PROCESS IMPROVEMENT IN PRACTICE

A Handbook for IT Companies

Tore Dybå

Torgeir Dingsøyr

Nils Brede Moe

SINTEF

Kluwer Academic Publishers
Boston/Dordrecht/London

PROCESS IMPROVEMENT IN PRACTICE

A Handbook for IT Companies

PROCESS IMPROVEMENT IN PRACTICE
A Handbook for IT Companies

by

Tore Dybå
Torgeir Dingsøyr
Nils Brede Moe
SINTEF Information and Communication Technology
Trondheim, Norway

KLUWER ACADEMIC PUBLISHERS
NEW YORK, BOSTON, DORDRECHT, LONDON, MOSCOW

eBook ISBN: 1-4020-7879-X
Print ISBN: 1-4020-7869-2

Contents

Preface

Faster – better – cheaper. This is the challenge that software companies face every day. How to meet customer expectations in a world where continuously changing environments, organizations and technology is the rule rather than the exception.

One way to meet these challenges is to share knowledge and experience – use what others have already learnt. This may seem self-evident, but experience shows that a lot of people struggle to get started.

A lot of good books have been written on the principles and theories behind for example total quality management, change management and knowledge management.

What makes this book different from the others is that this book is a practical handbook. It is intended for small and medium-sized software companies that develop software – and that need some help to get started with systematic improvement work.

The handbook is filled with useful hints and examples to help leaders and employees to get started quickly. There is not much theory in this book.

On the contrary, this is a book showing how we have carried out improvement work in a number of companies since the mid-nineties.

We have used the methods and techniques in many different contexts, in small and medium sized companies. We have found that they work, that they can be scaled according to the needs, and that they are simple to use. We therefore want to share these experiences with you so that your company can utilize what others have learnt.

The book consists of two parts: Chapters one to six of this book describes methods and techniques for handling processes concerning knowledge and learning, improvement and measurement. Chapter seven describes some of the techniques in more detail, to make it easier for you to start using them.

Read the chapter most relevant to you, take a look at how we have used the techniques, and compare them with your own experiences and get started.

Trondheim, Norway, January 2004

Tore Dybå, Torgeir Dingsøyr, Nils Brede Moe

Chapter 1

INTRODUCTION

1.1 WHAT IS THE PURPOSE OF THIS HANDBOOK?

This handbook is made to help project leaders and project participants in small and medium-sized software organizations to improve the work practices in their own company.

Here, you will find practical information on what you need to know to get started with and implement improvements in your organization.

The improvement process and techniques described in the book are adapted to software organizations from the principle of total quality management and continuous improvement

The handbook gives you an introduction to basic principles for *knowledge and learning* in software companies, it describes a detailed *improvement process* based on experiential learning, it describes how *measurement* and *process assessment* can be integrated into the improvement process, and it includes a section of *techniques* that can be helpful in the practical improvement work. References to these techniques are marked in italics, for example: *brainstorming.*

Before we start with some of the more detailed descriptions, we will take a short look at some basic terms, look at the advantages of process improvement and give some hints on how you might get started.

1.2 WHAT IS A PROCESS?

A process is nothing more than a structured set of activities and decisions to do a certain job. Everything we do involves processes.

Here are some examples:

- Plan a project
- Design a website
- Program a function
- Prepare a meeting
- Develop a requirement specification
- Write a document
- Implement a test
- And so on ...

As you can see, processes vary both in importance and complexity, and they can appear at different levels by main processes and sub processes.

Common to all the processes are that they consist of one or more activities. The activities require an effort and have a result, and they are implemented by adding some form of value to the effort. The value of the activities may vary, but the purpose is to implement something that is planned and thoroughly considered.

Several processes can be put together making up a life cycle model for software development. Depending on how the life cycle model is designed, we can distinguish between various development strategies as shown in figure 1.1.

By using the *waterfall* model, the activities and the steps in the activities are usually performed in a series: identify user needs, define requirements, design the system, develop it, test it, and deliver it to the customer.

In an *incremental* life cycle model, the development process also starts with a set of given requirements. Thereafter, the development is carried out in a sequence of increments. The first increment includes a part of the requirements, the next adds more requirements, and so on, until the system is developed. For each increment, all necessary processes and activities are implemented, for example detailed design, coding, test and integration, mainly in a series.

By using an *evolutionary* life cycle model, the system is also developed in many increments, but the requirements are not given in detail in advance. In this model, the requirements are only partially defined in the beginning, and then they are matured and detailed by each increment. Processes and activities used to develop the increments are carried out in series or in parallel and are partially overlapping.

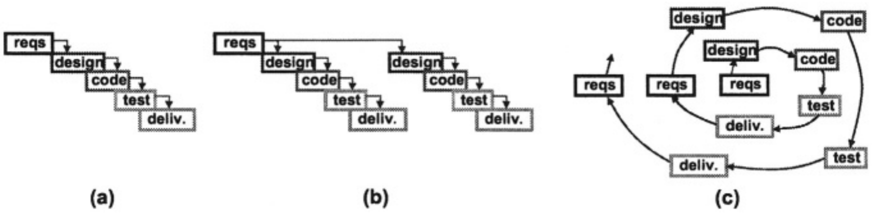


Figure 1.1. Development strategy: (a) waterfall, (b) incremental and (c) evolutionary.

It may be useful to use the Intranet to make access to the organization's standardized life cycle model easier. Intranet-based solutions make it simpler to adapt and define specific project models. In that way, the process model can also function as an *electronic process guide.*

1.3 WHAT IS PROCESS IMPROVEMENT?

Process improvement is about making things better – not about fire fighting or handling crises. It is about stop blaming "someone" for problems or faults. It is a way to look at how we can do our work better.

If we only solve a problem or correct a fault, we risk not finding what the underlying causes are. In the worst case, it can lead to things getting worse. In addition to identify problems, we have to find the causes, define, implement and follow-up appropriate actions, evaluate the results of the actions and carry out possible changes in the rest of the organization.

When we engage in process improvement, we want to learn about what happened in a process, and to use that knowledge to improve the process and the resulting services and products.

1.4 HOW CAN PROCESS IMPROVEMENT HELP SOFTWARE ORGANIZATIONS?

Process improvement will help the software organization to evaluate and improve its own performance. When all key people and employees actively take part in process improvement, the organization can easier focus on how things can be done better, faster and cheaper.

Participation and ownership are fundamental for all team- and project-based work. Use of the employee's collective knowledge and experience is therefore a strong instrument in the work of process improvement. Through teamwork the totality becomes larger than the sum of the parts.

1.5 HOW CAN WE GET GOING WITH PROCESS IMPROVEMENT?

The first step to get going is that the organization's managers set priority on process improvement, and that they clearly signal that continuous improvement is important to the company's overall goals and strategies.

The managers must arrange for process improvement, and make sure that the employees get sufficient time to reflect on their own practices. Such arrangements can include physical, organizational and technical conditions in the company.

To get started, it is important that managers and employees who have been doing fire fighting put away their **CO_2-bottles,** and start thinking about the following:

- Which processes have we got?
- Which process do we need to improve?
- How can we improve the process?
- What kinds of resources are required to improve the process?
- How can we make use of the improved process?

1.6 WHAT IS SPECIAL ABOUT SOFTWARE DEVELOPMENT?

Improvement techniques exist in most business sectors, such as total quality management in the traditional industry. The question is if we can make use of such ideas in the software domain, and at the same time allow for the particularities that characterize software development.

During the last couple of decades, a lot of things have happened that improve our understanding of software development. The following characteristics can be relevant for the improvement work in a software organization, and we should, therefore, pay attention to them in the development of an improvement process:

- Software is developed – not produced. This implies that a lot of the characteristics for the development process are unique, and that experience not necessarily can be transferred between projects without paying attention to these.
- Software development is an experimental process. This implies that we have a continuous need to collect experience and make them available for new projects.

- Most techniques for software development are human-based. This implies that the effect of using these techniques is often difficult to measure and to replicate.
- Each software product is unique when it comes to use, purpose and environment. This implies that purposeful development methods and measurements may differ for different projects.
- Assessment and packaging of experience to increase the reuse value for future projects require resources beyond the normal project environment. This implies that systematic improvement requires its own mechanisms outside the individual project.

1.7 BASIC PRINCIPLES

When designing the improvement process, we have used the specific needs of the software sector as a basis, and emphasized the following principles:

- The main focus is on product and service improvement. Process improvement is just a means to achieve this.
- Business orientation and evaluation of results are of critical importance to assure that the improvement work has the right direction
- The improvement work must be seen as a continuous learning cycle, with a high degree of participation from both managers and developers.
- The organization's development projects make the natural basis for a learning process through experimentation and knowledge sharing.
- Improvement results and experience must be analyzed and combined so that they are available for future projects.

1.8 OVERVIEW OF THIS HANDBOOK

This handbook consists of two parts: One process-oriented part and one technique part. The remaining chapters in part I are:

- Chapter 2: *Knowledge and learning,* argues that small and medium-sized companies should exploit their advantage of being small by letting the employees learn from each other. This is what we call sharing tacit knowledge. We will present several tips on physical, organizational and technical changes to enforce learning in the company. We also provide tips on how to improve the learning in the projects and how to learn more by stimulation of creativity.

- Chapter 3: *Improvement process*, gives concrete advice on how learning can be supported in development projects by organizing learning meetings in the initiating phase, short learning meetings in the execution phase, and post mortem analyses in the project closure phase. We also look at how iterations in the project, or processes in the project, can be improved by planning improvement goals, implement changes and at the end assess how the changes have worked.

- Chapter 4: *Measurement and feedback,* describes how we by collecting data on different aspects in a project can make a solid basis for improvement. We recommend a way to do measurement that clearly define the goals for the measurement, the questions we want to get answered, and at the end, what measurements we should do.

- Chapter 5: *Process assessment,* describes a participatory approach to software process assessment, based on the organization's specific goals and needs. The method emphasizes what is unique in each company, and how this uniqueness can be exploited as a competitive advantage

- Chapter 6: *Process guides,* is about what can be valuable for a company in having guidance to normal work processes in the company, and describes a method where many employees are involved in defining the work processes. The chapter also deals with adaptation of process guides to different types of projects, and maintenance of process guides.

In chapter 7 you find a series of *techniques* for help in various improvement activities. The techniques are described in a pragmatic and easy to understand manner. A lot of the techniques are also applicable in other settings than in process improvement work.

Chapter 2

KNOWLEDGE AND LEARNING

In this chapter we argue that small and medium sized companies must exploit their advantage of being small by letting the employees learn from each other. We will present several tips on physical, organizational and technical arrangements to enforce learning in the company. We also provide tips on how to improve learning in the projects and how to learn more by stimulation of creativity.

2.1 WHY LEARNING?

Being a good software developer requires that you are up-to-date on existing technologies and that you are able to understand the customers' needs. In knowledge-based companies it is a requirement that the employees must learn to do a good job, or to make an even better job. A job is also more motivating if the employees can develop through their work by learning more.

2.2 WHAT DOES LEARNING MEAN?

Learning is to acquire knowledge, and there are many ways to do that: The way we learn depends on the type of knowledge we are supposed to learn. Roughly we divide knowledge into two types:

- *Tacit knowledge* – this is knowledge that is implicit in people and influences the way we act. Another word for this type of knowledge is skills.

- *Explicit knowledge* – is knowledge that we can express and write down, for example in a description of how to perform a certain task.

Learning implies both tacit and explicit knowledge – in small and medium sized companies the tacit part is probably the most important. It takes too much time to write it all down, and it also takes too long to read all that has been written down before you start a job.

Acquiring the tacit knowledge from other people through observation, imitation and trial is often called *socializing*. Working together with other people in a project makes you learn a lot that you as a learner as well as the person you learn from need not be aware of.

Acquiring knew knowledge by reading books and documents is called *internalization.* We then use knowledge that is explicitly available and transforms it to make use of it later.

We may also learn by combining material that is explicitly available, this is called *combination.* We can for example combine a process description with an experience report from a project. Or we can make our own tacit knowledge explicitly available, for example by using a post mortem analysis that will be presented later. This is called *externalization.*

Effective learning depends on the combination of different learning strategies – all jobs depend on our tacit and explicit knowledge and that we are able to modify this knowledge according to changing environments.

We can learn a lot from colleagues at work, by participating in a network with people in other organizations or by trying out new methods and techniques and experimenting with them. Exchanging experience with other people sharing the same interests can be a good learning strategy as the information that is exchanged will be very relevant. Communication across companies is one of the features in the of Silicon Valley environment. But the best learning place will often be the practical project work.

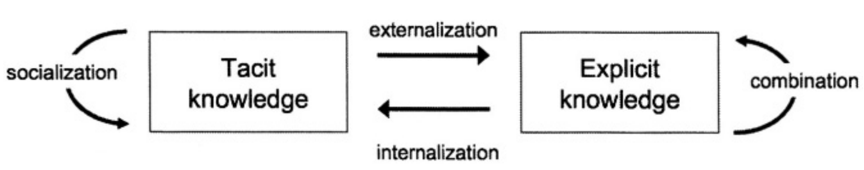


Figure 2.1 Learning from tacit and explicit knowledge.

How can we provide a good learning environment in a small company? We believe it is a matter of making room for communication and reflection in a hectic working day. It may be a matter of making arrangements in the physical environment as well as to the organizational structure and the technical infrastructure. We may also make arrangements within the projects.

We will first look at the physical arrangements supporting socialization. We will then look at the arrangements supporting the transfer of explicit knowledge. For small companies, however, we believe that transfer of tacit knowledge is by far the most important, as this is the most effective communication form.

Communication effectiveness

The most effective way to learn is to transfer ideas or knowledge when the communication is carried out face to face and with supporting devices for the transfer of information. Two people working round a whiteboard is a better way of transferring knowledge than one person e-mailing the other.

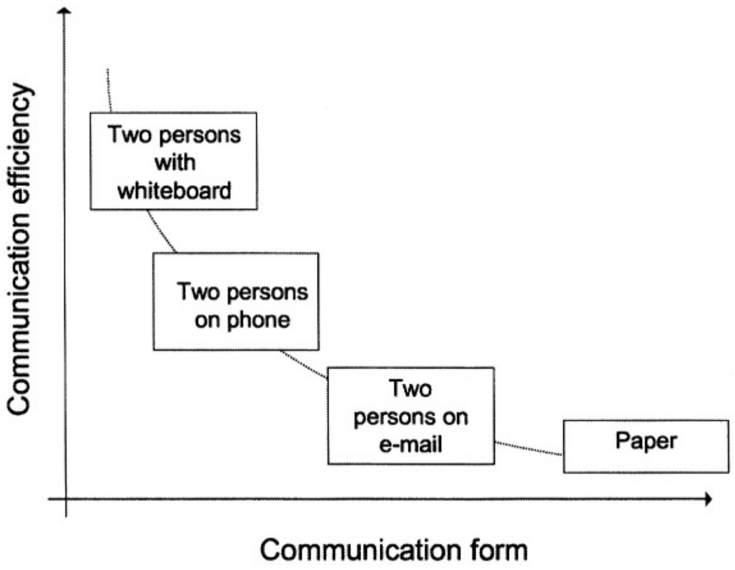


Figure 2.2 – Communication is most effective when people meet face to face with supporting devices for the communication.

2.3 PHYSICAL ARRANGEMENTS

We can provide a physical learning environment in the company by making arrangements in the project working room and by setting up rooms for general communication in the company.

Project room: When people work together in a project it may be an advantage to be located in the same room to simplify communication. If you need to leave your own office to go to your colleague's in order to ask a question, you will often stay put with your question longer than necessary. In a project room it would be normal to have equipment like a whiteboard and stickers to support your discussion visually. To provide an overview you could have stickers on a board showing the tasks and an overview of what the different members of the project are doing.

