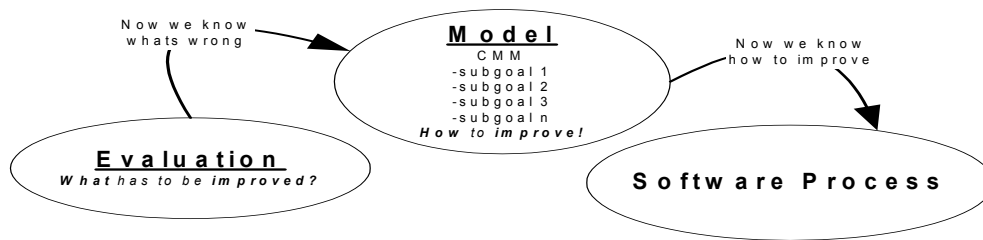


Figure 2-1. The Evaluation of Software Projects

The benefit of evaluating a software project is greater understanding of the software project organization, software development organization and people that are involved. This understanding is essential for the ability to perform software process improvements.

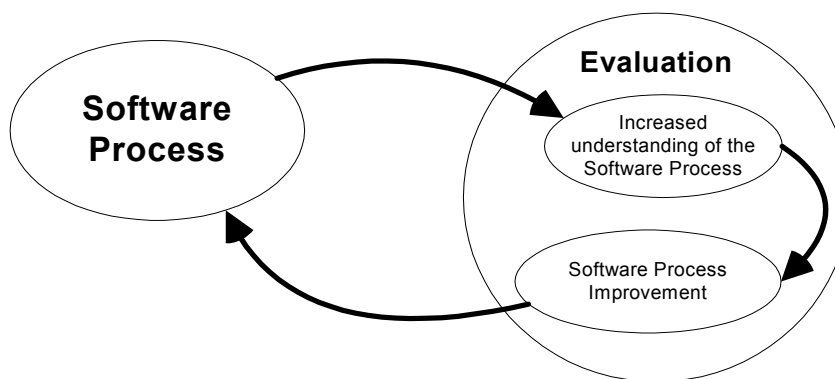


Figure 2-2. The bond between the software process and the evaluation process.

In figure 2-2 above, the bond between the software process and the evaluation is illustrated. This is a rather simplified illustration; the actual evaluation should be seen as a part in the software process.

As shown in figure 2-1 it is not enough to understand the process and then to improve it; goals need to be set. These goals have to be defined before the evaluation; otherwise, the collection of information can get out of hand and become of unreasonable size and cost [Sakamoto]. Furthermore, unnecessary information will be collected if goals are not defined.

For a software project evaluation, the main goal is set to a SPI model, for example the CMM. A small organization can then define subgoals from the SPI model and strive to reach those subgoals in their own software process. The focus of the evaluation is to find subgoals that are to be used for improvement in the software process.

2.1.3 Evaluation overview

To summarize the evaluation process:

- Find appropriate SPI model to use
- Use this model as the goal for the organization by implementing parts that are critical in a prioritized order
- Evaluate software projects
- Map improvement activities (parts from the SPI model) from the evaluation
- Perform improvements

Chapter 3

Software Process Improvement

Defining the software process is a first step in improving the software process. During a start-up of a business it is unusual that a software process is defined. When defining the software process the scope is set to documenting and understanding all parts of it. Although there is already a software process used, it is necessary to capture and understand it.

3.1 Process defined

There is a difficulty in explaining what a process is. This difficulty is described by [Zahran] where several definitions can be found but the main focus is set to three aspects.

The first aspect is that a process has to be defined. This incorporates some kind of document, either paper or electronically which specifies the activities and procedures used by the process.

The second aspect is the process learning that incorporates the knowledge being passed to the ones who are performing the process.

Finally, the third aspect is the result distinct by the products that are made when using the process activities. Figure 3-1 shows the aspects of the process.

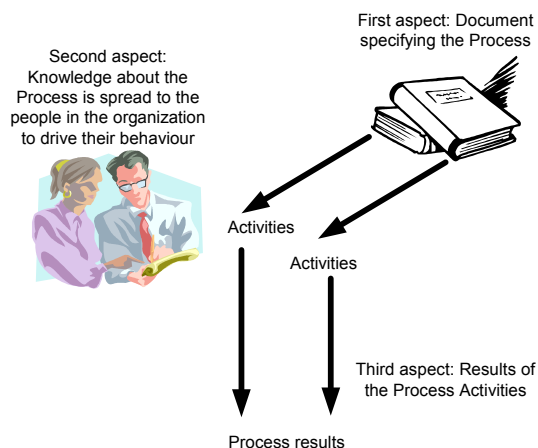


Figure 3-1. The aspects of the Process [Zahran]

Furthermore can be said; behaviours, activities and tasks that are performed to achieve a certain goal represent the process for achieving that goal [Zahran].

3.2 Process versus product oriented focus

Product oriented focus is concerned with tangible and concrete things, such as a final product. The process describes ways of doing things and this description has a result, the final product. The process is not necessary concerned with only one product, it could very well be for several products while the product oriented focus is set to one final product. This is very much seen in

companies that do consultancy. Different products could be developed by in housing, but still one process is followed to achieve quality.

3.3 Elements of quality

Quality is the major aim for any corporate level management and also many software managers and engineers try to strive for it. Both the CMM and SPICE are concerned with quality aspects, and so are many of the models developed during the past decades that are concerned with software process improvement activities.

We have to divide quality into two fields, one that addresses the product and one for the process. The products quality is concerned with aspects to the product as for the process quality it is concerned with the process and how it is able to generate a product with quality. In the quality of the process, we also have to define the making of the product as being of some quality, thus not entirely focusing on the products quality. A final product with high quality does not necessary state that the making of the product is of high quality. For example, a project that overruns time schedule with a result of over budget is not of high quality, perhaps very low quality, although the final product is of some quality.

An assumption that the quality of the product will be satisfactory cannot be made if software process quality is not present. On the other hand, if we have quality in our process then it is possible to make the result of the process, the product, with quality aspects concerned to the product. Another approach for defining quality for the process, would be the one given by [Ohara]: “The process is of high quality if the resulting product is perceived to be of high quality”

3.3.1 Defining quality

Quality can be said to have three factors [Dunn]:

- People
- Technology
- Management

The first factor is concerned with qualified personnel. The qualification includes attributes such as wise judgement, aptitude, education, training and attitude [O’Hara]. If the organization does not employ qualified personnel, it will be hard to aim for the final goal, the high quality product.

The interest for technology has to be present. This interest leads to a deeper understanding of the technologies that are used during the software development and to a higher knowledge in techniques that can be used. These techniques include tools for removing defects, programming languages, development methods, etc. Since the technology in software is in rapid improvement and change, I would say it is even more important to be able to adopt and evaluate new techniques all the time.

The last factor, management, especially software management, is concerned with cost estimations, business planning and resource allocation according to plans and the tracking of those plans with steps taken when tracking shows lack of being on course.

3.3.2 Software quality assurance

By introducing a software quality model to an organization, we need to be able to manage the model. This is attained by software quality assurance. It is important to understand that software quality assurance does not assure the quality of the software, but the effectiveness of the software quality model.

Software quality engineers usually inside the organization perform the software quality assurance. This is the cost in human resources and small organizations can find it hard to ensure funding for the activities of these resources. One way of implementing software quality assurance is by dividing the different activities over roles already found inside the organization, such as project manager, test leaders, head designers and so on. The other approach is simply devoting a role of a software quality engineer to the organization as a distinct role.

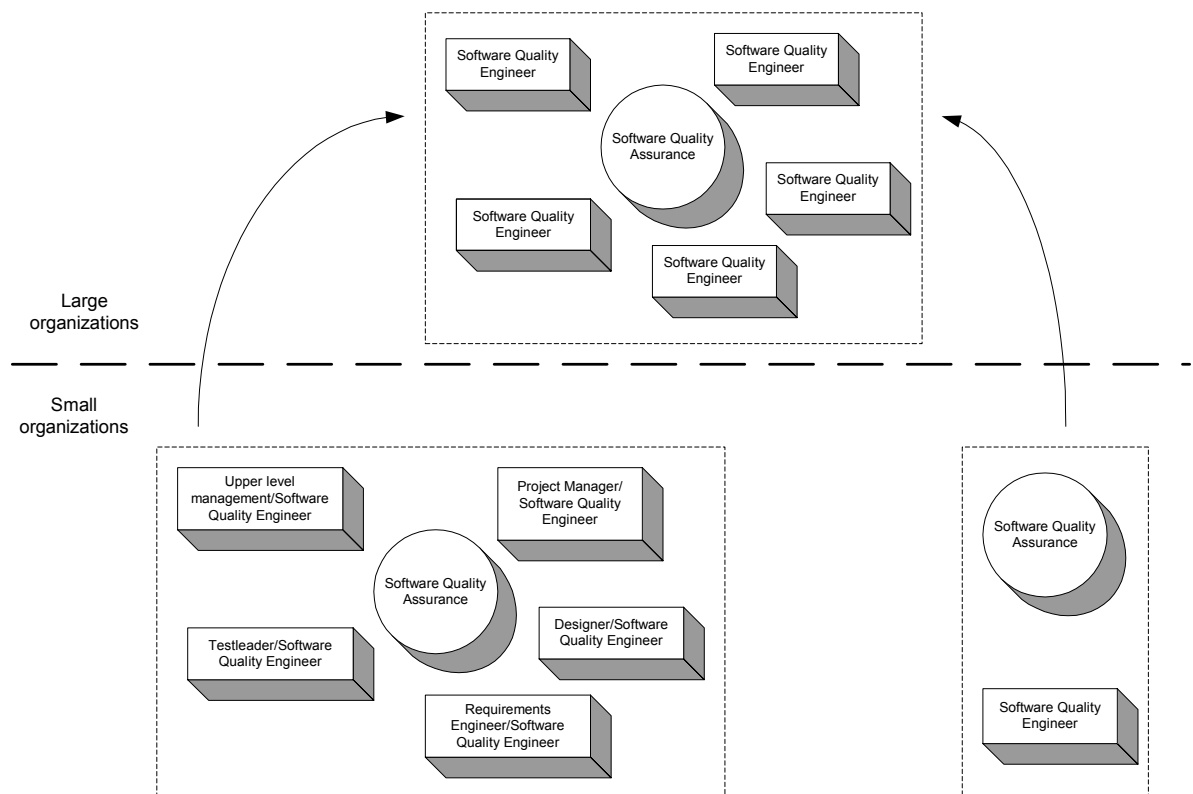


Figure 3-2. The software quality assurance resource allocation in a large and small organization

The literature often draws example from large organizations when defining or describing software quality assurance and its organization and roles. An easier approach for a small organization to begin with would be the formations described in figure 3-2. When an organization grows the software quality assurance may also grow in the direction where the large organizations SQA organization is defined.