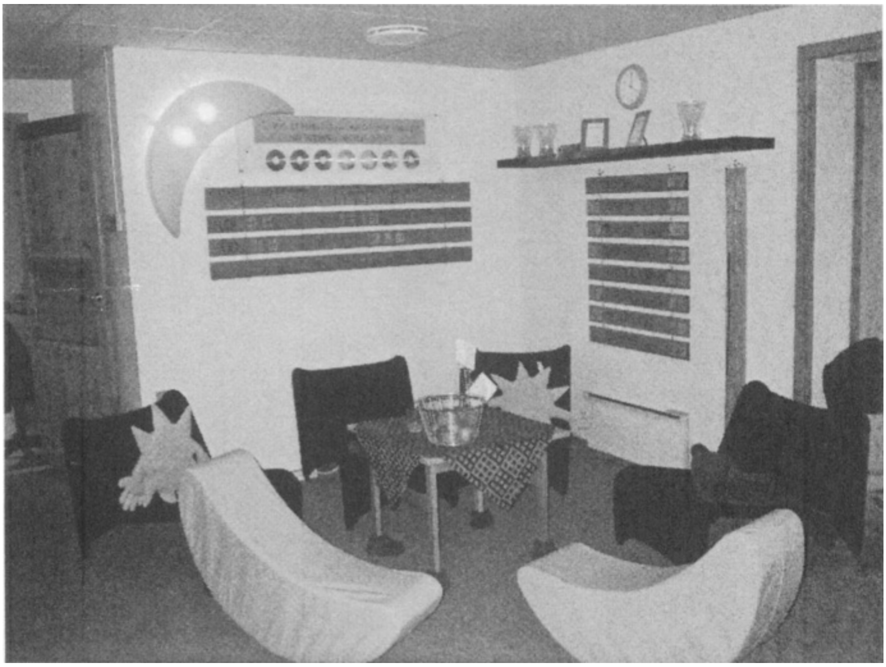


Figure 2.3: Part of the project room at Visma in Norway – posters showing project tasks and status are placed on the wall.

Meeting points: It is important to have informal meeting points in the company where people can get involved in conversations. Such places are typically around the coffee machine and by the copier or printer. It could be wise to arrange these places so that the communication will work even better. Provide a whiteboard nearby and a notice board where you for example can inform on what the different people in the company are working with. Providing a stand-around-table for short improvised meetings is also a good idea.

2.4 ORGANIZATIONAL AND CULTURAL ARRANGEMENTS

Several actions can be made to provide a learning environment – even if the company is a small one. Below follows some hints on how to organize the company, stimulate communication, and work towards a culture with sharing of experience.

Job rotation: The responsibility for some mandatory tasks should circulate – for example responsibility for software quality, or writing of user documentation are skills that wisely should be held my more than one person.

Pair work: Working closely with other people on one task makes us learn much more that when we work by ourselves. Match older employees with the newcomers to give the new ones an introduction to matters that are not written down in the company. It may also be a good idea to work in pairs when doing the programming work – to ensure that the solutions are discussed and the quality of the code is improved.

Inform each other: Use some time during meetings to inform each other on what you are presently working on, Giving advice or explaining problems will then become easier if you get stuck.

Arrange internal training: The best way to learn is to teach others. If you possess a special knowledge within the company, provide a course for the others teaching them what you know. It need not be a long course, but may for example be done during an expanded lunch break. Make professional groups for subjects that might be of general interest for others.

Show your skills: It is important that the people working in a company know what the others are good at. Some companies arrange "knowledge fairs" on a regular basis. Everyone presents their areas of special knowledge, for example by using posters in the area where it is possible to mingle and talk to others. This may also be a good idea for smaller companies.

Work on attitudes: Examples can be found on companies where a culture of unhealthy internal competition has hampered the sharing of information across the company. If a project failure damages the career for the project leader, this does not stimulate the discussions that could encourage learning.

Making it visible that the company is aiming at sharing knowledge: Make the employees know that experience sharing is appreciated. In one company they put up posters in the staff restaurant that encouraged people to spend time on experience sharing.

Pair programming

Pair programming is a way of working that provides learning for the participants and that ensures that the selected solutions are discussed.
The entire code is written on one computer with two persons placed together. The one using the keyboard and mouse is doing the actual programming while the other one is planning and making strategy considerations, checking for faults and assessing alternatives. The other one can also check on information sources and make sure that the code is complying with code standards. The two persons interchange the roles regularly.

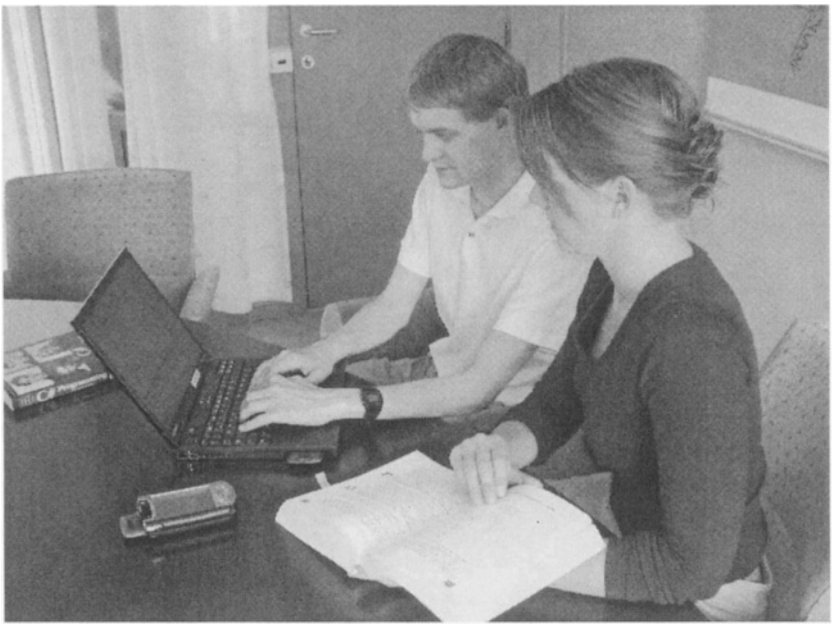


Figure 2.4 Two developers working together on writing code.

Deviation – gems of wisdom in the learning process!

In the company Visma deviations are reported to the management group that maintains a list of tasks to be worked on. These deviation are called *gems of wisdom* because they want deviations to be looked upon as something positive that makes the organization able to learn. The list with the gems of wisdom are distributed to the managers in the management group who are assigned responsibility for the follow-up. The person who reported the gem of wisdom can check in the list to find the status and see what actions have been or will be taken. In a learning process we often distinguish between single loop and double loop learning.

Single loop learning is to change practice as problems arise – in order to avoid the same problem another time. An example of this is to correct a bug in a software module after it was found during testing.

Double loop learning indicates that you use the problems to understand the underlying causes: assumptions and conditions for actions and methods – and then do something about them. That is, in addition to correcting the bug, you try to do something with whatever caused the bug to be introduced.

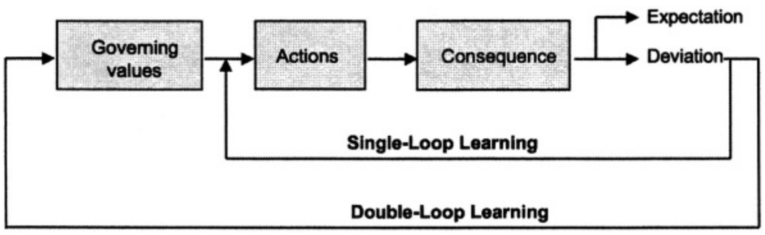


Figure 2.5 – Single and double loop learning.

By handling deviations as gems of wisdom, Visma provides an opportunity for both double and single loop learning.

2.5 ARRANGEMENTS FOR TECHNICAL INFRASTRUCTURE

How can we make sure that what you or someone else in the company has learnt is actually put into practice later? The most important thing is to make everyone in the company aware of the knowledge base available – either by letting people present their special knowledge, or by providing overviews of who knows what. Additionally it may be wise to structure the available written knowledge. You may for example present knowledge related to projects in a process model for projects. It is wise to keep such overviews in a place where it is easily accessible. Some technical arrangements may be made to support communication and reflection in the company. A simple action can be to regularly publish a newsletter. Companies with more resources may establish an intranet.

Newsletter: An internal information note may be useful for distribution of information on status for the different projects, interviews with colleagues, hints on relevant courses, books or websites.

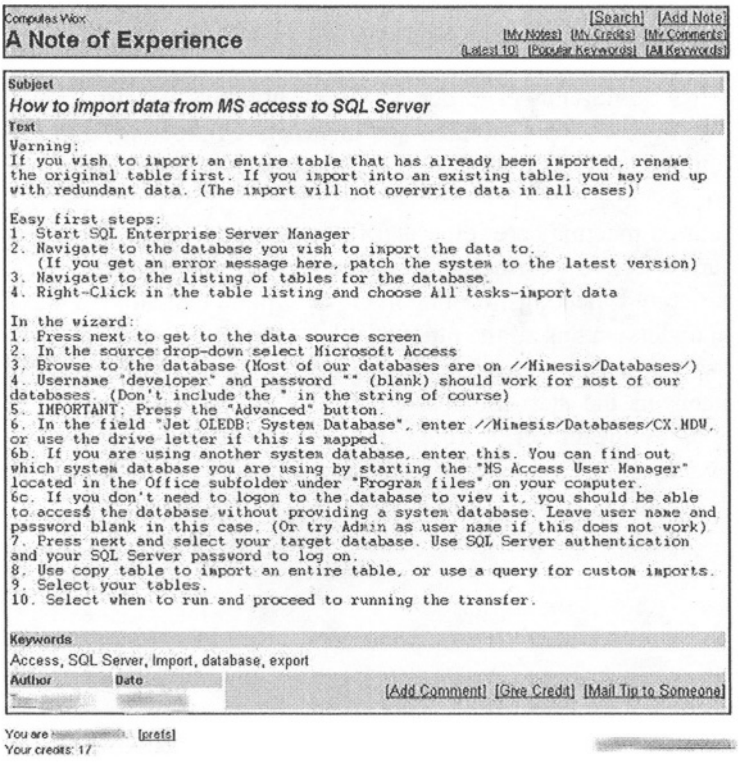
Intranet: The pages on the company's intranet may hold overview on projects, process guides for projects, overview of special knowledge in the company, along with checklists, for example for code standards and kick-off meetings for a project. In addition the company may publish experience reports from the projects, or testing reports for new technology or new methods.

Areas where collections of experiences may become useful are:

- Project management
- Quality management
- Risk handling
- Architecture design
- Detail design
- Implementation
- Maintenance
- Migration
- Configuration management
- Re-use
- Technologies
  - Java
  - ODBC
  - .Net

Experience notes on the intranet

The company Computas has provided the possibility for everyone in the company to contribute with their own useful experiences on the intranet – it is a system for collective yellow stickers.


Computas Wox
[Search] [Add Note]
[My Notes] [My Credits] [My Comments]
[Latest 10] [Popular Keywords] [All Keywords]

A Note of Experience

Subject

How to import data from MS access to SQL Server

Text

```
Warning:
If you wish to import an entire table that has already been imported, rename
the original table first. If you import into an existing table, you may end up
with redundant data. (The import will not overwrite data in all cases)

Easy first steps:
1. Start SQL Enterprise Server Manager
2. Navigate to the database you wish to import the data to.
   (If you get an error message here, patch the system to the latest version)
3. Navigate to the listing of tables for the database.
4. Right-Click in the table listing and choose All tasks-import data

In the wizard:
1. Press next to get to the data source screen
2. In the source drop-down select Microsoft Access
3. Browse to the database (Most of our databases are on //Mimesis/Databases/)
4. Username "developer" and password "" (blank) should work for most of our
databases. (Don't include the " in the string of course)
5. IMPORTANT: Press the "Advanced" button.
6. In the field "Jet OLEDB: System Database", enter //Mimesis/Databases/CX.MDW,
or use the drive letter if this is mapped.
6b. If you are using another system database, enter this. You can find out
which system database you are using by starting the "MS Access User Manager"
located in the Office subfolder under "Program files" on your computer.
6c. If you don't need to logon to the database to view it, you should be able
to access the database without providing a system database. Leave user name and
password blank in this case. (Or try Admin as user name if this does not work)
7. Press next and select your target database. Use SQL Server authentication
and your SQL Server password to log on.
8. Use copy table to import an entire table, or use a query for custom imports.
9. Select your tables.
10. Select when to run and proceed to running the transfer.
```

Keywords

Access, SQL Server, Import, database, export

Author	Date
[redacted]	[redacted]

You are [redacted] [prefs]
Your credits: 17

[redacted]

Figure 2.6 Experience note from "Well of Experience" – WoX – at Computas.

2.6 ARRANGEMENTS IN THE PROJECTS

The most important learning in a software company will take place in the daily run, which typically will be when working in a project team.

A team is usually assigned a very specific task, a set of resources and a way to organize. The team usually has a leader and one or more project members.

In order to carry out tasks in a team and to get the learning benefits it is important that everyone has the same goals, or at least is aware of the goals for the others in the project. In a team where people have different goals, conflicts will often arise, and much time is spent on tasks that may not be relevant. The project group shall be able to make considerations concerning complex problems, be able to act coordinated and innovative and at the same time have relations to other existing projects in the organization.

What is important in order to learn effectively during teamwork? Interchanging between structured and unstructured meetings and gatherings may stimulate the collective feeling and understanding and provide progress in the project.

Structured meetings are important for making decisions, handle conflicts and ensure effective information flow and distribution of tasks.

Unstructured meetings are important to stimulate creative solutions, and to get an understanding of the other people's opinions and views.

We will now first introduce some hints on how to stimulate learning in the projects in the start-up phase, and then look closer at dialogue and discussions as means for learning. Finally we take a closer look at some hints that may stimulate creativity in teamwork.

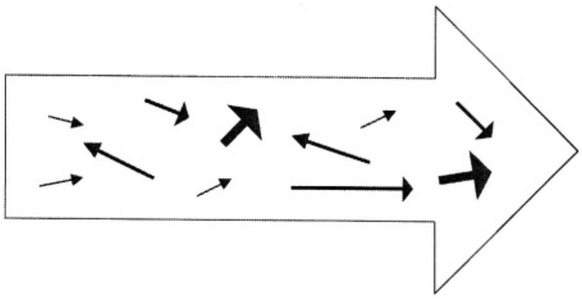


Figure 2.7 the participants in a project are not always working in the direction the project is supposed to have.

A simple way to get started with team learning is to regularly run briefings before starting an activity and debriefings when they are finished. If this is incorporated, people will draw experience from the projects. An expanded form of briefing – learning meetings – will be presented in the next chapter. Some hints for project start-up:

- Clarify what each member of the team recognizes as the task
- Clarify who is good at what in each team
- Clarify what learning possibilities there is for the organization in the task

A project will need discussion of ideas, creative thinking, and coordination of action and handling of conflict situations. If this is accomplished depends on the quality of the team spirit – the personal relationships, people respecting each other and having understanding for the different points of view. Techniques to create a good team spirit are often called team building techniques. Such techniques are described in other books.

In team work it is important to be able to exploit not only your own knowledge, but also the skills and knowledge of the others. We may call this for relational knowledge. If members of a project group do not know each other very well, it may be wise to discuss what each member is good at.

In order to learn this it is important to be challenged at work. Evidently it is not a piece of cake to know how big the challenge should be before the gap between what you know and what you want to accomplish becomes too big. In project work the members will need to talk about a lot of subject matters.

It is possible to distinguish between to types of communication – the dialogue and the discussion. Let's look at the dialogue first – a dialogue is a free, creative exploration of complex problems where everyone in the project group should participate. Dialogue is important because it provides insight in other people's points of view and attitudes and it stimulates the participants to come up with new ideas. It is important that a dialogue is not too controlled – nobody must feel that they leave a dialogue as "winners" or "losers". It is, however, the purpose that everyone should come forward with new ideas or with new understandings for a problem issue. It is important to be able to listen to other people, and not only do the talking yourself.