The roles and responsibilities of the software quality engineers, called the evaluation moderators, are described in chapter 6 Organization.

3.3.3 Software Quality Attributes

When describing software quality a definition of the attributes that are appropriate for software products is needed. There are four attribute-domains [Dunn], which should be defined. These are usually the ones most interested by the customer:

- Reliability
- Usability
- Maintainability
- Adaptability

These four attribute-domains can be divided into attributes that are more commonly understood by the software society (see table 3-1 below).

Attribute-domain	Attributes
Reliability	Completeness
	Consistency and precision
	Robustness
	Simplicity
	Traceability
Usability	Accuracy
	Clarity and accuracy of documentation
	Conformity to operational environment
	Completeness
	Efficiency
Maintainability	Testability
	Accuracy and clarity of documentation
	Modularity
	Readability
	Simplicity
Adaptability	Modifiability
	Expandability
	Portability

Table 3-1 Software Quality Attributes

The attribute-domain and attributes are also named as the factor and criteria [Fenton]. There are many models of software quality. Among them the Boehm model and the McCall model are one of the earliest. See appendix A [Dunn].

3.3.4 Quantification of Software Quality Attributes

The ability of measuring quality in a software system is accomplished by quantification of attributes. When quantifying each attribute, it becomes more obvious what to measure and how.

For example the usability with the attribute testability and metric degree of testing can be measured by statement coverage, branch coverage or test plan completeness. By these metrics the quality can be administrated and compared with other similar results, the result from either an earlier version of the same system or by a similar system.

3.4 Measuring the improvement

Measuring a software process performance implies measuring against process goals. These goals should reflect [Zahran]

- alignment between the process and the business goals
- parameters for managing the process
- key indicators for measuring the process performance
- should be SMART (Specific Measurable Attainable Relevant Traceable)

Measuring an improvement on a process should not be performed on the former process versus the present process. Since goals for the present process will change over time the measures on the former process is not relevant. Therefore, all measures of an improvement should be against the goals that are set up and not according to the former process.

3.5 Why goals are important

During the work with the evaluation of a project, data is collected that is to be processed. This processing of the data collected, leads to the software process improvements that are to be made. During a project a large amount of data can be collected, but all data is not necessary to come to an understanding of which improvements that are to be made. It is therefore important, before starting the data collection to define data domains. For example, an organization that is at level 1 (for example in the CMM), is not interested in collecting data that would be appropriate for level 3. It would be more appropriate to define data domains and collect data according to level 2, since level 3 is a sequence of level 2.

The point made, is that defining data domains before starting to collect data is important, because otherwise too much data is collected and this takes more time. More time in both collecting and analyzing irrelevant data is unaffordable for the small organization. For large organizations, that usually try to implement a whole SPI-model at once, the work-effort grows more than needed [Sakamoto].

3.6 Defining goals for evaluations

Defining goals for an evaluation depends on several aims of the organization. One view can be that earlier projects have failed in some way and the problems can be defined. Another view can be the one where projects are not failing but the goal is to increase the efficiency. Small organizations are growing and to keep up, the evaluation can show how to grow in a controlled way.

The common goals are on time and inside budget [Wisén]. These major goals need to be better defined and divided so that the organization can find means of measuring them.

3.6.1 The interest groups

There are three possible interest groups in small organizations. These are the top-level management, project management and the people working within the projects. All of these have different interests and those need to be taken into account. The connection between the different interest groups is shown in figure 3-3.

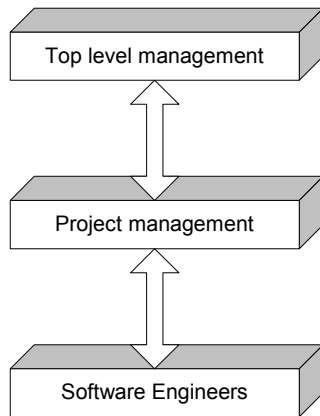


Figure 3-3. Three groups of interest groups in the evaluation process

3.7 CMM

3.7.1 Short introduction to the CMM

The Capability Maturity Model, called CMM, was released first time August 1991. Since then a version 1.1 has been released 1993.

In the CMM a framework is defined, that describes the key elements of an effective software process. There is a path in the CMM, from an ad hoc, immature organization to a mature and disciplined one. The path consists of five levels where the first level is defined as chaos. Each level affects different parts of the organization. See figure 3-7.

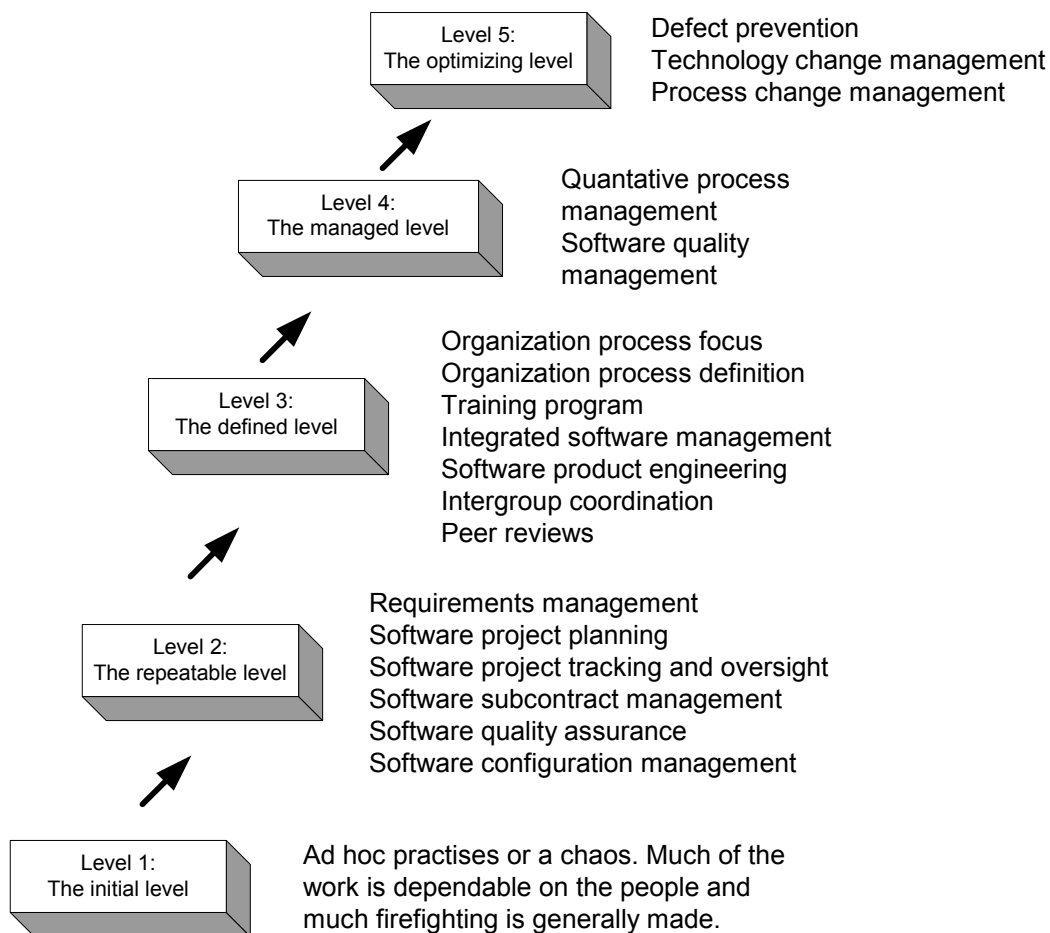


Figure 3-7. The Capability Maturity Models five levels

When moving from a level in the model the process is supposed to change, mature. Moving from level 1 to 2 the process is disciplined, moving from level 2 to 3 creates a standard consistent process, 3 to 4 generates a predictable process and the final level will give a continually improving process.

The CMM is constructed so that each level (except for level 1) can indicate the process capability of the organization. The process capability describes the range of expected results that can be achieved when following a software process. Each level (except for level 1) has key process-areas. These key process areas make achievement of goals for that level and are organized by common features. The common features are then organized into key practices that describe either the activities to undertake or the infrastructure.

It is recommended to follow each level in sequence and not skip any level until it is accomplished. There are possibilities of using key practices or key process areas from a higher level.

3.7.2 Benefits from CMM

Today several studies show the benefits and the problems of introducing the CMM into an organization. The benefits are clearly stated as higher quality but the cost is not so certain. [Herbsleb] indicate in their article that those engaged with software process assessments report an

increase in staff morale, ability to meet budgets, ability to meet schedules, productivity and product quality when changing from level 1 to level 2 or level 2 to level 3. There is one difference and that is the customer satisfaction, which is not increased when changing from level 1 to level 2. It might be the case that requirements engineering are introduced in a new way or that focus is set too much on software process improvement so that the customer cannot see the immediate profit.

3.7.3 CMM obstacles

A major obstacle with the CMM is the size and complexity. It embraces a lot of requirements and careful planning is needed. Also the cost to implement the entire model is high and many small organizations cannot find the funding for this.

As mentioned earlier, the cost of implementing parts of the CMM into an organization can be bearable and a faster break-even can be achieved.

Chapter 4

A Field Survey for a Model

4.1 The Field Survey execution

The field survey was conducted through interviews with Swedish companies that have departments of software development or similar. Companies that practice some kind of evaluation were included in the survey, just as companies that have no model for evaluation.

The companies that do not have a model for evaluation did perform some kind of evaluation but in a more none-formal way.

4.2 The Interviews result

When interviewing representatives of the selected companies, questions were asked about their administration of software project evaluation. During these discussions there were ideas of how the performance of the evaluation could be administrated differently, and observations made during the process of evaluation. Many of the models used had many similarities with the IEM described in the next chapter.

The interviews result is presented in the table below.