Creating a good dialogue is difficult – but it is possible to do some training in the project. A good dialogue also takes time, so make room for dialogues, especially in the beginning of the project. This is much more effective than spending a lot of time on discussions towards the end of the project. Additionally a good dialogue clarifies the different tasks in the project and prevents the project members from spending time on non-relevant tasks.

Listening training

A way of stimulating the ability to listen is to carry out a dialogue where everyone is blindfolded. Only being able to use your hearing provides increased concentration on paying attention to the conversation. Try to implement such an exercise by letting everyone sitting comfortably in a silent group room. Let the members be blindfolded and wait until the room is completely silent. Then tell everyone that they can talk about whatever they want, and let this go on for about 20 minutes. Ask everyone to remove the blinds and then talk about the experience.

In a discussion different views on an issue are presented and defended, usually because there is a need for a decision.

It might be useful to get feedback from the others on the way you communicate in discussions, in order to become more objective and fact oriented.

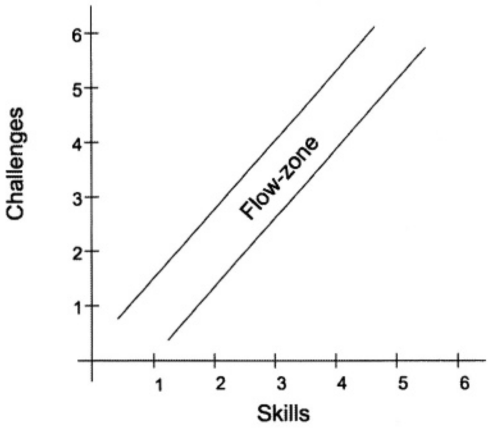
Practicing feedback on communication style

In order to give feedback to how people communicate, this exercise can be used. It lasts approximately two hours. Find a difficult question as a theme for the discussion. Make each participating person select a feedback partner who they feel comfortable with. Let the pairs discuss the question. Next you divide the participants into two groups, A and B with members from each pair. Place chairs in a circle for team A and an outer circle for team B. Let team A discuss the question and let team B observe.

The members of team B then give feedback on the behavior of their partners. The roles are changed and team B discusses the question in the inner circle while team A is observing. When the discussion is ended the partners provide feedback. Have a break and close the exercise with a discussion in plenum of what you have learnt about how you communicate.

Challenges and skills in soccer

In the book "Godfoten" (The better leg) the soccer trainer Nils Arne Eggen describes how the champion soccer team Rosenborg all the time provides the players with challenges that improve their skills. The challenge must be relative to the skills of the players. The players must be challenged in areas where they have skills in order to prevent them from getting bored or from getting into situations they cannot cope with. It is the responsibility for all the players to pass the ball in such a way that all players are challenged at the right level. Development through challenges that make you better is what Nils Arne Eggen calls working in the "flow-zone" - if you have skills on level four, you should also be given challenges on this level.



Figue 2.8 Rosenborg is working at giving people challenges according to their skills - they are working in the "flow-zone"

2.7 HOW DO WE STIMULATE CREATIVITY?

What is a creative person, and why is creativity so important in software development?

Being creative is to come up with new ideas. We often think of creativity related to innovation, which is to put new ideas into practice. Creativity may be new ideas on how to develop a better product or better ways of developing a product – learning new skills. There are a lot of techniques that can be used to stimulate creativity, in addition to the dialogue described earlier.

It is important that the company management realizes the importance of creativity and know how to stimulate creativity in the projects. A developer who had been working with creative techniques once said: Earlier I used to be far more secret when I had ideas that were "far fetched" – I felt that I should concentrate on the practical solutions. Now I do not care if people get the impression that I am daydreaming, - the management trusts our ability to generate creative ideas.

Some useful techniques are:

- *Brainstorming*: Make the participants freely come forward with suggestions concerning a problem issue - only new suggestions are allowed, no criticism of others. In this way a lot of ideas are generated and even "dumb" ideas may lead to new and "good" ones. (Brainstorming is closer described in chapter 7).
- *KJ:* Ideas are written down on stickers and put up on a whiteboard. The members groups the stickers to find connections and structures. (KJ is closer described in chapter 7).
- *5-W- and H-technique:* By answering questions starting with: Who–What–Where–When–Why and How, it may be possible to obtain a better insight in the problem issue and come up with new suggestions for a solution.
- *Peaceful place technique*:. Let the participants imagine they are in a peaceful place, for example on a mountain hike or by the sea, they may put themselves into a state of mind that makes them think differently.
- *Problem reversion:* Sometimes is may be useful to reverse a problem issue. If you are working on finding a good architectural structure for your program, it might be useful to ask: What is a bad program architecture for this system?

Chapter 3

IMPROVEMENT PROCESS

This chapter describes how process improvement can be organized, a model for project-based experiential learning, the different steps in the improvement process and the connection between them. The model is built on the principles discussed in the previous chapter.

3.1 ORGANIZING PROCESS IMPROVEMENT

Software development usually takes place within a project. Such projects may vary concerning available time as well as available resources. Usually the shortage in a project is human resources in the form of work hours. The participants in the project may have different roles – as project leader, test responsible, documentation responsible, and as responsible for customer relationships.

How do we organize process improvement in companies that mainly run their activities as projects? A lot of larger companies have their own Quality Assurance (QA) department or a Software Engineering Process Group (SEPG). Such QA departments or SEPGs usually have input to the project at start-up and at the end, and will be responsible for updating the processes and methods used in the company. They may also be responsible for ensuring mat the products are adequately tested before release. It may be very self-evident that such departments or groups also take on the responsibility for process improvement – often in cooperation with the ones working directly in the projects. A problem with such departments or groups is that they may easily be looked upon as "unnecessary" by the developers – unless they do not interact sufficiently with the projects where the products are being developed.

Other companies are organized as projects and processes. Each process has a process responsible who will usually have that role for 2-3 years. Then someone else takes over the role. Processes could be such as: quality, knowledge management, methods, development or customer support.

The process responsible need not work full time with this and may also use other people in the company to solve tasks related to the process.

Smaller companies may not have any administration at all and will almost only be working in projects. In such cases it will be possible for a larger project to assign a person responsible for process improvement – a person who in dialogue with other persons with the same responsibility in other projects may suggest improvement actions in the company.

The most important part concerning organizing process improvement is of course to make sure that someone is responsible for taking initiative and carrying through improvements.

It is also important that the initiators have discussion partners – either by having resources at their disposal in the company, or having a reference group. Another important issue is that process improvement should not be associated with “control actions” – the members of the projects are the experts in the company and they probably work better when they freely can organize their own work. Process improvement should help them to get an overview and provide help to prioritize the most useful tasks – making them work “better”.

3.2 A MODEL FOR PROJECT–BASED EXPERIENTIAL LEARNING

The improvement process consists of three main phases: *initiating, executing* and *project closure* as shown in figure 3.1 and described below. Overall, this makes an effective system for *project-based experiential learning* for small and medium sized companies.

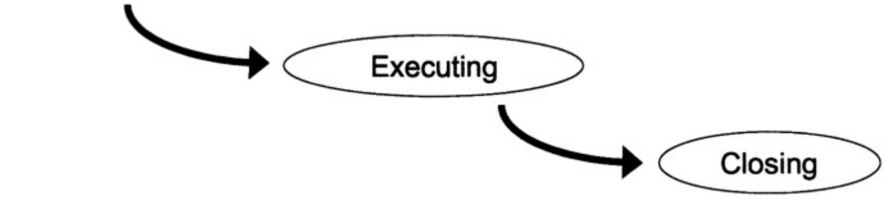


Figure 3.1 Learning processes in the project.

Initiating: Before the project starts we ask ourselves who have earlier been working on similar projects and what we may learn from them.

Executing: During the project work we regularly pause to check if we are on the right course, reflect on our experiences so far and identify and put into practice short-term improvement actions.

Project execution consists of development activities and processes that, in all and separately, go through a tree-step iterative learning cycle: plan, do and check.

Closing: When the project is finished we use some time to stop and reflect over what we have learnt in the project. We combine our experiences from the project in a way that makes them easily available to later projects.

As we can se from figure 3.2 there is an interaction between the main phases in the improvement process across the projects – the project closure phase contributes with input to the initiating phase of the next project. In this way we ensure effective transmission of experience and learning across the projects in the company.

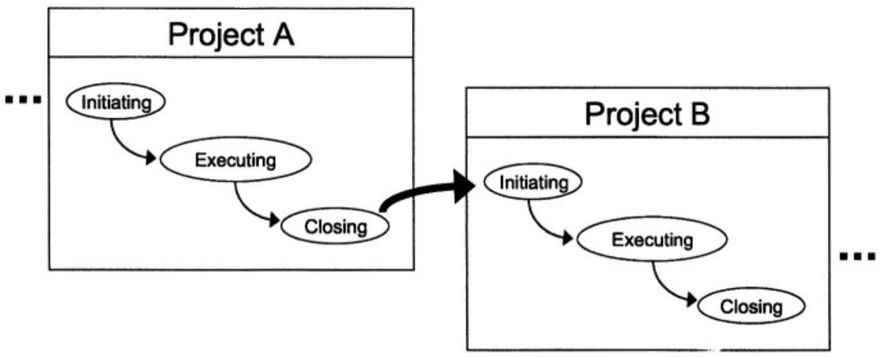


Figure 3.2 Interaction between projects.

3.3 INITIATING – LEARNING BEFORE DOING

At start-up – initiating a new project – it is important to take the necessary time to learn from others before we start. A simple way of doing this is to arrange a learning meeting or a workshop where the participants in earlier projects are invited to share their experience, insight and knowledge with the members of the new project.

Such a workshop – or peer assist – is useful when the costs of gathering the peers are small compared to the benefits of their contribution in resolving the challenges facing the new team or project.

Consider arranging a learning meeting assisted by peers outside the project group when:

- the multiple views from outside the project may contribute to evaluation of more solution alternatives,
- your project faces a difficult situation resembling something experienced in another project,
- you are new to the role and faces a challenge you know others have handled before, or
- it is a long time since you have accomplished a similar task, and you are unsure whether circumstances concerning the development process have changed or not.

Arrange the learning meeting early enough so that you can take advantage of the insight of others in your project.

Advantages with a learning meeting assisted by peers

- Gives assistance and insight from people outside the project.
- Identifies possible solutions and alternative work methods.
- Encourages learning across projects and teams.
- Develops a strong network between the employees in the company.

The word *learning* is important – the purpose of the meeting is to learn from others in order to contribute to the project, not to provide criticism of the project planning.

Another key word is *peers* – that the project members and persons are at the same level as you. The idea with a learning meeting is not to summon global experts or top management. It is all about finding people who "have done this before" and who come to share their insights with the new project team.

Both parties learn – the external peers acquire a broader knowledge base, while the new project team will be able to exploit earlier experiences. A learning meeting in the initiating phase will typically consist of eight steps:

1. *Clarify the purpose.* Define problem issues you want to address, and evaluate if a learning meeting is the most suitable way of getting help. Learning meetings function well when the purpose for and the expectations to the meeting have been established and communicated at an early stage.

2. *Decide the schedule.* Timing is important. Make therefore sure that the meeting is held so early that there is time to do something with the knowledge learnt from the others. In other words: If the learning meeting concludes with something unexpected, will you have time to do something about it? The point is to acquire increased insight, not to have your own ideas confirmed.

3. *Invite participants.* When the purpose is clarified you may identify and select potential participants for the learning meeting. Make sure you invite people with diverse knowledge and experience who may challenge you as well as provide new ways of thinking and new solution alternatives

4. *Get to know each other.* If the participants in the meeting are not familiar with each other, enough time must be reserved in the agenda so that people can get acquainted. This is important to make the group work as an open forum for experience sharing. One possibility is to have a half hour coffee break before the meeting starts. Or maybe there could be a dinner on the evening before?

5. *Provide context information.* The meeting should start with a presentation of the new project – the project team should tell what they know about context, history and plans.

6. *Ask questions and provide feedback.* The external assistants reflect over what they have heard, ask questions and discuss "what has surprised them" and "what they have not heard but have anticipated to hear something about"

7. *Analyze what you have learnt.* In this part of the meeting the project team analyses what they have learnt from the external assistants – what has work for others and how it worked.

8. *Summing up.* Sum up the meeting by focusing on the possibilities the project team has for alternative actions by combining what both parties know. Finally: Ask the assistants to reflect on what they have learnt that they can take with them from the meeting.

The following tools may be useful when going through the steps in the initiating phase (see chapter 7):

Mind Map, Brainstorming, KJ/Affinity Diagrams, Prioritizing, Action List, Root Cause Analysis.

As a summary of the initiating phase with its focus on learning from others before the project starts, we can use the 2x2 matrix in figure 3.3. By combining what I know in my context with what you know in your context, we can learn from each other and recognize what we both know. We are then in a position to work together to find out what possible improvements we can create through joint efforts, either by adapting practices to new contexts or by creating something new. Based on these opportunities we may act – and implement improvements.

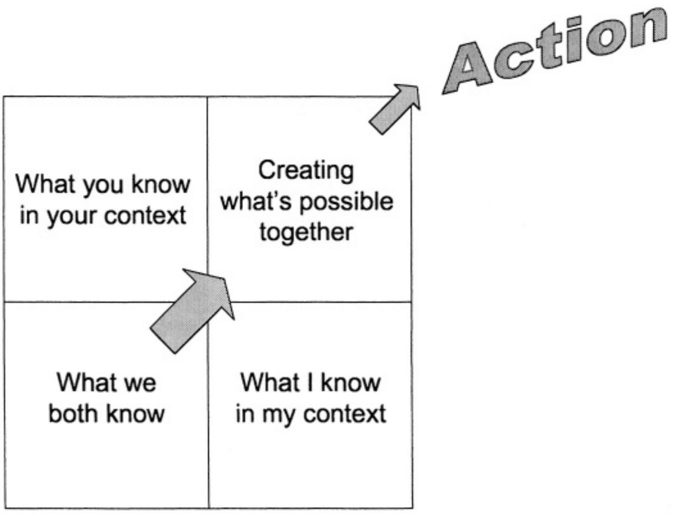

Action
What you know in your context
Creating what's possible together
What we both know
What I know in my context

Figure 3.3 Co-generative learning gives new possibilities for learning and action.

3.4 EXECUTING – LEARNING WHILST DOING

The core of the improvement work is learning whilst doing – it is not enough to remember the correct sequences or following the correct standards. Continuously changing environments and conditions often asks for adjustments and learning along the road implying the need for a large amount of creativity and improvisation. It is not sufficient to wait until the project is finished to draw conclusions – we need to do something *now.*

Short learning meetings following immediately after an activity or an occurrence makes it possible for us to learn from positive and negative experiences while we still are able to do something about them. Such continuous learning meetings are also called "After Action Reviews" or *debriefings.*

The point is that the continuous learning meetings are short and focused. They need not last for more than 15-20 minutes. Focus for these meetings are four simple questions.

Four simple questions to answer in an after action review

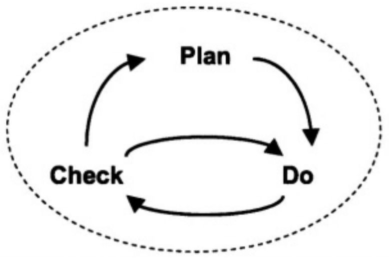
- What was supposed to happen?
- What actually happened?
- Why were there differences?
- What can we learn from this?

These questions are related to the three levels – *plan, do, check* – in the executing phase of the project – and can be used to carry out continuous improvement on all levels. These levels can span from a simple activity or process to iterations, releases and even to the project or company as a whole. The improvement steps include:

- *Plan.* Identify and define problem issues and improvement goals.
- *Do.* Carry out the planned changes your project.
- *Check.* Analyze and evaluate the results.

A last step in the traditional improvement cycle is *act* – describing the new or improved process and making use of it. In the model in figure 3.4 this corresponds to a new improvement level including planning, doing and checking.

Quite often there will be a short feedback loop between do and check where minor adjustments and improvised changes may be applied to the process without any planning up front.



Improvement levels:

- Activity/process
- Iteration
- Release
- Project
- Company

Figure 3.4 Improvement steps and levels in the implementation phase.

A learning meeting in the executing phase will typically consist of the following five steps:

1. *Arrange the meeting at once.* The purpose of the meeting is debriefing – instant learning while the event is still fresh in mind. In this way the new insight may come into use at once – may be the next day.

2. *Ask: What was supposed to happen?* Start the meeting by discussing the purpose of the activity, what the result should have been, if it was measurable, if everyone agreed or if there were different opinions of what should have happened.

3. *Ask: What actually happened?* Go through the actual results. Focus on measurable facts – not vague presumptions. Still we need to recognize the fact that there may be difference in understanding and interpretation of what actually happened.

4. *Compare the planned with the executed.* The real learning begins when the team compares the plan with what actually happened. Why were there differences? Did we do better or worse than expected? What contributed to the success, or eventually, what caused the problems?

5. *Summarize the meeting.* What have we learnt? Discuss the most important knowledge achieved from the activity and what to do with it. Such knowledge is often team specific and context dependent. Take good care of it – and do something with it.

Some rules for After Action Reviews (AARs)

- Plan it – include an AAR in the planning of all activities.
- Arrange an AAR immediately after an event – while it still is fresh in mind.
- All the involved persons participate – leaders and employees on equal footing.
- Learning – not blame or assessment.
- Open and honest attitudes – nothing must be kept secret.

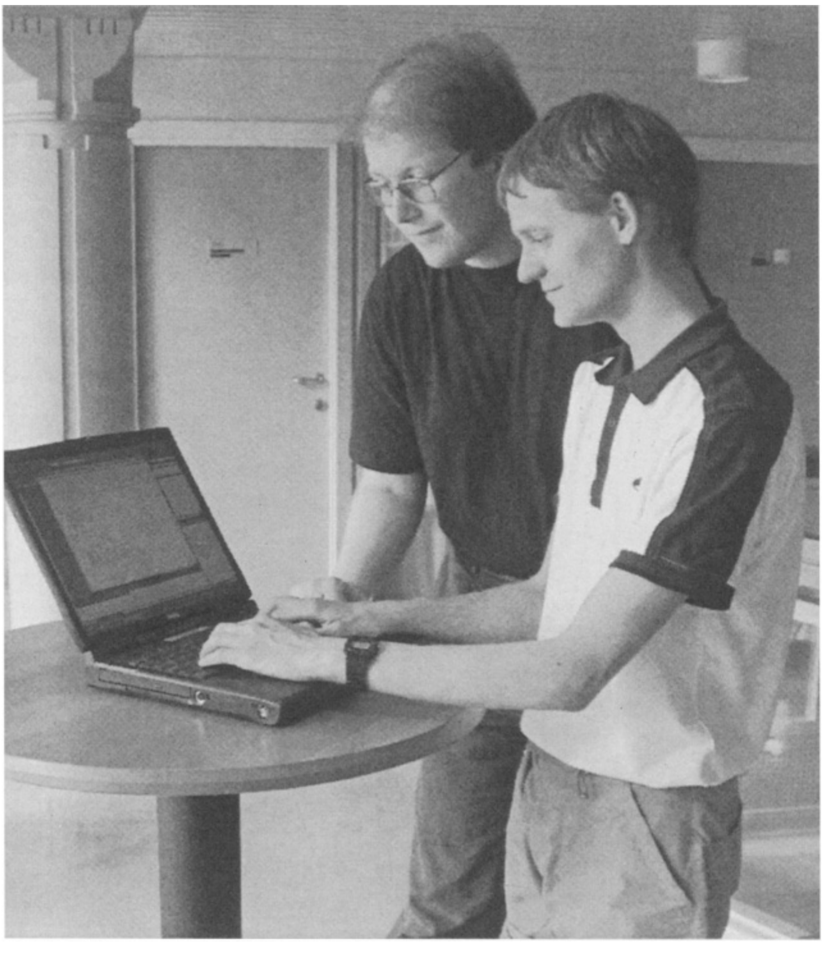


Figure 3.5 A short learning meeting in a corridor.

The following tools may be useful when going through the steps in the executing phase (see chapter 7):

Mind Map, Brainstorming, KJ/Affinity Diagrams, Prioritizing, Action List, Root Cause Analysis.

As a summary of the executing phase with its focus on learning whilst doing during project progress, it is possible to use the four questions from the learning meeting as a checklist.

What was supposed to happen?

- What was the purpose of the activity?
- Could it be measured?
- Did everyone agree?
- Were there different opinions on what should have happened?

What actually happened?

- What was the result?
- Could it be measured?
- Did everyone agree?
- Were there different opinions on what actually happened?

Why were there differences?

- Were there differences between what was planned and what was executed?
- Did we do better or worse than expected?
- What contributed to the success, or eventually, what caused the problems?

What have we learnt?

- What is the most important knowledge achieved from the activity?
- What should we do with this knowledge?
- How could we do something about it?

In the next chapter on measurement and feedback, we will take a closer look at how to use measurements during the project as a further supplement to the learning process.

3.5 CLOSING – LEARNING AFTER DOING

When the project, or the larger part of it, is finished, it is time to sit down and reflect on what happened - in order to get a better understanding and make our understanding available to others. The Danish philosopher Søren Kierkegaard said: “Life is lived forwardly, but it is understood backwardly.”

In the project closure phase we combine our collected experiences from the project in the form of new or updated models and processes, or as other forms of knowledge, and make them available for future projects.

The fact is, however, that such experience reports are not always read. There may be many reasons for that. One of the reasons seems to be that the experience reports are not written with the future reader in mind – they are more or less written as a duty task that must be performed in order to finish the project.

A simple way of doing something about this is to arrange a learning meeting when the project is finished, a so-called post mortem analysis (PMA). Arranging such learning meetings when the project is finished is a quick and effective way of collecting knowledge and experience before the project is dissolved. We also make sure that future teams may use the knowledge that the recently finished project has acquired. A PMA may also be used for immediate transfer of knowledge to a new project which is about to start.

PMA – post mortem analysis is all about learning from project experiences – good ones as well as bad ones

- What did we do successfully and would like to repeat?
- What useful things did we do, but with the need to be improved?
- What foolish things did we do that should not be repeated?
- What were the causes for something turning out to be foolish or a success, and what can we do about it?

A PMA is more through than the continuous learning meetings. In stead of lasting only 15-20 minutes, they typically last from a couple of hours to a whole day. The PMA is also different by focusing on collecting experience and knowledge for future projects – not only on learning in action, inside a running project.

Typical agenda for a post mortem analysis

- *Introduction*
  Clarify the agenda and the expectations
  Who are we, what are we going to do and why?
- *Brainstorming/KJ*
  What went well in the project?
  Presentation of experiences using stickers
  Structuring and prioritizing of experiences
- *Root cause analysis*
  Why were these aspects a success?
  Organizing and prioritizing of causes using fishbone diagrams
- *Brainstorming/KJ*
  What did not go so well in the project?
  Presentation of experiences using stickers
  Structuring and prioritizing of experiences
- *Root cause analysis*
  Why were these aspects not a success?
  Organizing and prioritizing of causes using fishbone diagrams
- *Summing up*
  Summing up of the seminar and further plans
  Priority of actions

A post mortem analysis typically consists of the following steps:

1. *Summon a meeting*. PMA-meetings are held face-to-face and should follow immediately after the project closing – it may very well be combined with a celebration of the finished project. The main rule is mat people meeting each other face-to-face is by far the most effective, but a video conference is better than nothing. Do not, however, try to arrange a PMA by using e-mail.

2. *Invite participants.* If a similar project is about to start or has already started, the members of the new project may benefit largely form participating in the PMA. In this way we may ensure real-time transfer of knowledge. The project leader and other key personnel from the project must be present. Ideally speaking the customer should also participate– at least in the first one. The importance of such customer attendance must be considered and weight up against the need for having an open atmosphere. Not everyone feels comfortable speaking out openly in front of a customer.

3. *Select a chairman for the meeting.* It may be wise to use a chairman from outside the project. In this way it may become easier to focus on "what should a new project team do" instead of making the meeting a more or less unreflected review of "what we did".

4. *Go over project purposes and project delivery.* Start the meeting with a short review of "what should have been done" and "what was actually done". Also pay attention to whether time limits were kept or not, and if the quality of the delivery met with the expectations. If the customer is there, ask: "Did you get what you had expected?"

5. *Ask: "What went well in the project?"* Always start with the positive. We all want to use our successes as a foundation for future work and avoid repeating mistakes. Ask: "What were the successful steps to realize the purposes?" "What in the project was really a success?"

6. *Ask: "What caused the success?"* Identify the success factors so that they can be repeated in the future. Try sticking to the facts. Opinions may be good, but recommendations to future projects should as far as possible be based on facts that the projects members can agree on. Focus on specific advice that can easily be implemented into new projects. If your are running out of time, ask: "What was the most important success factor?"

7. *Ask: "What did not go so well in the project?"* There certainly are areas where the project could have been more successful, where pitfalls were identified too late and where the process was anything but optimal. Ask: "What could have gone better?" Do not allow any sort of blaming to take place, rather encourage people to say: "My opinion of what happened is different." Remember that all opinions are just as valuable.

8. *Find out what theproblems really were.* Identify pitfalls and obstacles so that they can be avoided in the future. Ask: "What were the causes for the less successful parts?" "Given our present knowledge, what could we

have done better? What advise would you give a future project team based on the experience from this project?”

9. *Identify actions.* If a project team is about to start a new project similar to the one they just have finished, it could be useful to follow up the PMA with a start-up meeting for the new project. In this way we can close the learning loop. Action is important. Without action – no improvement. It is therefore important to identify actions, for example to incorporate the new knowledge into updated processes, procedures, checklists and models.

10. *Document the experiences.* It is important to make a good summary from the closing learning meeting in the project. Moments to consider are: advice and hints, experiences, causalities, actions, illustrating examples, name of project members and references to documents, models, guidelines, checklists and key personnel. It is a good idea to use quotes of what has been said to make the presentation livelier. The acid test on whether the summary is useful or not, is to ask yourself: “If I were a project leader for the next project, would this summary be useful to me?” An example of an experience package is shown in figure 2.6 in chapter 2.

What is the output from a PMA?

The main result from a PMA is more knowledge about the process that has been carried out – knowledge that can be reused as

- hints and experience notes in a knowledge base,
- new and improved checklists,
- updated and improved development model,
- a better understanding for the participants of what went well and what could have gone better.

The following tools and techniques may be useful during the activities in the project closure phase (see chapter 7):

Mind Map, Brainstorming, KJ/Affinity Diagram, Prioritizing, Action List, Time Line, Root Cause Analysis.

As a summary of the project closure phase with it's focus on after-project learning, we will take a closer look at what to do to turn the new knowledge into practice. Knowledge and experience are generated from the development processes – and must be returned to the development process in order to accomplish improvements.

In order to make this happen, the knowledge needs to pass through a life cycle consisting of several steps.

The knowledge life cycle

- *Identify* the knowledge and experience through dialogue and reflection.
- *Analyze* the knowledge and make experience packages.
- *Integrate* the experience into the company's development model.
- *Use* the new knowledge actively to improve the processes and the products.

This page intentionally left blank

Chapter 4

MEASUREMENT AND FEEDBACK

Goal oriented measurement is a key technique for systematic improvement and an important element in an organization's learning process. Such measurements may uncover underlying causes for problems in the development projects, contribute to a better understanding of the development process and be supportive during the implementation of software intensive projects.

The success of the company is to a large extent depending on successful accomplishments of the company's projects. Measurements on the project level are therefore the most important ones, in order to help the project leaders and project members to do a better job. Measurements contribute to more realistic plans and to better follow-up of the projects.

Measurements are, however, no guarantees that a project will be successful. On the other hand, they may help in making the right decisions and implementing the right actions. In this way measurements are helpful both for the project and for the organization in general.

Even if the measurements are useful to the project, there will also be an information need at company level that may benefit from the measurements. In most cases the base for this information comes from the projects. In other words: you need data of good quality from the projects to analyze the company as a whole.

Measurement on company level may then be looked upon as a combination and analysis of valid project data for all the company's projects.

In this chapter we describe a model for measurement based improvement that can be used as an integrated part of the learning process during the projects. It is defined in such a way that it should be possible to adapt it according to the individual projects' special characteristics and environment. In this way both the measurements and the process of getting the measurements done can be adapted to the individual needs for the project and for the company.

Prerequisites for successful measurement programs

- The scope of the measurement program must be improvement.
- Employees must not feel that they are being controlled by the measurements.
- Employees must participate in all phased of the measurement program.
- Measurements must be integrated seamlessly with the ordinary project activities.
- Management must commit themselves to actively support the changes that are suggested as a consequence of the measurement results.

4.1 PROCESS FOR MEASUREMENT AND FEEDBACK

Underlying the various methods of using data for systematic improvement is an implicit model of the process for using data. In its simplest form, any such method involves a means for collecting data, an approach for analyzing and interpreting the data, and a method for feeding the data back to relevant individuals and groups. While approaches on how to perform each of these activities vary, all of them assume some sequence of *collection, analysis,* and *feedback* will occur, as they are the core activities of any measurement-based method for systematic improvement.

In addition to these core activities there are also other activities that should be included: Before the collection of data starts, the organization should have a clear understanding of what data to collect, why they should be collected, how they should be analyzed and how they will be fed back. Thus, the measurement process should include a *planning* activity related to the use of data.

Most importantly, the degree to which individuals or groups receiving feedback use the data for problem-solving issues or purpose fulfillment, determines the ultimate effect of the data on improvement. Therefore, another activity of the measurement process should be added as *follow-up* to the actual feedback of data.

The measurement and feedback process therefore includes five activities (see figure 4.1):

1. *Planning to use data* – establish a measurement plan that should provide an answer to "what to examine", which data it will be relevant to collect and how the results should be analyzed.

2. *Collecting data* – collect all the data described in the measurement plan.

3. *Analyzing data* – organize the data that has been collected in such a way that they are well suited for presentation and interpretation.

4. *Feeding back data* – interpret measurement data and experiences, compare with the improvement goals, check that they are valid, and suggest improvement alternatives.

5. *Following up* – start and implement actions identified in feedback meetings in order to make improvements take place.

As a result of the follow-up activities it may be relevant to collect further data. If that is the case, the cycle starts over again with a planning activity before the new data are collected.

These five activities make up a process for carrying out measurement based improvements. The process is general and can be adapted to different situations and applied at different organizational levels – not only to the project level.

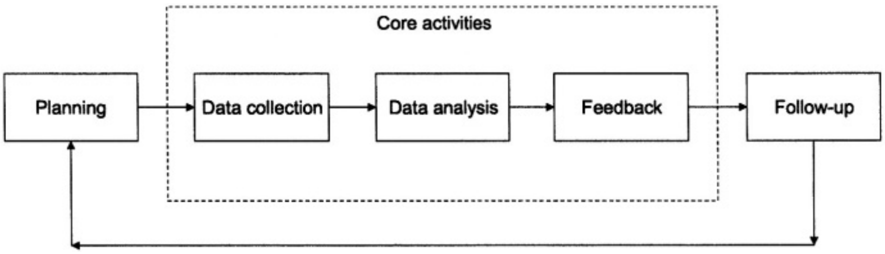


Figure 4.1 Process for measurement and feedback

4.2 PLANNING TO USE DATA

Many attempts to use data for systematic improvement fail before the first form has been filled in, before the first interview has been carried out or before the first observation has been made.

They fail because the organization has not prepared the measurement and feedback activities properly. They failed because it has not been properly clarified what data to collect, how to used them and why they should be collected in the first place.