Question	Do you evaluate?	Pros	Cons	Observations during implementation of process (evaluation or other process)
Company 1	Yes. Own model based on CMM, SPICE, ISO9000, USK. The whole company is involved.	<ul style="list-style-type: none"> • The input (good/bad) gives a direction on what can be improved. • Teambuilding 	<ul style="list-style-type: none"> • When gathering the project group for evaluation there has to be openness. Otherwise the outcome will not generate proper feedback that can be used for improvement. 	(evaluation process) <ul style="list-style-type: none"> • Seen as a positive matter • A fear for a negative effect on development • Management faced with dilemma
Company 2	Yes. Own model.	<ul style="list-style-type: none"> • Indication of the suitability of processes used • Project members are able to reflect over projects and processes used • Measures of improvement can be made 	<ul style="list-style-type: none"> • Time can be hard to set aside 	(evaluation process) <ul style="list-style-type: none"> • Seen with positive attitude • Tracking of the follow-up can be hard, hard to create good enough feedback • Generally hard to get a sense feeling of where the problem is

		<ul style="list-style-type: none"> • As an assessment leader input is generated of how other projects have been performed 		<ul style="list-style-type: none"> • Important to establish routines, a natural way for people to follow
Company 3	Own model called the TEAM-method. Indirect evaluation, not documented.	<ul style="list-style-type: none"> • Matures the organization • Demands better requirements specification from customer 	<ul style="list-style-type: none"> • Customer might not see advantages 	<p>(TEAM-model)</p> <ul style="list-style-type: none"> • Organization not mature • People used to certain routines, hard to change • More administration
Company 4	Yes. Own model.	<ul style="list-style-type: none"> • Good feedback • Information stays within the organization • Gives the best education/knowledge from the evaluation 	<ul style="list-style-type: none"> • Time space between performing evaluations can be too long • Risk of not planning for evaluations, time is not enough 	<p>(evaluation process)</p> <ul style="list-style-type: none"> • Too extensive process, hard to implement • Important that everybody participate • A defined software process is needed
Company 5	No. Uses a Software Process (CMM)	<p>(assumptions)</p> <ul style="list-style-type: none"> • Avoiding recurring mistakes 	<p>(assumption)</p> <ul style="list-style-type: none"> • People can have trouble handling everything 	<p>Model:</p> <ul style="list-style-type: none"> • should not be difficult to manage • minimal work effort when implementing • give personal value • Can be hard to persuade top level management • Needs several projects to ensure metrics • Work effort can be seen as tiresome

Table 4-1 Interview survey results

The companies that are in the survey are described further in table 4-2.

Company	# of empl	Business segment	Interview persons position
Company 1	>10	Software product development	Quality Manager
Company 2	>100	Software and hardware	Processes Manager
Company 3	>100	Software and hardware	Manager
Company 4	<10	IT Consulting	Quality Manager
Company 5	>100	Software and hardware	Software Engineer

Table 4-2 Company data

4.3 Analysis

The grouping of the interview answers is sorted the following eight groups:

Input/Feedback	Process	People	Measures
Time	Customer	Organization	Management

After the grouping an analysis can be made on each group. From the analysis there are checkpoints produced that will be in the checklist of requirements that the evaluation model shall fulfil.

The groups are grouped with pros, cons and observations together.

4.3.1 Input/Feedback

As an assessment leader input is generated of how other projects have been performed	Gives the best education/knowledge from the evaluation	Good feedback
The input (good/bad) gives a direction on what can be improved.	Generally hard to get a sense feeling of where the problem is	Tracking of the follow-up can be hard, hard to create good enough feedback

This group refers to the topic of input and feedback from the evaluation process.

The evaluation experience gives that a person within the quality department can gain knowledge about how other projects are going. Of course one of many responsibilities for such a person would be to collect this information ongoing, but the answer indicates a smoother insight into other projects when using an evaluation model.

A point made is that this is the best way of educating personnel and giving them knowledge about the organization and the processes used.

Generally the evaluation gives a good feedback. The meaning of good feedback, is that the feedback can be used to understand problems within a process and measures can be taken that will improve it.

The evaluation input from personnel is divided into good or bad.

Here a problem is described as to where the actual problem is. It is important that an evaluation model can point to where the problem is so that proper measures are made.

Tracking the follow-up and creating sufficient feedback can be hard.

Conclusion

A model for evaluation should:

- ✓ Show where the problem occurs, by using a simple scale consisting of values like good/bad when performing the evaluation.
- ✓ Information from the evaluation should be shared with other personal including quality staff, in educational and information purpose.
- ✓ Tracking of the follow-up should be possible and sufficient feedback needs to be made.

4.3.2 Process

Minimal work effort when implementing	Model should not be difficult to manage
Too extensive process, hard to implement	More administration

This group refers to the process of the evaluation.

Minimal work effort when implementing indicates that the process of the evaluation should be easy to implement. By easy the interview-answers give that the work effort of implementing the process should be minimal.

The model should not be difficult to manage is a further criterion.

If the model of evaluation will be too extensive, it will be hard to implement and this gives that it will not be implemented fully.

More overhead and administration is a fear for any organization, since it ties up resources and time, thus increasing financial cost.

Conclusion

A model for evaluation should:

- ✓ Not be too extensive or resource and time consuming/demanding, thus the overhead and administration should not increase noticeably.
- ✓ The implementation of the model should be made with minimal work effort for the organization.

4.3.3 People

(1) Work effort can be seen as tiresome	(2) Important that everybody participate	(3) Seen as a positive matter	(4) People can have trouble handling everything
---	--	-------------------------------	---

(5) Positive attitude	(6) Teambuilding	(7) People used to certain routines, hard to change	(8) Project members are able to reflect over projects and processes used
(9) Model should give personal value	(10) Important to establish routines, a natural way for people to follow	(11) When gathering the project group for evaluation there has to be openness. Otherwise the outcome will not generate proper feedback that can be used for improvement.	

The collected data about the people topic is concerned much with how to motivate project members to welcome and use the evaluation model. Although (3) and (5) gives that evaluation models that are used and evaluations performed, are seen positive by project members there are matters that needs to be taken under consideration.

Problems that need to be avoided are (1) and (7). The matter concerned is the work routines and the model needs to handle changing routines for people.

This can be made by incorporating the (2), (6), (9) and (10) into the model.

Conclusion

A model for evaluation should:

- ✓ Make everyone participate into the evaluation work.
- ✓ Give personal value back.
- ✓ Establish routines which are natural to follow.
- ✓ The person responsible for the evaluation needs to create openness when performing the evaluation.

4.3.4 Measures

Measures of improvement can be made	Needs several projects to ensure metrics
-------------------------------------	--

Out of the answers for the group of measures it can be determined that the evaluation needs several projects to collect data from.

Conclusion

The model for evaluation should:

- ✓ Support iterative collection of data over time

4.3.5 Time

Time space between performing evaluations can be too long	Risk of not planning for evaluations, time is not enough	Time can be hard to set aside
---	--	-------------------------------

When analyzing the top collected data about time, a fear of time not being enough can be seen.

The model for evaluation should be able to handle time routines such as frequent evaluations, time for planning of evaluations.

Conclusion

A model for evaluation should:

- ✓ Make it able to perform frequent evaluations
- ✓ Support for time planning

4.3.6 Customer

Customer might not see advantages	Demands better requirements specification from customer
-----------------------------------	---

Whenever an organization is changing this have implications not only for the people inside the organization but also outside. Thus the customer/client is involved. If a product is delivered to a customer with enough customer satisfaction then the organizational change might be seen as not feasible. The other situation that might occur, when a product is not delivered with enough customer satisfaction, the understanding for an organizational change is more accepted and understood.

To be able to handle the case of outside response (whether good or bad) it is important to incorporate customer involvement in the model.

Conclusion:

A model for evaluation should:

- ✓ Involve the customer into the process

4.3.7 Organization

Information stays within the organization	Matures the organisation	Indication of the suitability of processes used
Organization not mature	A defined software process is needed	

The topic of Organization gives that the maturity of the organization grows, but there can be a problem if the initial maturity is not sufficient. The solution is to have a software process defined before being able to perform an evaluation. By defining the software process, the indication of suitability of the processes used can be determined thus improvement is possible.

Another point made, from the collected data of the interviews, is that the information stays within the organization since the process is defined.

Conclusion:

A model for evaluation should

- ✓ Before implementing any model for evaluation, the process has to be defined.

4.3.8 Management

Can be hard to persuade top level management	A fear for a negative effect on development	Management faced with dilemma
--	---	-------------------------------

Top level management has to be convinced that the improvement made or process/model introduced will be profitable to the organization. It is also important the initialization of the model is made by the top level management.

Another problem that may occur is that the top level management is faced with dilemma of what improvement should be made. Thus the model should give very direct information, with attributes for measuring the profit on the improvement made.

Conclusion:

A model for evaluation should:

- ✓ Be initiated by the top level management, thus supporting the person responsible
- ✓ Have adequate information about what output is needed from the model to ensure the right decision when making an improvement to the process.

4.4 Summary

What requirements can be demanded of a model for evaluation based on the interviews performed? Here is a summary of the requirements:

- ✓ Show where the problem occurs, by using a simple scale consisting of values like good/bad when performing the evaluation.
- ✓ Information from the evaluation should be shared with other personal including quality staff, in educational and information purpose.
- ✓ Tracking of the follow-up should be possible and sufficient feedback needs to be made.
- ✓ Not be too extensive or resource and time consuming/demanding, thus the overhead and administration should not increase noticeably.
- ✓ The implementation of the model should be made with minimal work effort for the organization.
- ✓ Make everyone participate into the evaluation work.
- ✓ Give personal value back.
- ✓ Establish routines which are natural to follow.
- ✓ The person responsible for the evaluation needs to create openness when performing the evaluation.

- ✓ Support iterative collection of data over time
- ✓ Make it able to perform frequent evaluations
- ✓ Support for time planning
- ✓ Involve the customer into the process

The prerequisite for introducing an evaluation model are:

- ✓ Before implementing any model for evaluation, the process has to be defined.
- ✓ Be initiated by the top level management, thus supporting the person responsible
- ✓ Have adequate information about what output is needed from the model to ensure the right decision when making an improvement to the process.

This checklist can now be used to evaluate either an already used evaluation model within the organization or establishing a new model. In this master thesis the checklist is used to compare the model recommended (chapter 5) against the Experience Factory model.

Chapter 5

The Iterating Evaluation Model

5.1 The software engineering model

To understand how an evaluation process can be implemented into the software project process, a definition of the software engineering model is presented below in figure 5-1.

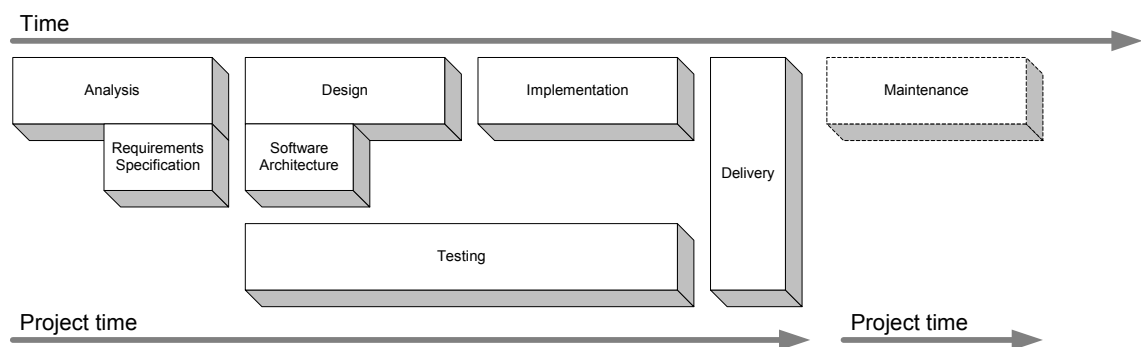


Figure 5-1. The Software Engineering Model

In the time scale there is a project defined with the activities of [Gilb] [Nicholas]

- Analysis
- Requirements specification
- Design
- Software architecture
- Testing
- Implementation
- Delivery

The maintenance activity is separated since it is defined as a project itself with activities similar from the main project.