Good and early planning is necessary if the measurements shall be of any use in the improvement project. As we will show later, the largest added value comes from the feedback and follow-up parts of the process. So it is important to realize that early decisions, such as what data to collect, how to collect them, how to analyze them, and how to act upon them will have influence on the feedback process.

In other words: Feedback will not contribute to improvement if the wrong data are collected, if analysis is not done correctly or the employees in the organization do not believe that the data will be used in a constructive way.

The purpose of the planning activities is therefore to make a plan for the actual measurements and for the feedback activities. The measurement plan, like all other plans, is an operative *action plan.* It should therefore provide an answer to "what to examine", what data to collect, and how the results should be analyzed and fed back.

Hints

- Involve the project members in the work of identifying what measurement data should be collected.
- Do not use the collected data to measure individuals.
- Create an atmosphere of confidence – to the collection process and the people responsible for collecting the data.
- Use the data for something useful – avoid "information graveyards".

Based on the project plan and the projects initiating learning meeting, and during the further work with measurement activities in the project, we may very well use GQM (Goal-Question-Metric). In short GQM is about defining specific improvement goals, ask concrete questions, define metrics, collect measurement data, answer questions and draw conclusions on whether the goals were reached or not.

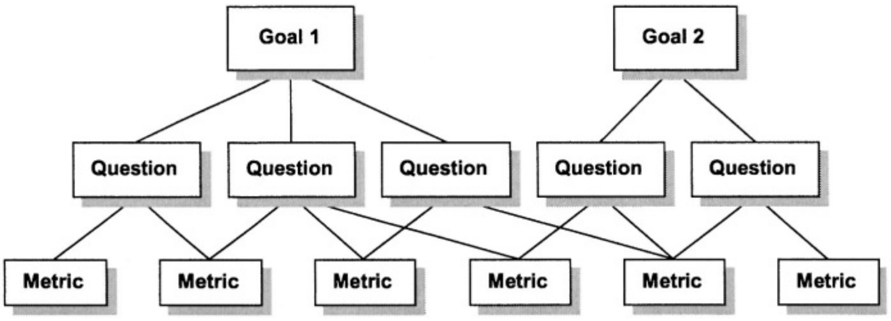


Figure 4.2 GQM model.

A GQM model contains information necessary to implement goal-oriented measurement and analysis. The elements in the model are goals, questions and metrics (see figure 4.2 and examples in table 4.1):

- *A goal* specifies an analytical task that should be achieved using measurement:

  "*Analyze the* <object> *for the purpose of* <purpose> *with respect to* <quality focus> *from the viewpoint of* <viewpoint> *in the following environment:* <environment>"

GQM's goal definition includes five elements

- *Object* – what is the object of the analysis?
- *Purpose* – what is the purpose of the analysis?
- *Quality Focus* – which quality aspects of the objects shall the analysis focus on?
- *Viewpoint* – from what viewpoint shall the quality focus be analyzed?
- *Environment* – in which environment shall the analysis take place?

- *A question* expresses a need for information. The answers should contribute to reaching the goal. The form of the questions depends on what we want to achieve. We need information both on quality focus – what we want to know something about – and on the environment – for later interpretation and reuse.

- *A metric* is a quantitative specification of data or information. Metrics are derived from the questions and contribute the data required for answering them. One or more metrics are needed for each question.

The planning activity typically consists of the following steps:

1. *Define the improvement goals.* Define specific improvement goals and expectations to the results based on the knowledge gained after the first learning meeting in the project's initiating phase. There are two types of goals: *knowledge goals* focusing on better understanding of the development process and the environment it is applied in, and *change goals* focusing on implementation and evaluation of the effect of specific changes, for example the effect of a new test tool.

2. *Describe the development project.* Describe the development project and its environment. In this way future projects may compare their context with this project's to consider the relevance of the results.

3. *Identify the evaluation method.* Identify the method to be used to evaluate the result of the improvement project. Three methods are commonly used: comparison with the company's baseline, comparison with a sister project or comparison of components within the same project.

4. *Make a measurement plan.* Make a measurement plan for the project describing the improvement goal, what questions to investigate, what data to collect, when to collect them, who should collect them and how the results should be analyzed.

Table 4.1 Example of using GQM.

Goal	Question	Metric
Analyze the development process *for the purpose of* understanding *from the viewpoint of* the software team *in* project X.	Q_1: What is the failure rate before delivery?	M_1: #failures found during system test M_2: Size of system test
	Q_2: What is the distribution of failures before delivery by severity?	M_3: Failure categories

Important information in the measurement plan

- What is the improvement goal?
- What measures should be tried?
- What data should be collected?
- How should the data be collected?
- When should the data be collected?
- Who should collect the data?
- How should the data be analyzed?
- How should the data be used?
- Who should have access to the data?

The following tools may be useful when going through the steps in the planning phase:

Brainstorming, KJ/Affinity Diagram, GQM Abstraction Sheet.

4.3 DATA COLLECTION

The project members should primarily concentrate on the development project. It is therefore important that as little time a possible is spent on data collection. We also need to make sure that we avoid collecting unnecessary data, particularly when it takes place manually.

This statement has from time to time been overheard: "When we are collecting data from area X, we might as well collect data from area Y, it does not make any difference" We strongly warn against such an attitude. Even if it does not take much extra work to collect the data, we always will use some, and the organization rarely used these additional data.

If we collect data not used for anything, the participants will soon discover that the importance of the collected data is relatively small. This can influence the attitude towards the rest of the data collection. Such a contamination of attitude will only happen when the data is collected manually – not when automatic collection is used.

When data is collected we must remember that the purpose is to improve the process. We are not supposed to control individuals. If we collect data on individuals in order to measure their output, we must expect to get the answers we deserve – made-up data that are useful for the individuals.

In some cases we need to collect data on individuals to be able to estimate total values or average values, such as the number of faults discovered in the code modules the person is responsible for or productivity data for their part of the system. In such cases we must be careful to organize the collection process in such a way that the individuals are fully confident that the data will not be used to their disadvantage at the next wage negotiations. If the involved persons have the slightest suspicion that the collected data might be used against them, there is a great chance that they will produce data for their own benefit rather than the correct data.

The collected measurement data must be used for something that the participants look upon as useful. If persons have been asked to collect measurement data and they experience the nothing comes out of it, they will eventually not bother about the quality of the delivered data.

They will of course deliver the required data, in order to satisfy the boss, but will not put any effort into it and some data may in the worst case be made up.

The data collection activity has two basic steps:

1. *Collect data.* The first step is to collect data according to the measurement plan in the project. The collection is implemented throughout the entire project and may involve a lot of persons in the company. Data can be measured objectively or be based on subjective evaluation.

   Mainly there are two ways to collect data: *automatic* and *manual.* We should, if possible, use tools for data collection. We may then ensure that the data collection is done in a consistent way, and that the results may be compared. Examples of tools for automatic data collection include static and dynamic analysis tools, configuration management tools and project management tools. Manual collection of measurement data is usually carried out by using forms. Typical forms are fault reports and timesheets.

   Automatic data collection is to be preferred, but is often impossible. The solution will then be manual collection using forms. The project members should have as few forms as possible to relate to. The forms must be simple and at the same time hold the information necessary to fill them in correctly. If ranking or classification relative to a measurement scale is used, the scale must be defined in a way that ensures a common understanding of the different alternatives. General data like project name should be filled in before the forms are handed out to reduce the amount of work for the participants. The same form may be used to collect data for several metrics. Forms may be electronic or on paper. Main types of data are shown in Table 4.2.

Hints

- Use existing form if such are present
- Expand existing forms if such forms exist to collect data on the same object
- Use as few forms as possible
- The forms must be easy to fill in
- The forms must hold all information necessary to understand how to fill it in
- General data should be filled in before the forms are handed out
- The fields in the forms must indicate what unit and accuracy level to be used
- Always include fields for comments

2. *Validate data.* Check that single values and aggregated values are reasonable as soon as they are available.

   Validation of measurement data is important because we need accurate and consistent data, but it is difficult because it must often be based on judgment. The best advice we can offer is that the persons responsible for validation must use their own experience to evaluate whether the data is reasonable or not.

   Data should be validated as soon as they are available. At a later point it may be impossible to correct the data, or at least be difficult.

   Validation of the measurement data is particularly difficult in the start-up period, before all the members have achieved a common understanding of the metrics.

   When we validate the measurement data, the first point to check is whether the data are reasonable when related to the definition of the metric. This check requires a good understanding of the content and intent of the metrics and of the environment in which the data were collected. If we have used a metric defining “work-hours spent on system testing” we first need to check that all timesheets have been collected and included in the estimate for the total. Next we take a look at the total figures to ensure they are reasonable. This may for instance be done by having a look at the estimated work-hours for system testing, the length of the test period combined with the personal experiences from system testing. Personal experience is important for the person validating the measurement data.

The forms are validated by looking through the fields that are filled in, checking single values and in some cases generating plots to see if certain data patterns are occurring repeatedly. Such patterns may indicate that parts of the data incorrectly "survive" from one data set to another. One example is that several consecutive fault reports may have the same date and refer to the same system. This may indicate that the person who filled in the report copied an old report and forgot to update some of the fields.

If data values – single ones as well as aggregated ones – are outside what is expected, we should try to find out why. Are there special circumstances that have caused the specific values, or is there a fault in the data collection? We should also check for consistency between data values in the different fields. If the result from one person deviates significantly from the other's, we should investigate if the person has misunderstood how to use the form, and for example thought that the extreme value "5" represents the best value instead of the worst.

The following tools and techniques may be useful when implementing the steps in the data collection phase:

Spreadsheet with Statistical Module, Simple Database.

Table 4.2 Main types of measurement data.

Data that may be plentiful	**Data that should be sparse**
• *Project start-up data* This is data that is collected once – during start-up of the project. Typical start-up data is estimated work-hours and duration of the project, the structure and experience of the project team, project characterization and environment. • *Project closure data* This is also data that is collected once – when the project is closing. Typical data are used resources that should be compared with the estimated data from the start-up.	• *Periodic data* This is data that is collected periodically, for example every week during the entire project period. Typical data are timesheets delivered every week. • *Event-based data* This is data collected when special events occur, for example when a failure report form a customer arrives, or when a project phase is closed. Typical data are fault and change reports, inspection data and resources used for a phase or an activity.

4.4 DATA ANALYSIS

Our experience is that it is always a good idea to start by plotting the data before further analysis is done. A plot may for example indicate if one or several measures do not fit into the general pattern. If we use a spreadsheet to record the data, a graphical presentation can easily be made. Spreadsheet packages such as Excel also include tools for some simple statistical analysis.

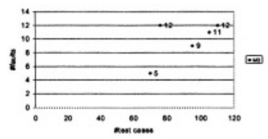
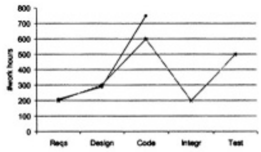
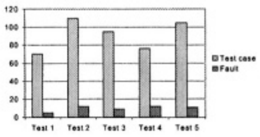
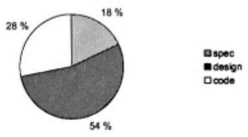
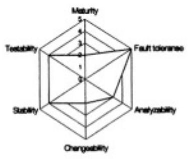
Plotting techniques can also be used for a first simple analysis of the data. We may look for trends and non-typical values in our data set. Often, this analysis will be a sufficient basis for decision-making. Some times, however, the first analysis must be followed by more extensive analysis in order to decide whether the preliminary interpretations are correct or not. The result may also be that it is not possible to make any decision based on the collected data.

This may be caused by several factors. Our data could be too sparse or not good enough, the metrics could have been incorrectly defined, etc. In such cases we may collect more data or drop the actual metric in the further analysis, or we may define new metrics in order to collect other data. New metrics may be added to – or replace – the data we have already collected.

Table 4.3 gives and overview of the plotting techniques most frequently used, along with some typical applications. The main difference between the different plotting techniques is whether the data related to the same metrics should be collected several times to follow the development over some time, or if we should look at properties separately or related to each other. Which plotting technique to select depends on the goal of the analysis. Do we, for example, want to look at the data scattering, or do we want to find non-typical data. The diagrams overlap in application area, and it may be a matter of taste what to choose.

Each diagram will typically give the answer to one question in the measurement plan.

Table 4.3 Overview of the most used plotting techniques.

Plotting technique	**Typical application**
Scatter plot 	• Relationship between resource usage and code volume • Relationship between #fault and code volume • Relationship between planned and actual use of resources • Relationship between #persons in the project and time spent on project management
Time diagram 	• Follow-up of resources used during calendar time or phase • Follow up of faults found in each phase • Follow-up related to plans or estimates, e.g. actual versus planned use of work-hours, #faults found versus estimated number or faults etc.
Histogram 	• Relationship between #faults found and hours of testing • Relationship between test volume and #faults found • Productivity per phase compared with the company's baseline • Types of faults compared with the company's baseline
Pie chart 	• Resource usage per phase • Faults distributed over type of test, for example unit test, integration test, system test, factory acceptance test (FAT), site acceptance test (SAT) • Faults distributed over introduction phase • Faults distributed over type of faults
Kiviat diagram 	• Characterization of properties, one property per axis. The properties may be connected to software modules, projects, products, processes, etc. • Visualize the results from surveys

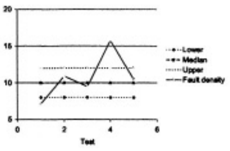
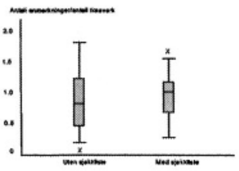
Plotting technique	Typical application
Control diagram 	• Control that the defect density is within specified range • Control that the time spent on non-planned activities does not exceed specified limits • Control that the productivity does not deviate from what has been the average values in the company
Box plot 	• Study productivity variations in several projects • Study variation in defect density in several modules • Compare the distribution of defect density in several modules in several projects • Compare the number of remarks in quality reviews in two or more projects

Figure 4.3 shows an example of measurement data from a company. The question to answer is: What are the costs of software inspections? The costs are measured in number of work-hours per document page. The costs include preparations, meetings and summing up of the results.

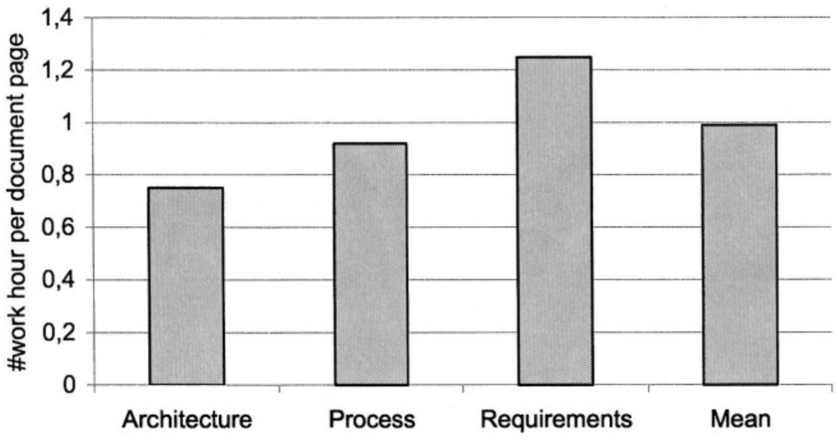


Figure 4.3 What are the costs of software inspections?

Our experience is that in an improvement project – while we are still most occupied with getting an overview – use of histograms is the most commonly used technique. Later, when looking at relationships, we will use several of the other plotting techniques. As the amount of data increases, it is also possible to use simple statistical methods.

Simple statistical methods

The question: "Is there any difference?" is a recurring question in all improvement work. It is asked in slightly different forms depending on the situation. We will take a closer look on some usual cases:

- *Linear regression*: Is there any trend in the number of faults found during inspection when increasing the time used for preparations? If we plot the corresponding data pairs – preparation time and number of faults – in a scatter plot we may get a certain idea of possible relationships. By implementing a regression analysis we will be able to see if this is a possible trend, or if it is just random.
- *T-test*: Do we, on the average, find more faults in the inspections when we use checklists? We may start with the number of faults found per page per spent work-hour etc. The number of fault may vary regardless of whether checklists were used or not, due to factors like complexity and the skills and interests of the developers. What we can establish by using a statistic method, in this case the t-test, is if there are differences that cannot very likely be explained by the natural variation in the data we have collected.
- *ANOVA*: Is there any difference in the number of states in the module descriptions that the developers hold as simple, and the ones that are held as complex, or do the number of states vary so much within both categories that any systematic difference are drowned in "noise". One way to answer this question is to classify the entire data set depending on whether they come from simple or complex modules, and find out how much of the variation can be removed from the data in this way.
- *Chi square test*: Is the number of faults distributed over a set of fault categories identical for several projects? If we collect data from several projects we will most likely not get the same distribution of faults on the different categories every time. The question that needs to be answered statistically is how large the difference may be to be able to state that the differences we find are just random.

For all these methods there are a lot of data support available – from simple spreadsheets like Excel, via smaller statistical packages like Minitab to heavier packages like SPSS and SAS. Whatsoever – begin with what you think and use the data analysis to find out if what you think is true.

Strictly speaking there are a lot of and mathematically complicated assumptions behind a stringent statistical analysis. Practical experience, however, shows that we will be OK even if we do not meet all these. Just be aware that the more assumptions you doubt to be fulfilled, the more careful you must be when you interpret and use the results.

The following tools and techniques may be useful when implementing the data analysis phase:

Spreadsheet with Statistical Module, Simple Database.

4.5 FEEDBACK

The project members must regularly be given feedback. Feedback meetings are an important tool. Several organizations regard such meetings to be the most important means for a successful measurement program. The purpose of a feedback meeting is to present the data for those who collected them in order to get their interpretations. By combining the participants' knowledge and understanding of the development process with the collected data, we will often be able to make better decisions based on fewer data than if we had only based the decision on a pure statistical analysis.

As the improvement work to a large extent is a learning process, we must not expect too much from the first meeting. We will often need the two or three first meetings to validate and adjust the data collection, but too much time must not be spent before we draw the first conclusions.

The smaller process adjustments that are implemented during the improvement project will often give results in the short run, and have proved to be of great importance. They are important to help the participants keep up the spirit and also to satisfy impatient managers that often are in doubt.

Feedback meetings must also be viewed related to the total learning process in the company. An important result from the feedback meeting is therefore to be able to give an answer to the following question: "What experiences may be transformed from these meetings to the company's collective memory?

Feedback meetings should be managed by the person who is responsible for the improvement project. In these meetings, metrics and questions from the measurement plan are presented with the corresponding data, but no conclusions are drawn. A discussion will follow where the project members interpret the data. This is a creative process which should not be too harshly controlled. The results from the meeting should be summed up and sent to all the participants when the meeting is finished.

Typical steps in a feedback meeting are:

1. *Interpret the data.* Look for non-typical values caused by special situations, remove them from the data set and put them into a separate data set for exceptional analysis. Look for trends in the data, but remember that the plots only give indications. Make sure that apparent trends are followed by statistical analysis before the final conclusions are drawn. Are the values as expected? Unexpected values may indicate that the procedures for data collection were not good enough, as for example that not everybody have collected data in the same way. It may also mean that the assumptions of what is going on are wrong and that they should be corrected.

2. *Answer the questions in the measurement plan.* After having made the interpretation of the measurement data, we should try to answer the questions in the measurement plan. Are the answers as expected? Unexpected answers may have natural explanations linked to non-typical events in the period the data were collected, but may also indicate that one or several of the hypotheses from the measurement plan must be rejected.

3. *Suggest changes to the measurement plan.* As a result of the previous steps we may have found that the measurement plan should be changed. Such changes may for example involve defining new metrics for some of the questions, defining new hypothesis or updating the metric definitions.

4. *Suggest improvement actions.* In feedback meetings suggestions for improvement will usually turn up. Some of these may already be relevant in the on-going improvement project, for example adjustments to the project plan or to the development process. Suggestions may also be made for more long-term improvements. We will take a closer look at them during the post mortem analysis of the project.

The following tools may be useful when going through the steps in the planning phase:

Brainstorming, KJ/Affinity Diagram, GQM Abstraction Sheet, Spreadsheet including Statistical Module, Simple Database.

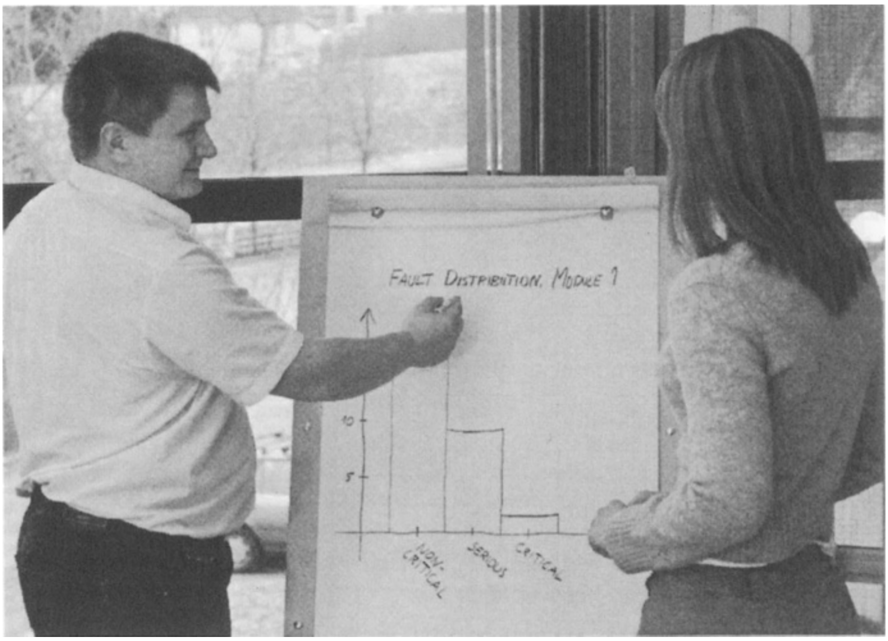


Figure 4.4 Feedback meeting.

4.6 FOLLOW-UP

Follow-up primarily consists of summing up the experiences, drawing conclusions concerning goal achievement in the improvement project and initiating long-term improvement actions identified in the feedback meetings.

In projects with several improvement goals, many questions, or large amounts of data, it may be wise to have a separate feedback meeting to assess goal achievement. Alternatively this can be done as a part of the final PMA in the project.

The follow-up activity typically consists of the following steps:

1. *Post project analysis.* Make a post project analysis of all data collected in the improvement project. Use the collected data to answer the questions in the measurement plan. Make specific suggestions for improvement actions for later projects.

Purpose of the post project analysis

- Assess whether the improvement goals were achieved or not.
- Follow-up the improvement actions initiated during the project.
- Identify suggestions for future improvement actions.
- Analyze strong and weak aspects of the execution of the improvement project.

2. *Package experiences.* Assemble the results from the post analysis and document the experiences in such a way that they are made available for future projects (see section 3.4 on post mortem analysis).

The following tools may be useful when going through the steps in the evaluation phase:

Spreadsheet including Statistical Module, Simple Database, Publishing Tools, Feedback Meeting, Prioritized Action List, Root Cause Analysis.

Chapter 5

PROCESS ASSESSMENT

When you start working with improvement, you need to know the state of your organization's current software processes and the goals for the future. You also need to know whether you have reached your goals or not when the planned improvement activities are finished.

There are a number of ways to determine the state of the organization's current software processes, for example your gut feeling, interviews, or the opinion of a majority of the employees in the company or of a particular group. A lot of companies would, however, prefer a more formalized assessment before starting with extensive improvement actions. In this chapter will take a closer look at a method for such assessment that has been used successfully by a number of software organizations.

5.1 WHY PERFORM AN ASSESSMENT?

Not all software companies are equally skilled at identifying the causes of their problems or to identify the most rewarding opportunities for future competition. Without a preliminary problem analysis, "solutions" are seldom effective; on the contrary, they are often irrelevant to the underlying causes of the symptoms that are being treated and only add more noise to the system. Consequently, it is of utmost importance that complex problems in software development must be thoroughly understood before a solution is attempted.

In many cases, process assessment can help software organisations improve themselves by identifying their critical problems and establishing improvement priorities before attempting a solution. The objective is to identify and prioritize problem factors as early as possible. Therefore, the main reasons to perform a process assessment are:

- To understand and determine the organisation's current software engineering practices and to learn how the organisation works.
- To identify and prioritize strengths, major weaknesses and key areas for software process improvement.
- Find information on business environment issues that may influence the process improvement work.
- To facilitate the initiation and planning of process improvement activities, and enrol opinion leaders in the change process.
- To help obtain sponsorship and support for actions through following a participative approach to the assessment.

As we will see later, the last point – a participative approach – is crucial for a successful process assessment as well as for successful implementation of the improvement activities.

The time schedule for the assessment process usually depends on the relation between the need to have broad participation in the organization and the need to keep up the motivation and get quick results. The size of the company will also be decisive for the possible speed on involvement for all the relevant parties in the process.

5.2 BASIC PRINCIPLES

There are three ways in which a software organization can make an assessment of its development practices:

- Benchmark against other organizations.
- Benchmark against "best practice" models.
- Assessment guided by the individual goals and needs of the organization.

Benchmark against other organizations. This way of doing an assessment is a traditional benchmark exercise used to gain an outside perspective on practices and to borrow or "steal" ideas from best-in-class companies. This type of benchmarking can be characterized as an ongoing investigation and learning experience that ensures that best practices are uncovered, analyzed, adopted, and implemented.

Traditional benchmarking is often a time-consuming and disciplined process that involves a thorough search to identify best practice companies, a careful study of one's own practices and performance, systematic site visits and interviews, analysis of results, development of recommendations, and finally, and most importantly, implementation of change.

Benchmark against "best practice" models. The second way of performing an assessment is to benchmark the company against one or more of the so-called "best practice" models on the market. Over the years, several assessment models have emerged in the software industry, and there is a range of possible assessment models that one can choose from. Examples of such models include CMM, CMMI, ISO/IEC 15504, ISO 9001, TickIT, The European Quality Price, Bootstrap and Trillium, just to mention some of the most important.

The various models focus on different aspects of the software processes and the organisation, and they are all associated with specific strengths and weaknesses. However, they share a common set of problems; they are often based of idealized and unvalidated practices, intended for large companies, which are not always relevant for small and medium-sized companies.

Furthermore, most of these models also emphasize an improvement approach based on statistical process control (SPC), which is a rather dubious strategy for the majority of organizations doing software or product development (also see the section on measurements).

Assessment guided by the organization's needs and goals. This way of performing an assessment emphasizes the uniqueness of each organization and how this uniqueness can be used as a competitive advantage. This method is participation oriented and considerably quicker to implement than benchmarking – both traditional and model based. Furthermore, the method may easily be tailored to the specific needs of an organization, and will therefore also be more relevant and valid than the model-based approach.

Benchmarking – both traditional and model based – have been thoroughly described in other books. In this section we will therefore describe our experiences in performing a participative approach to software process assessment tailored to the organizations' specific needs. First we will, however, take a closer look at the principles of gap analysis.

5.3 GAP ANALYSIS

Independent of the method we use for process assessment, it will always use some sort of gap analysis to compare current practices with where we want to be in order to meet future challenges. Gap analysis, therefore, constitutes the foundation for our evaluation of the organization's improvement potential and the execution of actions to fill the performance gap. Figure 5.1 shows the principles for such a gap analysis.

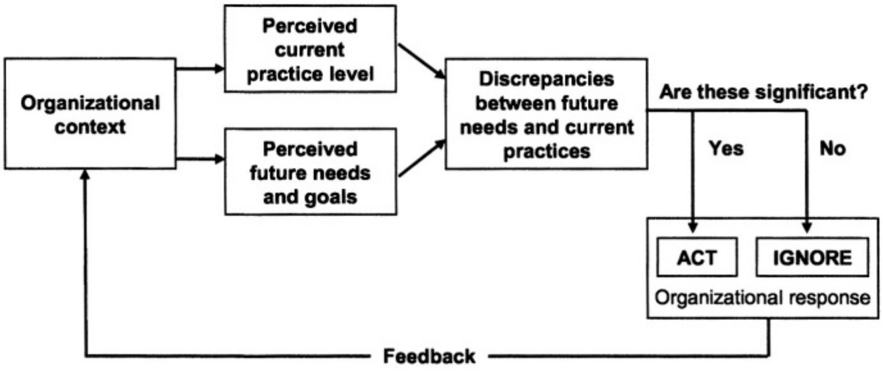

Organizational context
Perceived current practice level
Perceived future needs and goals
Discrepancies between future needs and current practices
Are these significant?
Yes
No
ACT
IGNORE
Organizational response
Feedback

Figure 5.1. Principles for gap analysis

When working with process assessments we need to acquire an understanding of the external business environment as well as the internal processes having an influence on the development of products and the experiences concerning their use.

Process assessment then involves:

- Gathering information on the business environment such as strategic position and direction, strategic goals, and competitive strategies that might influence the process improvement work.
- Examination of the organization's activities and infrastructure for software development in order to identify and prioritize problem factors that should or must be resolved.

During the planning of the improvement process it is crucial that we understand the company's relationship with the environment as well as the internal processes.

Assess the environment

In order to provide information on the close environment it is important to analyze the competitive forces that influences the development unit. The purpose of this is to identify the most important forces in the environment and the limits they enforce on the company's options concerning process improvement. Typical questions are:

- Who are the company's customers, suppliers and competitors, and what are their strong and weak points?
- Which criteria do the customers emphasize regarding the company's products and services?
- What are the characteristics of the successful companies and products in the marketplace?
- Which trends are of importance concerning the market development, which economic, social or marketing factors are most important?
- What type of technology is used in the trade line? What are the future plans? Within what areas is it most likely that new methods and tools will have a potential?
- How may processes improvement influence the company's strategic position and have an impact on the competing forces in the environment?

Assess the organization

Value-chain analysis is a method that has proved to be helpful to assess internal circumstances in the company. This analysis examines and evaluates the way values are created in the development process. The value-chain is a simple model of the company's internal organizational relationships. An example of a value-chain for software development is shown in figure 5.2.

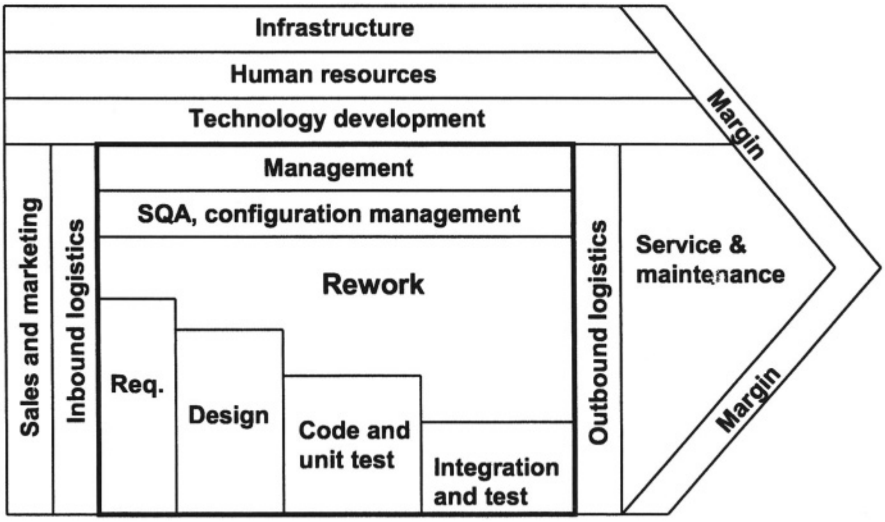


Figure 5.2 Example of value-chain for software development.