This is a common way of performing a software project [Larman] [Eriksson]. The model is called the waterfall-model. Other models such as evolutionary, spiral or prototype also exist.

Variations exist such as testing starts during the analysis where the requirements specification is tested. A different approach is to start testing during the end of the design-phase. The analysis phase is sometimes performed as a pre-study project.

The analysis phase is concerned with what has to be made and not how. The outcome of the analysis phase is the requirement specification. The specification is a live document and can be

changed during the life cycle of the project. The contract and the system development plans are based on the specification [Kotonoya].

In the model there is a border between the analysis and the design activity. The border between where the analysis finishes and the design begin is a floating border and not as distinct as illustrated in the figure 5-1. The design activity has several parts such as the design of database schema, software architecture, system design and more. Together with the analysis it defines and forms the actual system.

An important assistance to the work with the requirements engineering is the software architecture which simplifies the communications with the stake holder [Bass].

Testing the system is an important activity and also extends the requirements specification [Robertson], the design and the final implementation. The major tests used are unit testing, integration testing and system testing.

Finally the system is implemented by using tools such as compilers, change management systems, automated test tools, etc.

The common part in every activity is the requirements specification and the correctness of it. This can be hard to handle for a large or complex software project, which is why it is a live document throughout the life cycle of the project although only minor changes may occur.

5.2 The evaluation model

An evaluation should be seen as an activity in the SE-model¹. The figure 5-2 below shows how the evaluation is placed into the model.

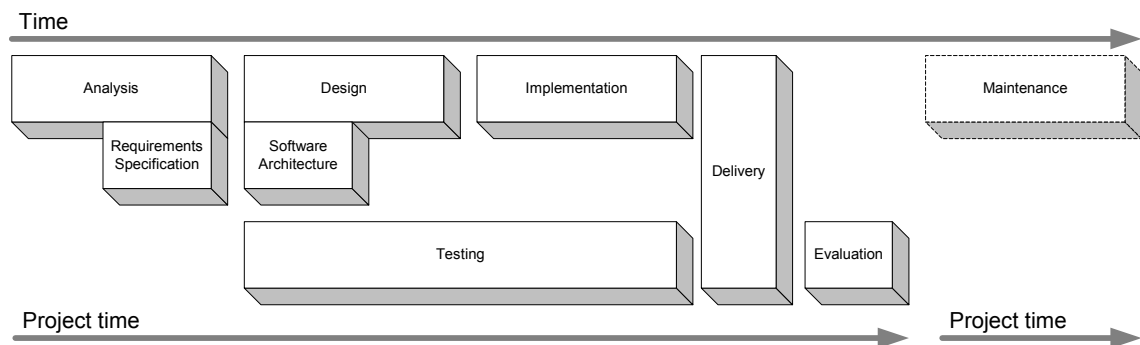


Figure 5-2. The evaluation activity in the SE-model

By placing the evaluation as an activity into the SE-model, it can be treated as a natural part of the project. The process of development will then include the evaluation process as a part of it.

There are two phases in the evaluation, the pre-study and the evaluation. The pre-study is concerned with collecting data from the interest groups and collecting project-data.

¹ Software Engineering model described in chapter 5.1

Having short informal sessions where problems are discussed openly enables collecting data from the interest group. These sessions are only open for a specific interest group since the interest of each group differs from the other ones and enables a natural openness. For example, a group of developers will most probably discuss topics concerning practical issues like design or tool problems. On the other hand, a group of management would probably be more interested in economical issues and how to cut overtime budget issues. All issues are of course connected in some way, but by using homogenous groups, specific topics can be easily found.

The collection of the project-specific data is made from the test-phase of the project and by an economical overview for the entire project. This is illustrated in figure 5-3.

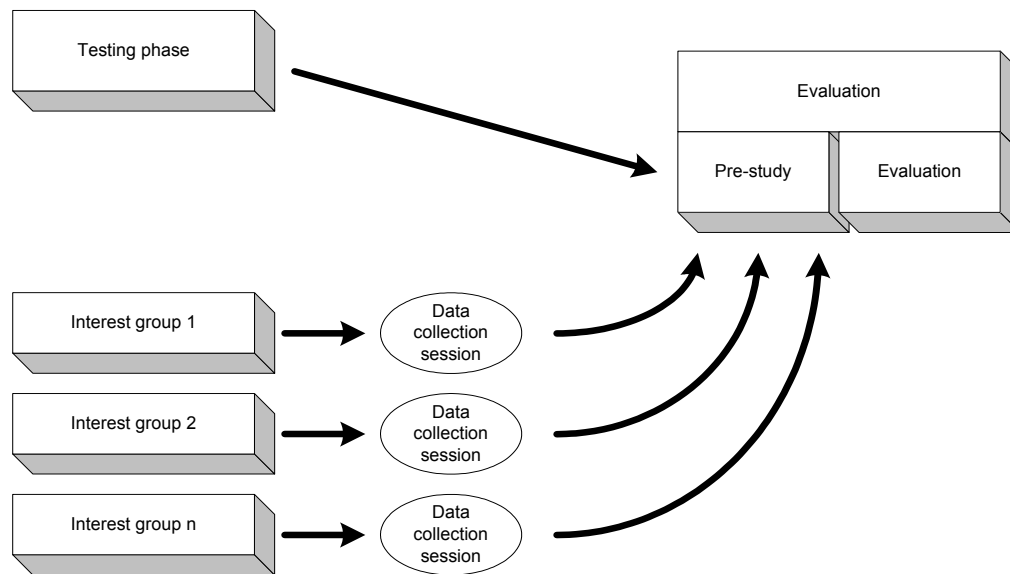


Figure 5-3. Data collection in the pre-study of the evaluation

When the data-collection is finished from each interest group, the actual evaluation of the project can be started. The process of the evaluation is built up by organizing the data collected into problem-domains that can be mapped to the SPI model. The outcome of the evaluation phase is an action plan for what improvements that should be made.

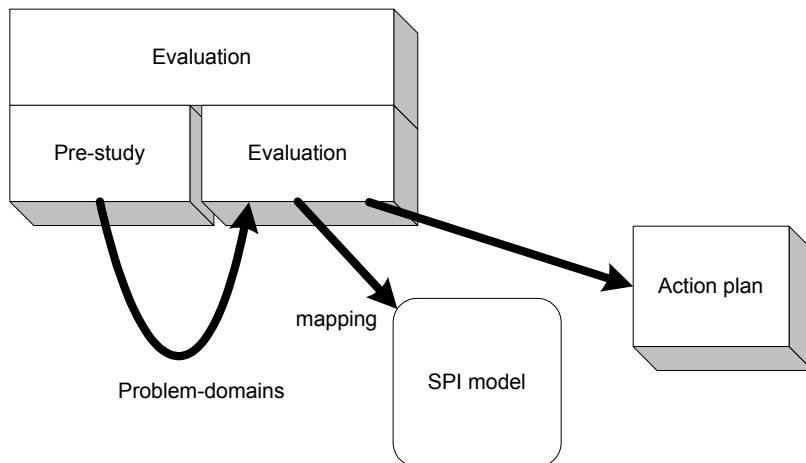


Figure 5-4. Goal of the evaluation phase, the action plan

5.2.1 The action plan

The purpose of the action plan is to show what improvements are to be made in the organization to succeed in implementing a SPI model. This is the outcome from the evaluation.

The creation of the action plan is divided into two separate parts, the action proposals and the action activities. From the evaluation come one or more action proposals concerning the activity that is to be carried through for enabling a specific implementation of an activity in the SPI-model. The activity is then divided into action items that are assigned to personnel in the organization.

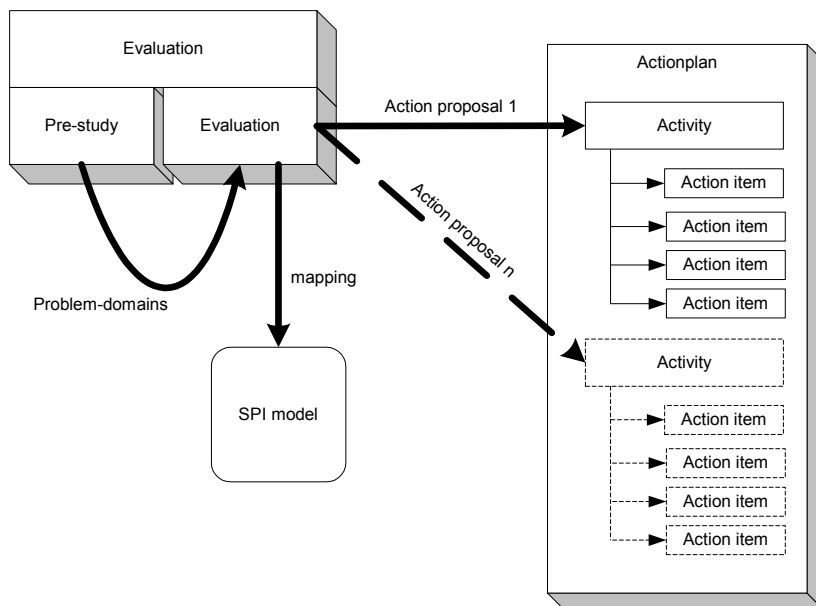


Figure 5-5. The action plan

The purpose of the action plan is to clearly state how a recommendation from the SPI-model shall be carried through. The activity itself, describes what to be done and the action item describes how to implement the activity, which was a former recommendation. Further work can be done, for example prioritizing each activity or forming work-packages that involve several activities; this is up to the evaluation moderator and not further investigated in this master thesis.

Chapter 6

Organization

6.1 Introduction

The organization of the evaluation is inherited with ideas from formal inspections, first documented by Michael E. Fagan [Fagan]. It was developed for IBM and the first release was in a technical report 1974.

6.1.1 Aspects of formal inspection — against

There are many different aspects argued for and against the formal inspection. Usually the main critics against, emphasize the process of inspection being large and therefore resource demanding. Other critics argued against are the time spent on inspections, psychological issues like the author feeling unpleasant (miss credited) when his/her work is being inspected and the actual defects found.

6.1.2 Resources

The resource allocation consists of an inspection moderator (original IBM-title, usually called inspection leader in recent documents published), secretary, several inspectors and the author.

As one can see, already we have an organization consisting of three distinctive roles, the inspection moderator, secretary, and inspectors. Minimum resources used would be four resources (one inspection moderator, one secretary and two inspectors).

For small organizations resources and time could be spent on formal inspections which the organization can find hard to fund. The formal inspection is better fitted for medium or large-sized organizations. Still benefits from inspections are present, not depending on the size of the organization (of course, an organization consisting of two people might have some problems conducting the inspection).

The most important factor is to conduct the inspection correctly and this is made available through training. Training is based on two factors; special courses in inspections and the knowledge earned from past inspections.