Value-chain analysis follows the organization's processes from sales/marketing of development competence to the finished product and its maintenance. The most important thing is to focus on the processes that are crucial to the company's competitive edge. Typical questions for a value-chain analysis are:

- What does the value-chain look like? What primary and secondary activities does it comprise of?
- What process areas should be benchmarked? Should they be benchmarked against other organizations or "best practice models"?
- What are the critical factors in the organization's value-chain related to costs, quality and innovation?
- How well do we exploit our resources? How quickly do we get results?
- How well do we control our costs?
- How well do we create quality advantages? How good are we at creating new products, processes or technologies? What have we learnt from experience?

In addition to showing the weak points it is important to understand the strong points that are crucial for the organization's present and future competitive edge. Is this, for example, the good relations to crucial customers? Superior organizing of the development work? Or core competence that cannot be copied by the competitors?

In the rest of this section we will take a closer look at how we may implement software process assessment related to the company's goals and needs using a participative approach to gap analysis.

5.4 A PARTICIPATIVE APPROACH TO PROCESS ASSESSMENT

Participative software process assessment emphasizes the uniqueness of each organization through active participation from the organization's managers and employees in identifying problems and challenges. In several companies it has also proved to be useful to involve key customers in this process.

The key word is *participation.* There are several reasons for that. First participation contributes to making both leaders and employees accept the results from the assessment as relevant and useful. Furthermore, participation makes it natural for employees to take on responsibility for the identified problems and get started with improvement work.

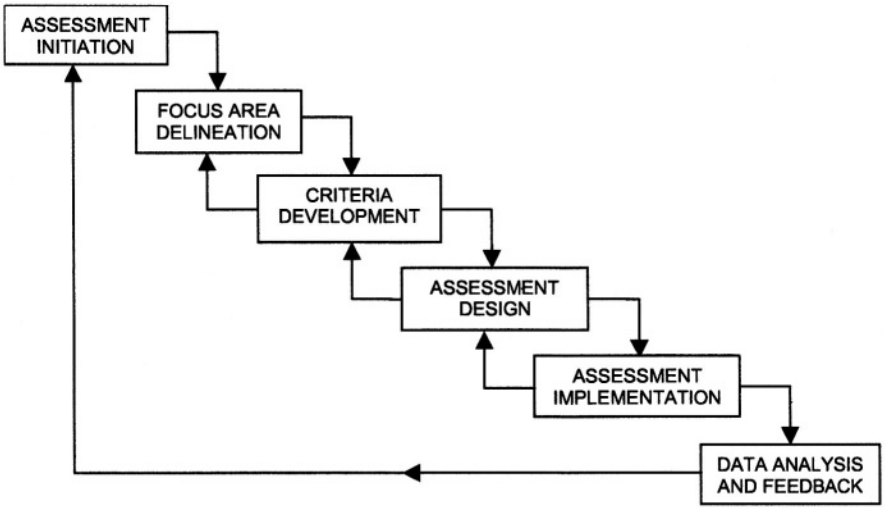


Figure 5.3. Participative assessment process.

Active participation from managers and employees is important to make process assessments provide useful results in the improvement process.

Gap analysis includes assessment of both the business environment and of the organization, concerning present strong points and future relevance/desired situation. The process is participation oriented and is based on a version of action research called "survey feedback".

The process for participative assessment comprises six steps, as shown in figure 5.3:

1. *Assessment initiation.* In the first step, the insiders and outsiders of the organisation should clarity their respective roles and the objectives of the assessment by answering the following questions:

- What are the purposes of performing software process assessment?
- Who are the users of the assessment, and how will the results be used?
- What is the scope of the assessment in terms of organisational units and issues?
- To what extent is there a commitment to using scientific methods (e.g. psychometric principles) to design and implement the assessment?
- Who should conduct the assessment, and what resources are available?

It is important that due considerations are taken in answering these questions, since they are crucial for determining whether an assessment is relevant in the first place, and for tailoring the process and content of the assessment to the specific needs of the organisation.

2. *Focus area delineation.* The second step is an exploration of the overall issues identified for the assessment in step one. In our experience, most companies do not have a shared understanding of their specific goals. A conscious analysis of commonly used high-level performance goals and focus areas in standards and reference models can, therefore, be useful as a starting point for group discussions in this step. Examples of such focus areas are software processes (e.g. customer-supplier, engineering, support, management and organisation), competitive priorities (e.g. price, quality, flexibility, and service), organisational learning (learning from past experiences, learning from others, and current SPI practices), and perceived factors of success.

   Note! Even if a lot of this is very important, it is not realistic to believe that you may do everything at once. Setting the priorities is therefore very important.

3. *Criteria development.* In the third step, multiple operational criteria are developed for each of the high-level goals that were given priority in the previous step. This requires special attention concerning the selection and definition of each criterion, as these will be measured and used as indicators of goal achievement. A decision also has to be taken regarding the use of aggregate or composite measures and any weighting of single criteria.

   To operationalize the criteria, we define one question for each characteristic we want to know something about. We also define two subjective rating scales for each question: one to rate the current strength or practice level and one to rate the future importance. Figure 5.4 shows and example of a question format from such a questionnaire.

Current strength						**Future Importance**				
1	**2**	**3**	**4**	**5**	**DELIVERY**	**1**	**2**	**3**	**4**	**5**
☐	☐	☐	☐	☐	***Ability to deliver on schedule***	☐	☐	☐	☐	☐

Figure 5.4. Typical question format from the questionnaire.

4. *Assessment design.* The issues pertinent to step four relate to where the assessment will be conducted (organisational units), the role between employees and any hired staff, the time horizon of the assessment, the unit of analysis, who are supposed to answer the questions, how the data will be collected, how aggregate concepts will be measured, and how the data will be analyzed and fed back..

   It is important to note that the more rigorous the assessment design becomes, the greater the time, costs, and other resources expended on the assessment are likely to be. Therefore, one should ask the question at every decision point whether the benefits that result from a more sophisticated design to ensure accuracy, confidence, generalizability, and so on, are worth the investment of more resources.

5. *Assessment implementation.* In step five, the assessment is implemented according to the procedure decided upon in the previous step. The main considerations during this step are completeness and honesty in data collection procedures and the recording of unanticipated events that may influence the assessment results.

6. *Data analysis and feedback.* The major concerns during step six are to provide opportunities for respondents to participate in analysing, interpreting and learning from the results of the assessment. And, furthermore, to identify concrete areas for improvement. There are many ways in which this could be done. We have relied upon half-day workshops in which preliminary findings on initial questions and problems are presented verbally, in writing, and with illustrations.

   These workshops begin with a review of the objectives of the assessment, the focus areas and the design and implementation of the assessment. Findings regarding the scores on current strengths and future importance are presented in terms of a gap analysis. Normally, the participants raise a multitude of questions and issues when the findings are presented, and they take part in group discussions and reflections as they review and evaluate the preliminary findings. Some of the questions can be clarified and answered directly with the data at hand, other questions can be answered by reanalysing the data, and finally some issues are raised which cannot be resolved with the current assessment data. In this case, a decision has to be taken regarding further data collection.

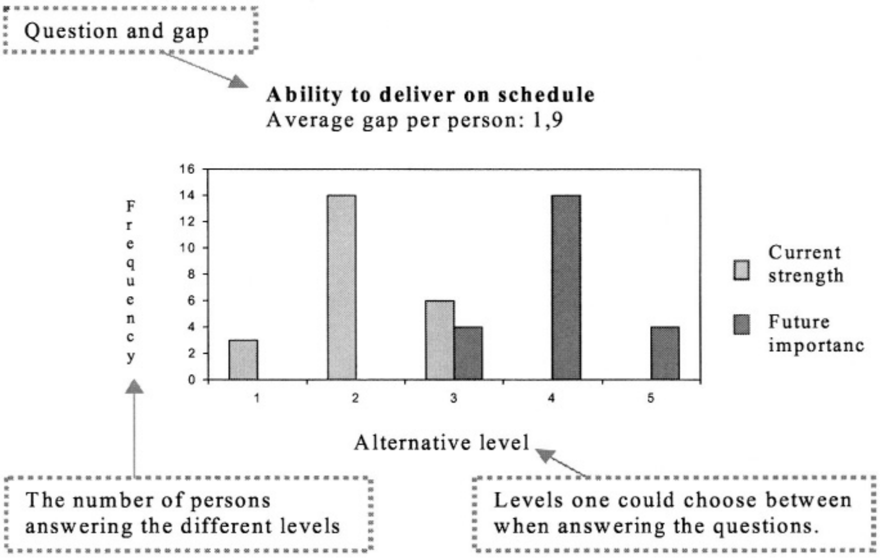


Figure 5.5. Illustration of preliminary findings from an assessment.

Typically, we use scatter plots, bar charts and histograms to illustrate preliminary findings from an assessment in order to highlight gaps between current levels of practices and future importance and the dispersion of responses both within and between groups.

Figure 5.5 shows an example of an illustration that was used in a feedback session presenting preliminary findings from an assessment. Some of the information was presented verbally. The extra information for this graph was "100% of the respondents have a gap that is larger than or equal to one".

The following tools may be useful when going through the assessment steps:

Spreadsheet including Statistics Module, Simple Database, Publishing Tools, Feedback Meeting, Setting Priorities, Action List.

Chapter 6

PROCESS GUIDES

In this chapter we take a closer look at the value for a company in having guidance to normal work processes in the company, and describe a method where many employees are involved in defining the work processes. The chapter also deals with adjustment of process guides to different types of projects and maintenance of project guides.

6.1 WHAT IS A PROCESS GUIDE?

In the introduction we described a process as a set of activities and decisions to do a certain job, for example planning a project or making a new requirement specification.

What then is a process guide? A *process guide* is a structured document describing the workflow for one or several processes.

The purpose of a process guide is to help people implement the process, to do the job effectively and with a good result.

The description of the processes must be simple to understand and easy to follow in practical work. Such guides may have textual and graphical descriptions showing the normal practices in your company. We may also look at the process guide as a short of local Who - What – How!

An *electronic process guide* is a hypertext version of the process guide providing good possibilities for adjustment to the individual projects.

It may be wise to link the process description to what has been done in real projects, in this way it will be easy to find examples of how other projects were carried out. Several companies have their own project webs, with an overview of finished and running projects, and with information on the different processes used in each project.

A lot of people probably have poor experiences concerning internal manuals and procedures - they are stuck in a binder in the shelf and are never used. Some people think of such material in the same way as documentation of code – a necessary evil you need to do because some bureaucratic person needs it.

The reason why they do not get used is that the descriptions are too complicated or too simple, they are not updated, they are written by someone with little or no insight on how things are carried out in practice, or just the fact that people find it too tedious to look things up in a binder.

But if they are developed in a proper way, process guides may be very helpful. The same job need not be done several times, the number of faults may be reduced and it will become easier to keep the time estimates in the projects. To ensure that the process guide becomes a tool that will be used, it is important to involve those who are going to use it when making it.

Process guide at Firm

The company called Firm developing software for market analysis - has developed a process guide for their development process. The process comprises of analysis, design, code, module test and integration, with some decision points or milestones on the way. The model distinguishes between administrative support activities (or "line activities") i light boxes and "project activities" in dark boxes.

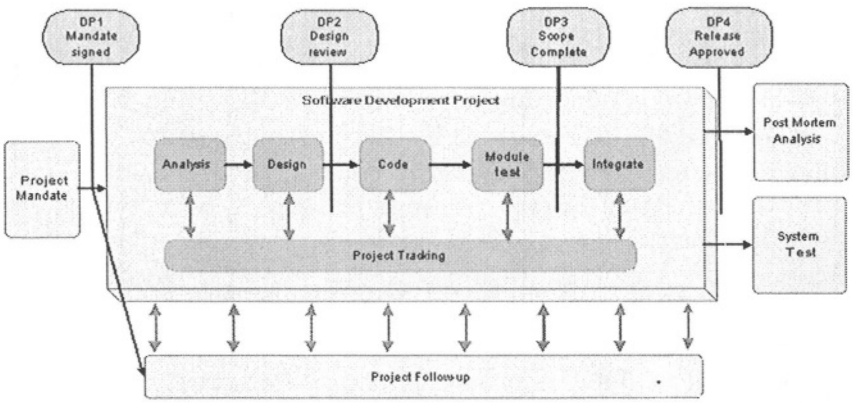


Figure 6.1 Process guide at Firm.

6.2 WHY IS A PROCESS GUIDE USEFUL?

What is the purpose of having a project guide in a developing company? Sometimes you get stuck in your work, or you do not remember how you performed a task the last time you did it. Then a guide may be practical, describing how a task normally is performed. This will be particularly useful for newly employed people who more easily will find out what the framework for the projects is and what is actually done in the company. It will also be useful when estimating the scope of new projects. With a description of how projects normally are carried out it will become easier to evaluate the size of the project. Additionally the project guide is useful because it may cause discussions concerning working methods in the company. It is easier to give other people hints when they can be related to a single process, and it is easy to discuss how you work when the process is described.

A disadvantage for a lot of people is when the process descriptions become too normative or extensive, implying that you must work in a defined way no matter the special needs in the project, or that the tasks are described in such a detail that the job is reduced to routine and becomes tedious.

It is important to find a balance to make the process guide useful in practical work. What do you need to know in a project? The process guide can be used in many ways to:

- *understand problems* – develop understanding and context concerning a problem in a project.
- *do a job* – read instructions in a guide to get trivial tasks done quickly and correctly.
- *find facts* – find facts on how a task usually is solved.
- *anticipate project development* – anticipate what will happen further along the project based what usually takes place.

A lot of standards, such as ISO 9001-2000 require that the processes are defined in the company. Additionally a process guide may simplify the work if it contains documents that you frequently need. We will take a closer look at that in the next section.

6.3 WHAT SHOULD WE PUT INTO A PROCESS GUIDE?

From the model "Entry – Task – Verification and Exit" (ETVX) we may extract a number of elements in the description of a process.

A process guide may also hold checklists for the tasks in each activity, and for the fulfillment of preconditions or post-condition It may also hold examples of input and output documents, standards that should be followed in an activity and templates for documents to be written in the process.

Input: Data or material necessary to implement the activity.

Entry: Necessary conditions to implement the activity.

Task: What to be done and how to do it.

Verification: Measurements and feedback during the activity. See chapter 4 for a thorough description of measurements.

Exit: Requirements that must be fulfilled to finish the activity.

Output: Data or material developed or modified during the activity.

In addition a process may define different

Roles: The members in the activity and their responsibilities.