6.1.3 Aspects of formal inspections — for

Although the organization of the formal inspection is rather resource consuming, and therefore time consuming, there are benefits that are important to a software organization. One of these benefits is defect detection early in the project phase. Defect removing is less costly if made early in the projects lifecycle. A design defect is often easy to remove during the design phase, but is much more costly if found during the final period of the test-phase (since more phases are affected).

Also goodwill is gained if defects are found before the final shipping to customers and users.

6.1.4 Short introduction to formal inspection

The formal inspection is structured by a pre-inspection phase, an inspection-phase and a post-inspection phase. During the pre-inspection phase the inspection package/s is generated by assembling the document that is to be inspected, dependencies of other documents and a guideline for what to inspect and how. This is usually conducted by the inspection moderator together or accompanied by the author/s. The organization of the inspection is commonly involved with the following roles:

- Inspection leader
- Inspection moderator
- Inspector/s
- Secretary

Furthermore, the author is also involved. As seen in figure 6.1 below the inspection moderator is the person that holds the inspection meeting. Usually the inspection moderator and the leader can be seen as the same role.

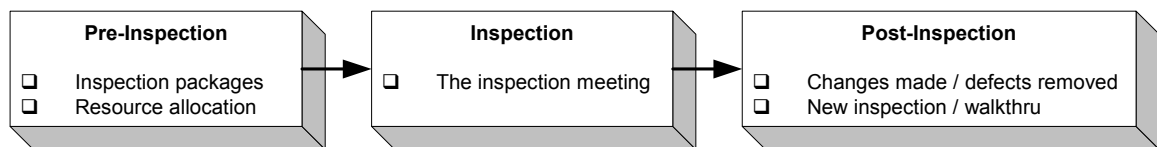


Figure 6-1. Overview of the organization of the formal inspection

6.2 The Organization of the Evaluation

6.2.1 Roles

The evaluation is composed of the following roles: the evaluation moderator, the interest groups and a secretary.

The evaluation moderator role is assigned to the project manager of the project or a specially assigned evaluation moderator in the organization. Since the evaluation process needs to be well understood by the evaluation moderator role should be assigned to one person. Customization of the evaluation-structure is the evaluation moderator's responsibility. This customization involves activities of creating protocol templates, acting as a channel of feedback to the different interest groups and making certain that the Improvement Action Plan is generated. He or she could be a senior developer or manager with an ability of objective thinking.

6.2.2 Time management

Time planning for the evaluation is based on two main phases, the pre-evaluation (or evaluation preparation) and the evaluation.

When doing the pre-evaluation phase the necessary activities are

- Finding interest groups by grouping people according to their common (business) interests inside the organization
- Creating a meetings plan involving each interest group and channel the information

One general idea of finding an interest group is by looking at the business organization and locating the most homogenous groups. For example, there is usually a group of top-level management, development and project management. These three groups could be enough for a small organization. There is no need creating very small groups since this will generate many common ideas and time will be spent on too much overhead.

When the different interest groups are informed about the meetings-schedule and which groups they belong to, the evaluation meeting is performed with the project in focus. Time usage should be set so that there can be discussion around different topics. A large project could very well have several interest groups meetings to ensure that project members do not forget details.

6.2.2.1 Activities

The evaluation phase consists of three activities:

- the assembling of meeting protocols
- the mapping-process
- the creation of the improvement action plan.

All meeting protocols from all interest groups shall be collected from the secretary and assembled to one document, the evaluation-document, called the E-protocol. The E-protocol is organized, when assembling, to groups that are in common for the project. For example tools functionality, customer relations, requirements specification etc. This enables the work of mapping the different groups to the SPI-model. Finally the result of the evaluation; the improvement action plan (IAP-document) can be created.

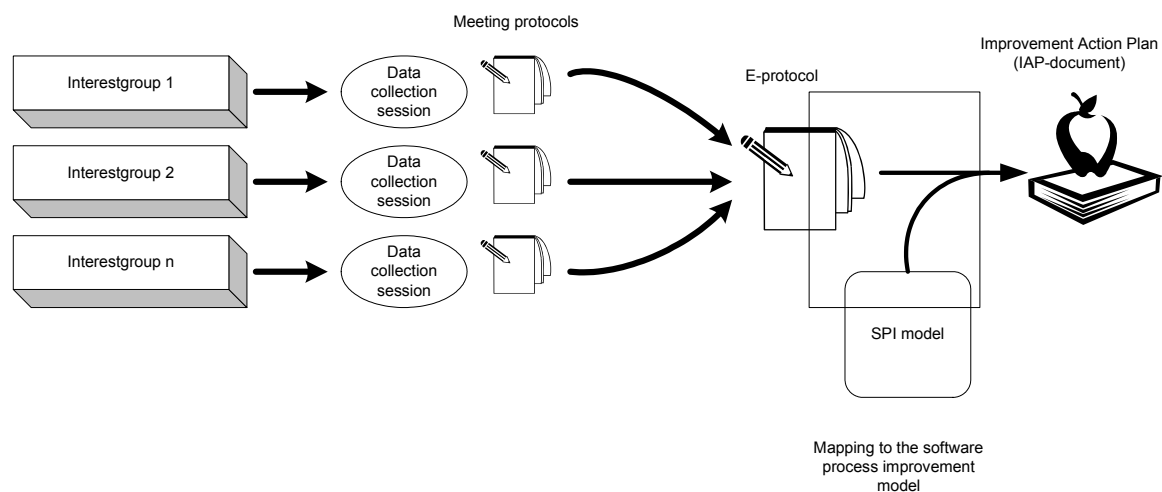


Figure 6-2. The process of creating the IAP-document

6.2.3 Complementary roles

During a software process improvement, S3 (Silicon & Software Systems) [O'Hara] [Zahran] used process mentors to support the introduction of new processes. This idea can be used when introducing the evaluation process into the project organization. A senior developer or manager could act as a support for the evaluation moderator. This mentor could be inside the organization or hired as an external consultant.

6.3 Training

An effective introduction of a process or SPI-model needs acquiring appropriate training. If the training is omitted an inconsequent process change is highly possible. Also process goals are to be missed or overtime the process is deteriorated [Zahran].

Before the training is performed the interest groups needs to be defined and goals to be set. Without goals, proper training cannot be planned. The obvious groups are those who are performing the evaluation and the earlier described interest groups.

Focus is set on training the SPI-model and the evaluation-process. First, the management needs to be oriented about the outcome of an SPI-model. This enhances future support to the responsible for the evaluation. Becoming well aware of the benefits of the chosen SPI-model also indicates that a dialog has occurred previously about weaknesses and strengths in the organization and what should be done. At this early stage external consultants should be used and comparison from other companies that have evaluated the same SPI-model be shared.

The responsible personnel of the evaluation-process have to acquire knowledge of the chosen SPI-model to be able to perform the second phase of the evaluation, the mapping of the E-protocol to the SPI-model. Different courses for different SPI-model can be offered by specialized companies which focus is to educate in this area. Often organizations with their own SPI-models have educational possibilities to offer.

Spreading information about the evaluation process to the entire organization enables personnel to gain knowledge about the different elements it consists of. A very important group is the newly recruited employees who join when the organization expands. It is important to make the evaluation process a part of the software process, so that it is not forgotten. Otherwise it will be neglected overtime [Humphrey].

6.4 The Change Process

The change process involves two phases, the change insertion and making the change permanent [Humphrey]. A resistance is always present in every organization and to succeed a process change one has to succeed in making the change permanent. This is not unique for software organizations but awareness of the technical character existence inside the organization is important. This character is concerned with technical solutions, and often these solutions are distinguished, mostly by the individuals, and seen as the best solution. A view like this can be an obstacle for making the process change permanent.

Management has an important role here; they are the ones who have to lead the change. Authority needs to be present for the change to be instituted. It is common to have the idea of improving by making a change. Although for the technical character a change has a factor of uncertainty which

makes the change less attractive. This could create a resistance towards the change. Succeeding in the change incorporates the idea of unfreezing and refreezing [Humphrey]. Unfreezing has the purpose of taking the individuals to a state where the change can be accepted and accomplished and refreezing makes the change permanent.

It is important to plan a change. The implementation of an evaluation process is no different. The planning needs to be thorough so that the unfreezing can be made. An agent [Humphrey] can be used. An agent is a person that understands the new process well and is enthusiastic about it. The agent should also be technically as politically skilled to understand the problem(s), and understanding the benefits that the process change will bring. Management has to support this agent fully and he or she needs to have the respect from fellow coworkers.

It is important to involve those performing the change to also be a part of the planning. This makes it possible for the responsible person to get feedback for the ongoing change process. It will also make the unfreezing easier.

Technical changes can be implemented sooner, changes concerned with human behavior are more difficult, takes longer time and precaution needs to be taken. A slower pace is recommended when performing the change, so that everyone in the organization will keep up with the change work. Information about the progress is essential, so that those concerned with the change can see the use and benefit of it and be updated how far the change is made.

The last step is the refreezing, which enables the change to be permanent. Here an important factor is to make those who initiated the change to remain as responsible after the change is performed. Let the new procedures and routines in the change be a part of the bureaucratic organization. If the change is complex it has to be tracked by specifically assigned people such as the quality engineer and an assistant.

The difficulties concerned with the unfreezing phase, both for large and small organizations, is heavy workload and lack of time. Another difficulty is that technical personnel are deeply involved in their work and might not be able to see the benefits from a change immediately. As mentioned before, management needs to understand the problems and goals for the change. It can still be difficult for the personnel to make the change in their day work since it is often neglected. Using evaluations and a process for that, the changes can be made by the personnel and solutions can be found when mapping them to a SPI-model.

6.5 Documentation

The documentation of importance which needs to be present when implementing and maintaining an evaluation are

- Process definition; a document that describes the evaluation process
- Template for the E-protocol; a document template for the meeting protocol made during the interest group meetings.
- Template for the IAP-document
- History checklist; a checklist of the parts from the SPI-model that have been implemented, with date, change date and metrics.

It is important that the process for the evaluation is defined (see chapter 2.1.2). The definition from this document or an alternative one more enhanced for the organization can be used. A process definition is the quality engineer's responsibility to make, notifying parts involved and performing training. The parts involved are the top-level management and those interest groups involved with project work. Also external consultants or newly recruited personnel need this notification and training in the evaluation process.

If the quality engineer is replaced or leaves the organization, continuous documenting enables this change to become more effective. Progress can be measured of how much of the SPI-model is implemented if the process and process related work is documented. Without documentation, metrics that are appropriate cannot be found and measuring the progress and quality of the process is impossible.

Chapter 7

Implementation

7.1 How to begin — pre-evaluation phase

7.1.1 Setting goals

A first step towards implementing the evaluation model is to generate understanding of the goals that the organization has. The goals are collected from a SPI-model. Goals cannot be defined if the SPI-model is not introduced and understood by those who will manage the evaluation process. This is an important step for the second part of the evaluation model; the mapping of the outcome from the interest groups to goals. As explained earlier the action plan is based on the actions that are needed to generate a certain goal for the organization. This action plan is important for further implementation of the SPI-model.

7.1.2 Training

Adequate training is required for a process to become permanent (refreezing). The practical approach would be, after the quality engineer is selected that he will in conjunction with top-level management choose a SPI-model. It is important that the collaboration between the quality engineer and the top-level management is good, since from this, mutual goals can be set up and understood.

After that the SPI-model is chosen, the quality engineer needs to gain deep understanding and knowledge of the SPI-model, by self-studies or external courses. This training is important to understand the different parts that the SPI-model has and how they depend and build up the model. Correct mapping of the SPI-model to the E-protocol can then be made.

The quality engineer provides further training. It incorporates training of the evaluation process at a high abstraction level of understanding, and deeper training in the evaluation process for those interest groups that are involved with different parts of it.

7.1.3 Documentation

First create the document that describes the evaluation process. It could be of a general nature, taken from this document, or a refined process definition for the specific organization. In this process description document, the documents that are to be used are defined and specified. From this specification, templates are made of those documents, including instructions on how and who is to use it.

Those documents described in this recommendation are the E-protocol and IAP-document. The E-protocol is based on meeting-protocols made from every interest group meeting. The IAP-document holds the action plan of which activities that are to be performed to reach some goal. It is important to document how to map the E-protocol to the SPI-model so that the IAP-document is finally made.

7.2 How to conduct the evaluation

The evaluation is performed after a projects completion. In a time perspective, the last activity within the projects lifecycle is the evaluation. Usually there is a short time gap between two projects combined with a kick out (to mark the end of the project). This gap should be used for evaluating the project.

The first evaluation differs from the coming evaluations in a way that no result from any prior evaluation exists. This result, the IAP-document, which is created during the final phase of the evaluation, is used in the next evaluation. The information in the IAP-document is introduced into the organization and for the next project the improvements should be implemented. This way a natural flow of improvements is inserted into the organization and during the test-phase or near the end of the project, data can be assembled to verify the improvements made.

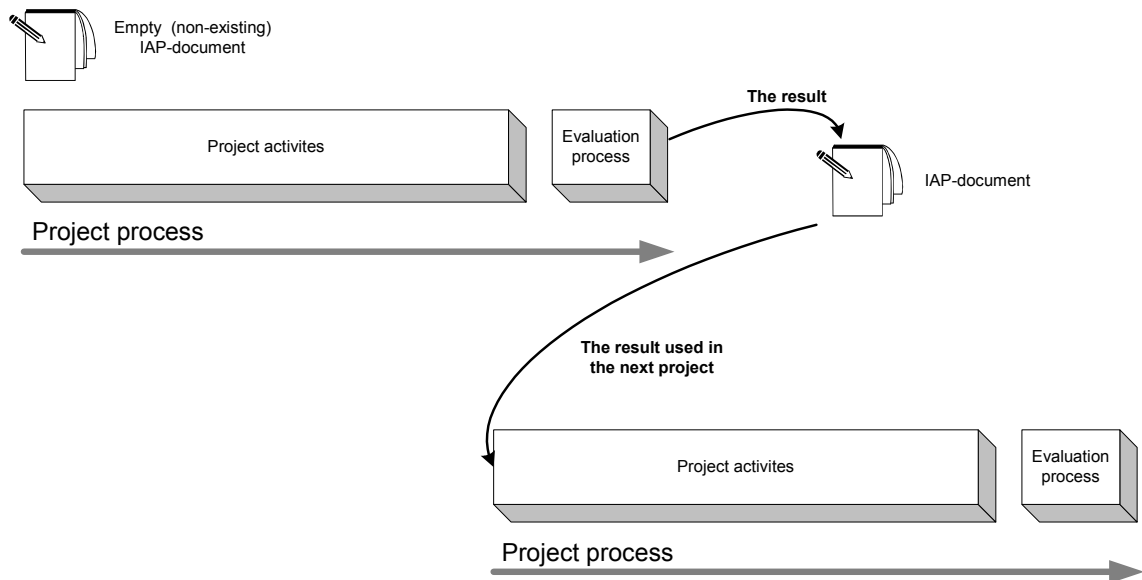


Figure 7-1. The iterating evaluation model

7.2.1 The Improvement Action Plan

The result from the evaluation is the IAP-document. This document has activities that need to be performed to improve the projects process. These activities are directly mapped to the SPI-model used.

7.2.2 Areas of evaluation

The areas to be discussed during the interest groups meetings can be such as [Briner]

- Project life cycle; when was the work effectively made, where did the problems occur?
- Success criteria's; where they correct, where they applicable?
- Project organization and funding

- What problems arose and how where they handled
- Factors of surprise — did any occur?
- Strong and weak points with stake holders
- The core group of the project and project manager — how did the organization act, how was the communication?

These areas can be used for every interest group since all of them have different views and opinions. It is important to have an open discussion within these groups and bring forward the essential thoughts and views.

7.3 The evaluation of the evaluation — post-evaluation

The evaluation process needs to be evaluated. The thought of a complete and optimal evaluation process that is only created once does not apply here (does it ever?). It is therefore important to evaluate it and make process improvements over time.

The quality engineer is responsible for this work. It is the quality engineer's task to track and monitor the quality of the evaluation process. Evaluating the progress of reaching goals that are setup by the organization and necessary changes can be made.

By the use of a project log the quality engineer together with the project manager can measure if the performed improvements have reached the goals with satisfaction. Activity records and standards plan should also be kept for the future to evaluate [Lockyer].

A final meeting when the evaluation is completed should be held to inform everyone of what the outcome of the evaluation is, so that everyone is informed of what changes are to be expected.

7.4 Support and top level management

In every organization it is important with support and understanding from top-level management. This is also the case here, top level management need to be well aware of the goals and requirements set for an evaluation process and generate this support for the people who are executing the process. As said before, the outcome of the evaluation process is the improvement action plan. It is important that personnel conducting the evaluation and generating the IAP-document make sure the improvements are made. If not, the risk of falling morale in the organization and authority for the management may be endangered. This is not unique for the evaluation process, but rather common in whatever changes that is to be made within the organization [Macheridis].

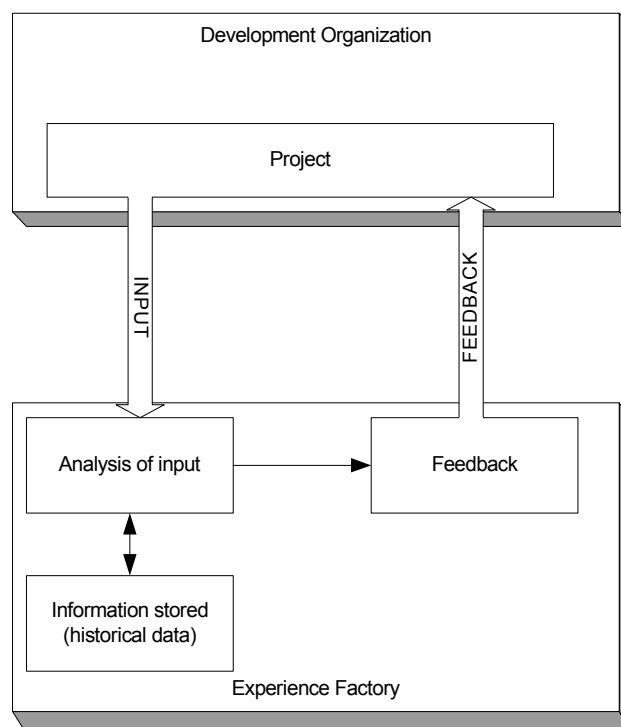
The support from the top-level management is not only concerned with funding, but also with knowledge and leading. Management needs to show the way of the evaluation process at an early stage. Furthermore, knowledge needs to be passed to those involved with the evaluation so that proper guidance is made available.

Chapter 8

Experience Factory

8.1 Introduction

The Experience Factory is a model for utilizing reuse of life cycle experience and products [Basili]. It is defined as both a logical and/or physical organization and the activities are independent of the development organization, thus it is implemented outside the development organization. The development organization provides information to the experience factory, which analyzes the information and returns feedback.



8.2 The Quality Improvement Paradigm

The Experience Factory uses the Quality Improvement Paradigm, developed by Basili, et al., as a basic methodological device. The QIP is the outcome of a scientific approach to the problem of software quality improvement and is related to the Shewart-Deming Cycle Plan/Do/Check/Act [Deming] which is used in the industry for the implementation of quality management plans.

The QIP paradigm is divided into six steps [Basili]:

Characterize: Understand the environment based upon available models, data, intuition, etc. Establish baselines with the existing business processes in the organization and characterize their criticality.

Set Goals: On the basis of the initial characterization and of the capabilities that have a strategic relevance to the organization, set quantifiable goals for successful project and organization

performance and improvement. The reasonable expectations are based upon the baseline provided by the characterization step.

Choose Process: On the basis of the characterization of the environment and of the goals that have been set, choose the appropriate processes for improvement, and supporting methods and tolls, making sure that they are consistent with the goals that have been set.

Execute: Perform the processes constructing the products and providing project feedback based upon the data on goal achievement that are being collected.

Analyze: At the end of each specific project, analyze the data and the information gathered to evaluate the current practices, determine problems, record findings, and make recommendations for future project improvements.

Package: Consolidate the experience gained in the form of new, or updated and refined, models and other forms of structured knowledge gained from this and prior projects, and store it in an experience base so it is available for the future projects.

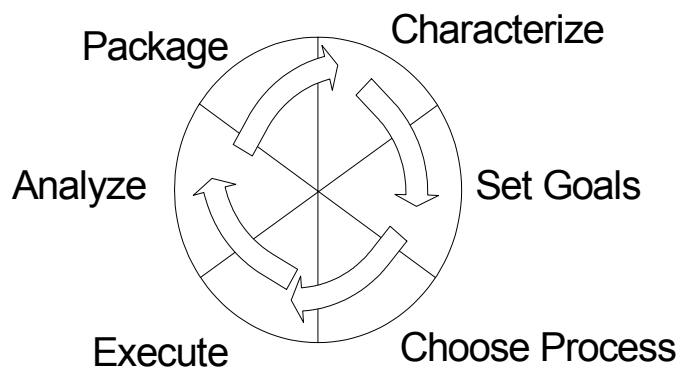


Figure 8-1 The Quality Improvement Paradigm, QIP

There are two types of feedback cycles, the project feedback cycle and the corporate feedback cycle. The project feedback cycle also called the control cycle provides feedback to the project during the execution phase. Feedback from the corporate feedback cycle is the feedback that is provided to the organization and has the purpose of providing analytical information about the project performance when the project is completed such as the deviation from other projects or a nominal range.

Another purpose of the corporate feedback cycle is to accumulate experience as software artifacts that can be used for similar projects and improving them.

The characterization is about finding the right characters for the project and making it able to isolate a class of projects with similar characteristics and goals compared to the project at focus. The aim point of the characterization is to provide a context for goal definition, reuse of experience and products, process selection, evaluation and comparison.

After the characterization is made, goals are evaluated and defined. The technique used in the QIP is the Goal/Question/Metric Paradigm.

When the characterization is made and the goals are defined a process is chosen. The process is a generic life cycle model with a set of methods and techniques. This generic life cycle method is then tailored towards the characters and goals of the project. In the QIP there is a distinction between technique, method and life cycle model. The technique is defined as a basic algorithm or set of steps to be followed in constructing or assessing the software. A method is an organized approach based upon a technique and defines guidelines such as how and when to apply the technique such as entry and exit points. The life cycle model is a collection of methods that are used through the entire life cycle of the software product.

To be able to execute the processes, experience should be accessible in a packaged form. The package has information about processes that have been chosen, prior products available for reuse, resource and data models and software development models that can handle the reusable packages. An important point made in the QIP is that the data collection should be a part of the process.

8.3 The Experience Factory

The QIP is based on the notion that improving the software process and product is based on collecting and accumulating of evaluated experience, learning. This is captured in a form that can be understood easily and modified as experience models. These experience models are collected in a repository of integrated experience models, the experience base. This experience base can then be accessed and modified by the current project as a reuse of experience.

The Experience Factory, EF, is a logical organization that may also be physical. The separation between the project organization and the EFO is natural since the QIP implies a logical separation between the project development and the systematic learning and packaging of reusable experiences.

Support to the project development from the EF, is made by analyzing and synthesizing all kinds of experience. The EF acts as storage for such experience, and supplying the project development with relevant experience.

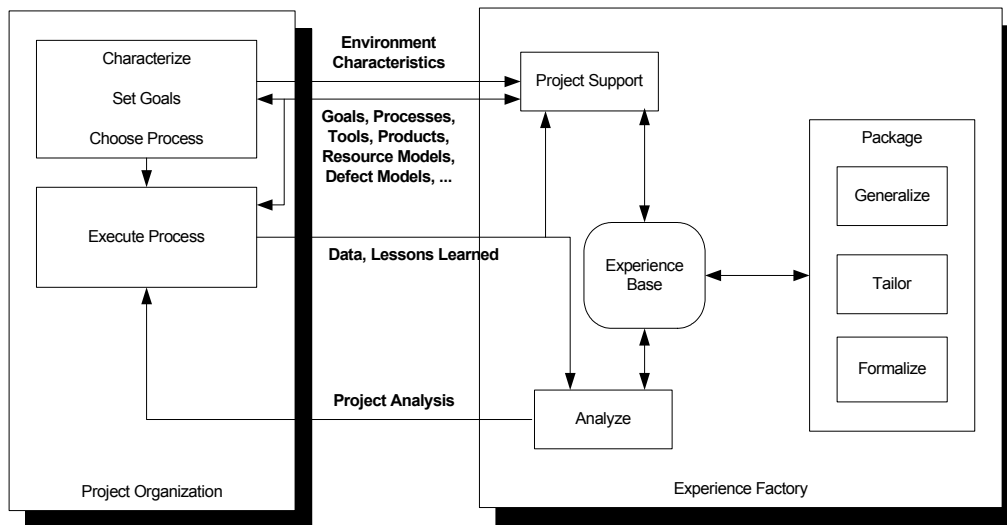


Figure 8-2 The Experience Factory and the Project Organization

In Figure 8-2 the separation between the project organization and the EF is illustrated. The project organization provides the EF with project and environment characteristics, development data, resource usage information, quality record and process information. Also feedback on the actual performance of the models that the EF has processed and is used by the project is made. The EF will then, by processing the information given, return direct feedback to each project as goals and models tailored from similar projects. If the project requests, the EF can also give baselines, tools, lessons learned and data, parameterized according to the characterization of the project.

8.3.1 Examples of Packaged Experience

The EF can package all kinds of experience. There are a variety of forms and some examples are

- Equation defining the relationship between variables
- Histograms or pie charts of raw or analyzed data
- Graphs defining ranges of “normal”
- Specific lessons learned associated with project types, phases, activities or risks and recommendations

8.3.2 Examples of Experience Packages

The main product of the EF is the Experience Package. The content and structure of each package depend on what kind of cluster of experience is packaged. In each packaged there is a central element which defines what kind of package it is, for example a software life cycle product or process, a mathematical relationship, a database.

Examples:

- Product Packages; the central element is the life-cycle product clustered with the information needed to reuse it and lessons learned in reusing it. (Programs, Architectures, Designs)

- Process Packages; the central element is the life-cycle process clustered with the information needed to execute it and lessons learned in executing it. (Process models, Methods)
- Relationship Packages; the central element is the relationship or a system of relationships from the collected data of a software project. (Cost and Defect Models, Resource Models)
- Tool Packages; the central element is a specific tool. (Code Generator, Configuration Management Tool, Static Analyzer, Regression Tester)
- Management Packages; the central element is a reference information for project management. (Management Handbooks, Decision Support Models)

8.3.3 Summary

The EF is a model for reuse of experience. It is implemented as a separate organization from the development organization and supports the project organization with collecting data, analysis and feedback.

EF can handle several different software life cycles, methods and techniques and reorganizes this experience into experience packages, which is the main product of the EF.

It is refined when reuse is made from ascending number of similar projects.

The minimal requirement for implementing the EF model is two people and the cost is around 5% of the total project cost.

Chapter 9

Evaluation of the Iterating Evaluation Model and the Experience Factory vs. the interview survey

Requirements	Iterating Evaluation Model	Experience Factory
Show where the problem occurs, by using a simple scale consisting of values like good/bad when performing the evaluation.	Yes	N/A Probably defined during the analysis in the EFO when data is collected and analyzed.
Information from the evaluation should be shared with other personnel including quality staff, in educational and information purpose.	Yes	N/A Depends on how the project organization handles the experience feedback given.
Tracking of the follow-up should be possible and sufficient feedback needs to be made.	Yes Since the preceding project is evaluated the follow-up is made and feedback reviewed.	Yes Data is collected from the project organization and analyzed in the EFO.
Not be too extensive or resource and time consuming/demanding, thus the overhead and administration should not increase noticeably.	Yes 1 person is required to implement the model. Not necessary fulltime.	Yes/No 2 persons at least required to implement the model at a minimal level. An average overhead is said to be 5% of project cost.
The implementation of the model should be made with minimal work effort for the organization.	Yes	No New organizational structure needs to be made. The creation of an EFO separated from the development organization, thus requiring its own funding.
Make everyone participate into the evaluation work.	Yes By dividing people into interest groups.	No Only those in the EFO make the analysis.
Give personal value back.	Yes Indirectly in small organizations.	Yes/No Depends on how the development organization handles personal feedback.
Establish routines which are natural to follow.	Yes	Yes
The person responsible for the evaluation needs to create openness when performing the evaluation.	Yes It is easier when handling specific interest groups.	N/A

Support iterative collection of data over time	Yes	Yes
Make it able to perform frequent evaluations	Yes	Yes
Support for time planning	Yes	No need from project organization, but implemented in the experience factory organization although the project organization has to have data collection incorporated into the process.
Involve the customer into the process	Yes	N/A If that is an experience it will be handled in an experience package.
The prerequisite for introducing an evaluation model are:		
Before implementing any model for evaluation, the process has to be defined.	Yes	Yes
Be initiated by the top level management, thus supporting the person responsible	Yes	Yes Creating an EFO expects separate funding, thus initiated by top-level management.
Have adequate information about what output is needed from the model to ensure the right decision when making an improvement to the process.	Yes	Yes

Table 9-1 Data analyzed for the Iterating Evaluation Model and the EF

9.1 Experience Factory vs. Iteration Evaluation Model

What is better? What aspects should be considered when choosing a model for handling reuse? Remember: the main focus is set on small organizations and software development organizations.

The main differences can be said to be:

The EF is concerned with separating the experience/analysis of data to a separate organization, the EFO; the IEM is included as a phase within the software life cycle.

The EF is better suited for a large organization since it can handle different types of projects; the IEM is concerned with small size organization where usually limited types of projects are conducted.

The EF can handle different software live cycles, process and methods. The IEM has its main focus on a SPI-model.

The main argument for choosing the IEM is

- it does not need a large change in the organization to be implemented
- the handling of one software process improvement model is sufficient to get started
- a personal staff of one person, the quality engineer, is justifiable when comparing economic cost

Since the IEM is better suited with small organizations that have only a small software development organization and can only handle a few types of projects the Experience Factory is a good model when the organization grows.

When introducing a SPI-model, the small organization will probably only benefit from some parts of it. These parts should be implemented as a start. When the organization grows and funding among other issues is possible the rest of the SPI-model can be implemented. Perhaps other project models and software life cycles will be handled, which makes the Experience Factory a possible model to use.

Chapter 10

Summary

10.1 Conclusion

The topic for this master thesis is to find a way for small organizations to improve the final delivered product's quality and this is made through improving the software process improvement.

Software process improvement models do exist such as the CMM or SPICE, but they are too exhaustive for small organizations. The focus is laid on small organizations because larger organizations usually have the funding required to implement large SPI-models. Furthermore, small organizations grow (hopefully) and if there is a focus set on process improvement early, the cost will be less over time when doing process improvement.

The idea is to start with the project process and evaluate it. Using this collected information an approach towards an improvement should be possible. By using an evaluation model it will enhance the organization to be able to develop routines and aid the improvement process.

The first step towards creating an evaluation model is to determine how evaluations are made out on the field. For this an interview survey has been made which shows important criteria for such a model. Among those criteria's the main aspect is

- The work effort and cost should be minimal
- A high gain in personal awareness of the software process improvements made

The main prerequisite is that the software process must be defined before performing any improvement otherwise measurements for where to improve cannot be made.

The interview survey supports the IEM. An already existing model, the Experience Factory, has been evaluated towards these criteria's and the analysis shows that the IEM is less cost and work effort demanding than the Experience Factory and makes personnel within the software development organization participate. The Experience Factory does have many of the criteria's needed so it could very well be used further on to implement or succeed the IEM. Perhaps IEM can be used as a "step-in"-model for software process improvement.

10.2 Strength and weaknesses

The IEM is general. It is made that way so that it can be used and adopted for any software organization. A possible objection is that it is too general and does not give enough guidance.

The strategy of the evaluation model is to find the most essential parts of an SPI-model to implement. This might not be applicable on every SPI-model, and therefore useless. If parts from a SPI-model are used incautious it will not result in any improvement of the project process, but instead chaos.

The IEM is concerned with small software organizations thus only focusing on one SPI-model at a time and one software life cycle. The container for the experience of the evaluation is stored

within the software process defined documents, thus enabling the information to be stored within the organization. Any further experience might be lacking when changing to a different software life cycle or SPI-model.

The strength of the IEM is the low cost and simplicity for implementing it. From the survey conducted, evaluations are seen with satisfaction when performed.

Improvements can be implemented and monitored with a simplicity well suited for the small organization.

10.3 Future work

This master thesis is concerned with improving the software project process by using evaluation of software projects. The definitions of process, quality and software process improvement are given and a procedure on how to perform the evaluation. The IEM's final outcome is the IAP-document. This IAP-document involves parts from a SPI-model. The weakness of the IAP-document is that it can hold many action points to be implemented, especially when doing the evaluation at an early stage when just a small part of an SPI-model is implemented. If a large amount of actions points are to be implemented than the workload could get to heavy for a small organization. To avoid this problem the action points need to be prioritized.

The Experience Factory is a good model to convert into when the organization grows. But regarding how well the IEM suited as an entry point to the Experience Factory is not examined in this master thesis.

Appendix A

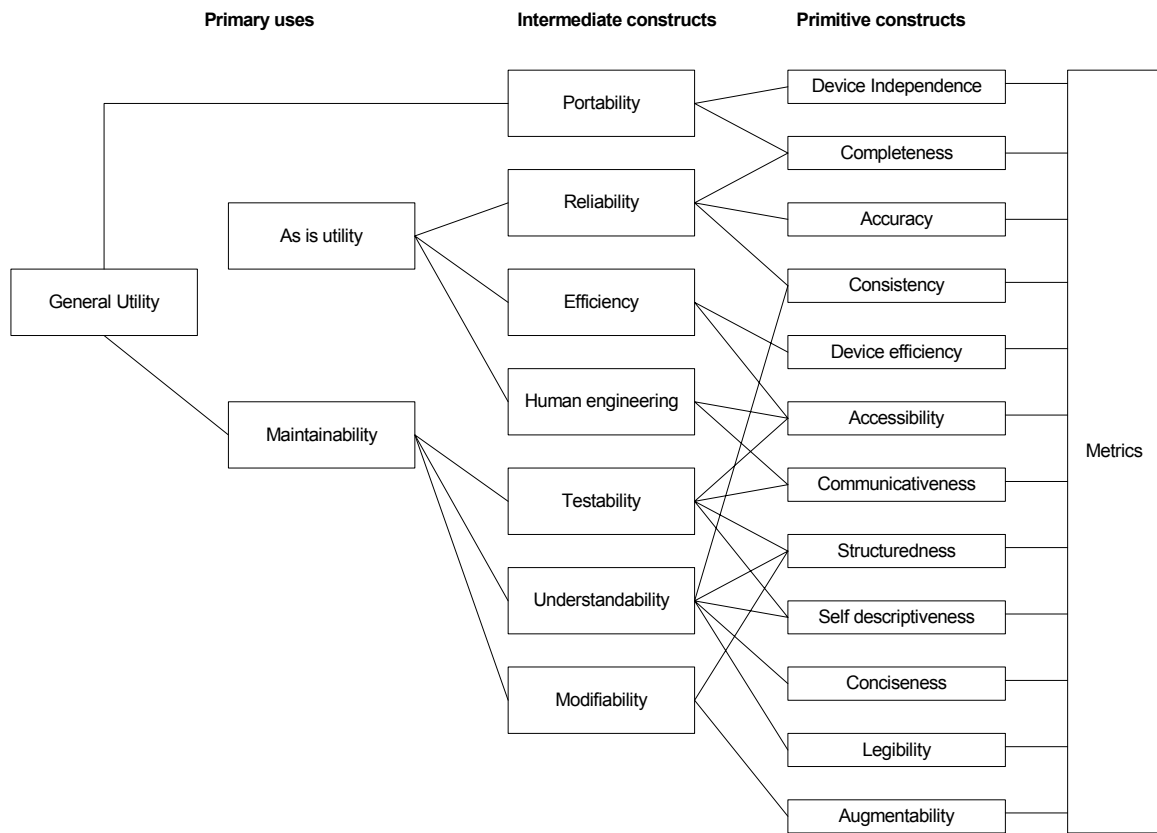


Figure 10-1. The Boehm software quality model

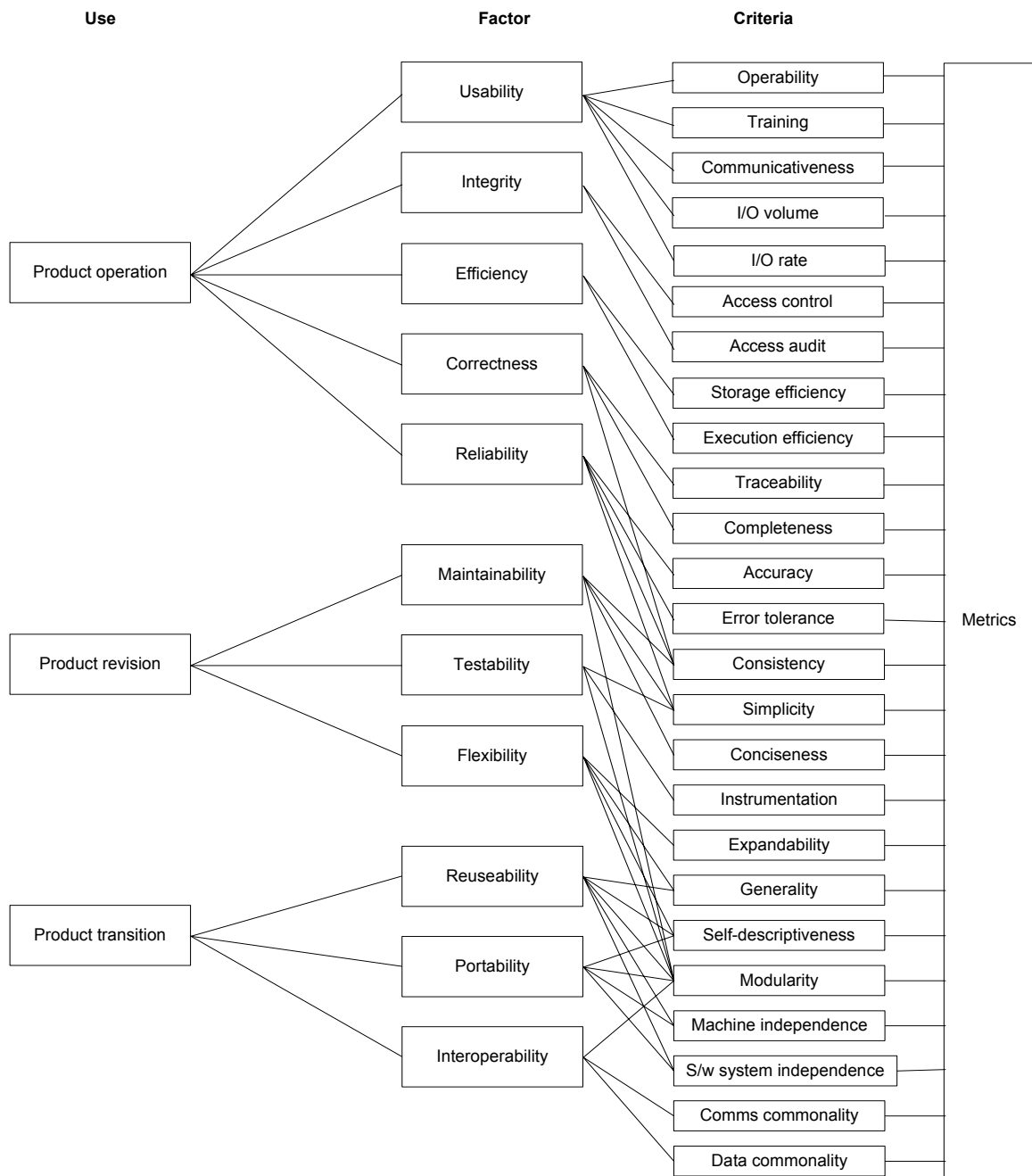


Figure 10-2. McCall software quality model

Reference List

- [Basili] V.R. Basili, G. Caldiera, H.D. Rombach, "The Experience Factory", Encyclopedia of Software Engineering Volume 1, John Wiley & Sons, 1994
- [Bass] L. Bass, P.Clements, R. Kazman, "Software Architecture in Practice", Addison Wesley Longman, March, 1998
- [Briner] W. Briner, M. Geddes, C. Hastings, "Projektledaren", Svenska Dagbladets Förlags AB, 1990
- [Deming] E. Deming, "Out of the Crisis", MIT Center for Advanced Engineering Study, MIT Press, Cambridge 1986
- [Dunn] R. Dunn, 1990 "Software Quality: Concepts & Plans", Prentice Hall, 1990
- [Eriksson] H. Eriksson, "Objekt-orienterad programutveckling med C++", Studentlitteratur, Lund 1992
- [Fagan] M. E. Fagan, "Design and Code Inspections to Reduce Errors in Program Development", IBM Systems Journal, 1976
- [Fenton] N.E. Fenton, S. L. Pfleeger, "Software Metrics", PWS Publishing Company, Boston 1997
- [Gilb] T. Gilb, "Principles of Software Engineering Management", Addison Wesley Longman Inc, 1988
- [Herbsleb] J. Herbsleb, D. Zubrow, D. Goldenson, W. Hayes, M. Paulk, "Software Quality and the Capability Maturity Model", *vol 40 Communications of the ACM*, June 1997
- [Humphrey] W. S. Humphrey, "Managing Technical People", Addison Wesley Longman Inc, December 1998
- [Kotonoya] G. Kotonoya, I. Sommerville, "Requirements Engineering", John Wiley & Sons Ltd, 1997
- [Larman] C. Larman, "Applying UML and patterns – an introduction to object oriented analysis and design", Prentice Hall PTR, New Jersey 1998
- [Lockyer] K. Lockyer, J. Gordon, "Project Management and Project Network Techniques", Financial Times Magazine, 1996
- [Macheridis] N. Macheridis, "Projektaspekter", Studentlitteratur, Lund 2001
- [Nicholas] J. Nicholas, "Managing Business & Engineering Projects", Prentice Hall Inc, New Jersey 1994
- [O'Hara] F. O'Hara, "European Experiences with Software Process Improvement", *ICSE 2000/ACM 23000 1-58113-206-9/00/06*, Ireland 2000
- [Robertson] S. Robertson, "An Early Start to Testing: How to Test Requirements", *EuroSTAR '96*, Amsterdam December 1996
- [Sakamoto] K. Sakamoto, K. Nakakoji, Y. Takagi, N. Niihara, "Toward Computational Support for Software Process Improvement Activities", *IEEE 0-8186-8368-6/98*, 1998
- [Wisén] J. Wisén, B. Lindblom, "Effektivt projektarbete", Norstedts Juridik AB, Stockholm 2001
- [Zahran] S. Zahran, "Software Process Improvement", Addison Wesley/Pearson Education Limited, 1998