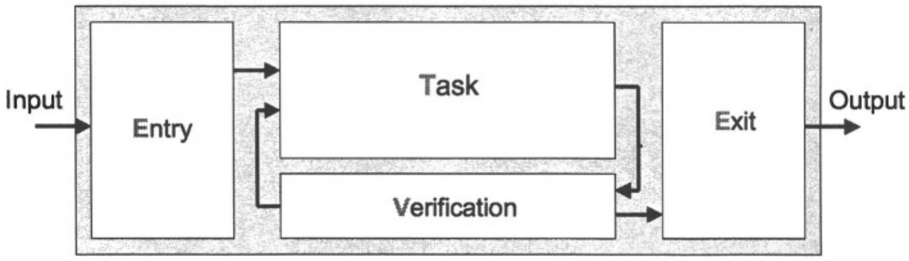
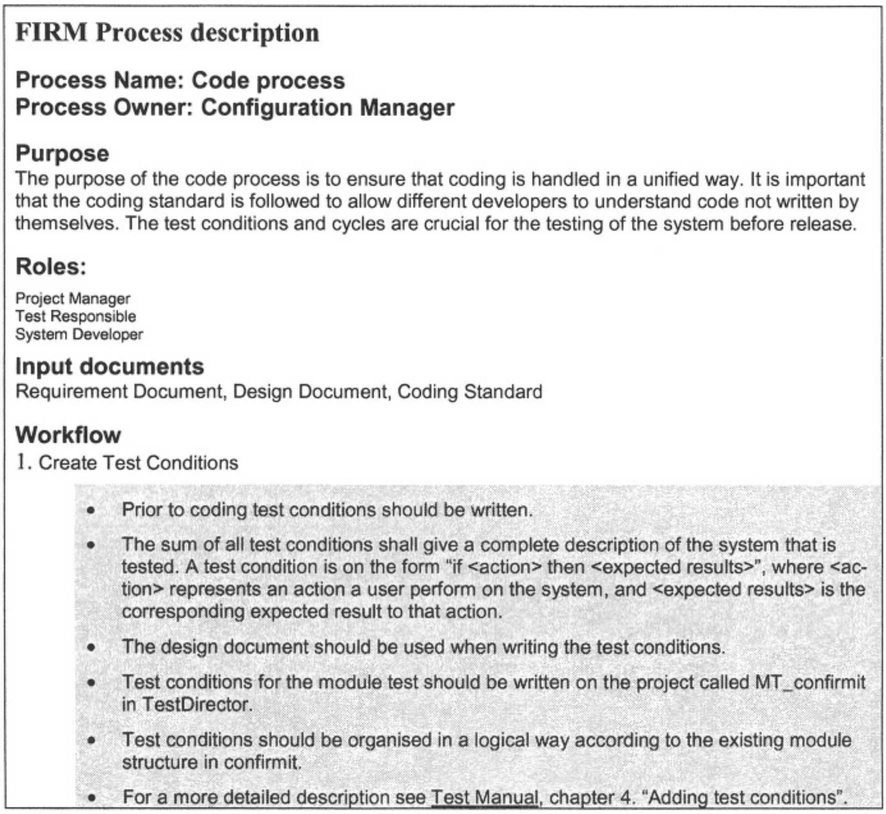


Figure 6.2 Elements in a process: Entry, Task, Verification and Exit.

6.4 HOW TO DEVELOP A PROCESS GUIDE

The main parts of a process guide can be developed during one or several workshops. We strongly recommend this method in order to involve as may as possible of the people going to use the processes in the development of the process guide. This establishes ownership towards the processes – and the discussions on what to include or not may reveal many different practices.

Such a workshop may take from 4 hours to one and a half day with 4 – 8 participants. In larger companies you will need more time.



FIRM Process description

Process Name: Code process
Process Owner: Configuration Manager

Purpose
The purpose of the code process is to ensure that coding is handled in a unified way. It is important that the coding standard is followed to allow different developers to understand code not written by themselves. The test conditions and cycles are crucial for the testing of the system before release.

Roles:
Project Manager
Test Responsible
System Developer

Input documents
Requirement Document, Design Document, Coding Standard

Workflow
1. Create Test Conditions

- Prior to coding test conditions should be written.
- The sum of all test conditions shall give a complete description of the system that is tested. A test condition is on the form "if <action> then <expected results>", where <action> represents an action a user perform on the system, and <expected results> is the corresponding expected result to that action.
- The design document should be used when writing the test conditions.
- Test conditions for the module test should be written on the project called MT_confirmit in TestDirector.
- Test conditions should be organised in a logical way according to the existing module structure in confirmit.
- For a more detailed description see Test Manual, chapter 4. "Adding test conditions".

Figure 6.3 Cutout of a display with description of the coding process at Firm

We have the following assumption as a base when we recommend using this method:

- Those who perform the daily work know a good deal about what they are doing, it is therefore important to involve them in the modeling of the processes and not leave it to the "experts".

- We visualize the models in order to spend as little time as possible on syntax.
- We regard both the workshop and the result coming from it as important ways of learning in the company.

A process workshop has the following steps:

1. *Decide on the process*: Define the process where you need a process guide. Examples may be the "development process" or the "test process".

2. *Invite members.* Invite as many members as possible from the people who will be involved in using the process guide, people with various background and working tasks is an advantage. If there are so many participants that the work should be done in groups, it may be wise to first use groups of people at the same level in the company, and then mix them the next time.

3. *Define activities.* Define the main activities in the process. Use techniques such as KJ or brainstorming to find them. It may be wise to start from the "back" when defining techniques – for example from "tested product" and backwards to the "test process" and the "test plan". Descriptions of activities should include what information is needed to implement the activity.

4. *Define the sequence of activities.* You may for example make stickers for each activity an put them on the wall. Then you find the flow between them.

5. *Define input and output.* Find out what documents should be available (and any conditions to be met) to start each activity, and what documents (and any possible conditions to be met) to close each activity. Use another shape on the stickers (for example circular) for input and output and put them on the wall together with the activities.

Conditions to be met in order to start or close an activity may be presented as checklists. Below is an example of a checklist of what should be included in a project plan:

- The project goals
- Project milestones
- Roles held by the members of the project
- Project hazards

- Procedure for handling of changes
- Plan for testing and quality assurance
- Configuration management

6. *Define roles:* Find the different roles (developer, project manager, tester and so on) that should contribute in each process - and define the responsibilities.

7. *Find related documents.* Find which documents already exist that may be linked to the process guide. Such documents may be templates, checklists and good examples of input and output documents.

8. *Delegate the responsibility for implementation.* Give somebody the responsibility to develop a draft for the process guide based on the general description of the processes. Each activity probably needs to be described more in detail that what has been established on the workshop. Divide the work between you!

The following tools may be useful when going through the steps in the process workshop (see chapter 7).

Time scale, brainstorming, KJ/affinity diagram, prioritizing.

Make sure that the text in the process guide is easy to read and informative. Structure the document in such a way that it is easy to find your way through. Do not organize it according to the author's stream of consciousness, or on where on the time scale the activities take place, focus must be on delivering the message. Here are some hints that may be helpful:

- Emphasize the importance of parts of the text by using bold or italics, but use these means sparely and consistently.
- Bulleted lists make the text easier to read, but include only elements of equal importance in the list.
- Use examples, analogies, scenarios and illustrations to give a better and wider impression of the points.
- Later it might be useful to test the process guide in a pilot project - in order to collect feedback.

When all projects may start to use the project guide it may be "published" in a company meeting where the main issues and the experiences from the pilot project are presented.

If the company has a lot of employees and a separate group for process improvement, it will be natural that this group does a lot of the practical work on the process guide. But it might be very useful for example to interview people working in the projects, or observe how people work, in order to get a description that is not very unpractical compared with how people actually are working. Learning meetings may also provide valuable input to process guides.

Visio is a suitable tool for making illustrations of processes – supporting standards for process models such as IDEF0. Otherwise, PowerPoint is a useful tool – but updating will be a bit more difficult.

Tools for Process Guides

The simplest way is to publish the process guide on the company's intranet. This might be difficult as the company may need several specially adapted processes – for example related to the size and complexity of the project. In such cases it might be useful to use tools that can generate process models, such as Spearmint.

How much in detail? How much in detail should each activity be described? It is difficult to say anything in general about that. Activities should be describes in so much detail that it actually helps you to find out what to do. On the other hand, the process guide will easily become difficult to find your way through if there are a lot of lengthy descriptions.

It might be useful to have descriptions on several levels – in this way the users may click their way to the detail level they need.

Examples are very often a good help.

In extensive process guides it might be useful with navigation tools– either by providing a widow with the headlines for all the processes, or by having a clickable process model available in all windows.

6.5 HOW TO ADAPT A PROCESS GUIDE

Sometimes projects will not fit into the "standard" process guide. Then the process guide must be adapted to make sure that we do not generate extra work when sticking to the guide.

Larger project probably need more support than smaller projects, and complex projects have different needs that simple ones. The company may have projects that involve both software and hardware, and they are perhaps run in a different way than the projects with software development only. The company may also have some project for web applications – with far more focus on the user interface and then needing more user interaction than other projects? Differences in time schedules, expected quality, the experience of the participants, costs and the level of technical difficulties may also imply that the default process is not always is suitable.

Regardless of the type of project, it will be wise to go through the outline of the process guide when starting the project to find out what you want to use. Some companies develop different versions of the process guide; adapted to the type of project they are running, for example small, medium-sized and large projects.

Other companies have a set of process elements, and each project combines its own process guide from the elements.

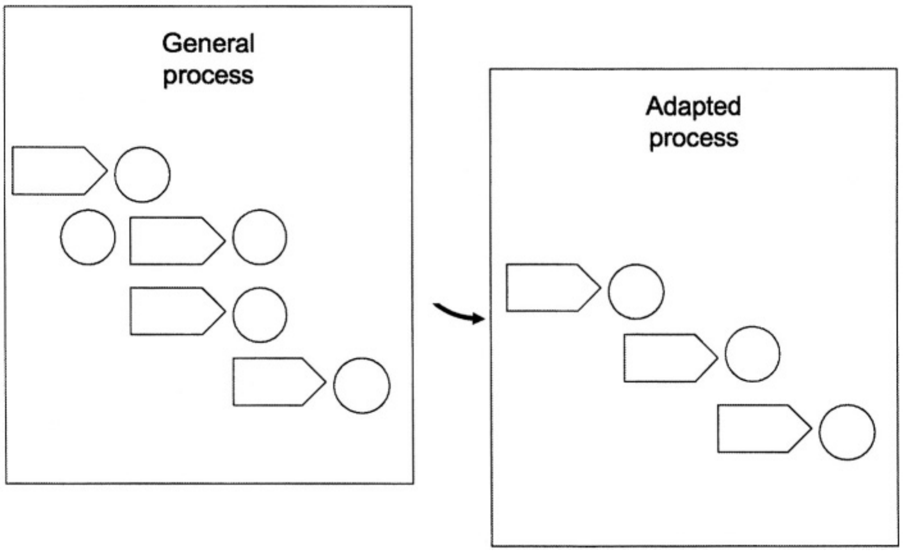


Figure 6.4 Adjustment of a general process by simplifying it.

The arrows indicate activities and the circles input or output.

Adapting the process model to a project may imply following parts of the process, merging activities in the process, put special emphasis on some activities or define separate contents for some activities.

The process guide may be adapted in the following steps:

1. *Evaluate the scope and complexity.* Evaluate the size of the project compared to the project supported by the process guide. Make considerations concerning special needs in the project caused by complexity or other circumstances.

2. *Make an instance of the process guide.* Make your own process guide for the project based on the template which is most suitable (if more than one is available).

3. *Adjust the process guide.* Go through the project outline and evaluated what to include and not – and what should be expanded.

It might be a good idea to develop a local guide for how to adapt processes in the company.

The following techniques may be useful when adjusting the process guide (se part II).

Set priority.

Acitivity	**Process**		
	small	*medium*	*large*
analysis	must	must	must
design	should	must	must
code	can	should	must
module test	drop	should	must
integration	drop	can	should

Figure 6.5 Example of overview on what "must", "should" and "may" be included in projects of different size.

6.6 HOW TO MAINTAIN A PROCESS GUIDE

A project guide should be revised regularly in order to incorporate changes coming from changed practice or from suggested improvements.

We may distinguish between two types of maintenance of the process guide during a project.

Updating as you go along. During the project work people may find faults in the process descriptions, or smarter ways of doing things. Either people must be allowed to revise the process description directly, or they must have the possibility to send a suggestion to an editor who has the general responsibility for the entire or parts of the process guide.

Revising – when the project is finished the project group should assess which processes worked well and which did not, and then revise the project guide accordingly.

Such maintenance can be a part of the learning meeting described in chapter 3.

This page intentionally left blank

Chapter 7

TECHNIQUES

TECHNIQUE 1: FEEDBACK MEETING

What is a feedback meeting?

A feedback meeting is a meeting where the results from a measurement program are discussed.

To make a measurement program function it is of importance that the feedback meetings are of high standards. The purpose of such a meeting is to combine the measured data with the knowledge of the people collecting them. Feedback meetings may also successfully be combined with other methods where we collect data, for example root cause analysis.

Why should we arrange a feedback meeting?

A feedback meeting is an important tool to validate, analyze and interpret measurement data and is a way to identify improvement actions in the project. The purpose of the feedback meeting is to present the data for those who collected them in order to get their interpretation. By combining the participants' knowledge and understanding of the development process with the collected measurement data, we will often be able to make better decisions based on fewer data than if we had only based the decision on a pure statistical analysis. A feedback meeting provides a lot of participation, which again contributes to motivation and to keep the measurement program alive.

How do we implement a feedback meeting?

Before the meeting the collected data must be analyzed and transferred to diagrams that will be presented in the meeting. The presented data is then interpreted by the participants. The data will provide answer to the questions in the GQM plan, and based on this it should be possible to conclude whether the goals in the measurement plan are achieved or not.

Select a chairperson for the meeting and follow the steps below:

Step 1:
Make a quick review of the GQM measurement plan.

Step 2:
Present the diagrams with the measurement data for the project members.

Step 3:
Discussion. Discuss and interpret the measurement data and look for non typical values, trends etc.

- Non-typical values caused by special situations are removed from the data set and put into a separate data set for analysis. An example of a non-typical value is little production in a period because several members of the project group were busy correcting faults for another project.
- Look for trends in the data, but remember that the plots only give indications. Make sure that apparent trends are followed by statistical analysis before the final conclusions are drawn.
- Are the values as expected? Unexpected values may indicate that the procedures for data collection were not good enough, as for example that not everybody have collected data in the same way. It may also indicate that the assumptions of what is going on are wrong and should be corrected.

Step 4:
Update the measurement plan. Identify suggestions for changes to the measurement plan, for example new metrics, changed data collection procedures etc.

Step 5:
Improvement actions. Identify changes to the ongoing project and ideas for long term improvements. The changes may be documented in an action list.

Step 6:
Document. Meeting minutes should be written for each feedback meeting.

The minutes should not be a very extensive document, but the crucial results from the meeting should be written down.

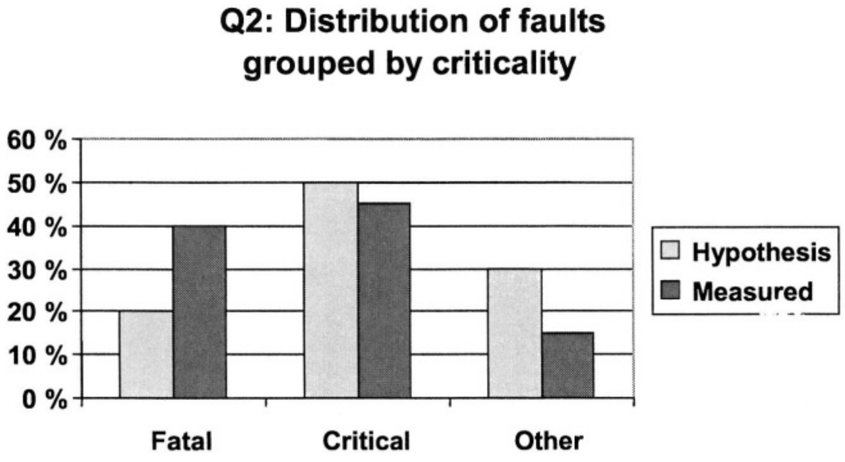


Figure 7.1. Presentation of measurement data.

Hints for feedback meetings

- Minor process adjustments during the improvement project giving quick results are important to keep up the participant's engagement and to satisfy impatient – and often reluctant – managers.
- The presentation should not hold more than 15-20 slides. If there are more slides the meeting will last too long. A meeting should last 1.5-2 hours, never more than 3 hours.
- It could be a good idea to start with a meeting every month, and then after some time have one every second month.

- We will often need the first meetings to validate and adjust the data collection, but too much time must not be spent before we draw the first conclusions.
- Statistical analysis methods may be used where there is a lot of data. In these cases it will, however, be wise to present simple diagrams first and get feedback from the project members.

TECHNIQUE 2: GQM ABSTRACTION SHEET

What is a GQM Abstraction sheet?

The abstraction sheet in GQM is a tool to document and structure the information. It consists of a sheet divided into four squares where we document the relevant aspects of the investigating objects – one sheet per goal.

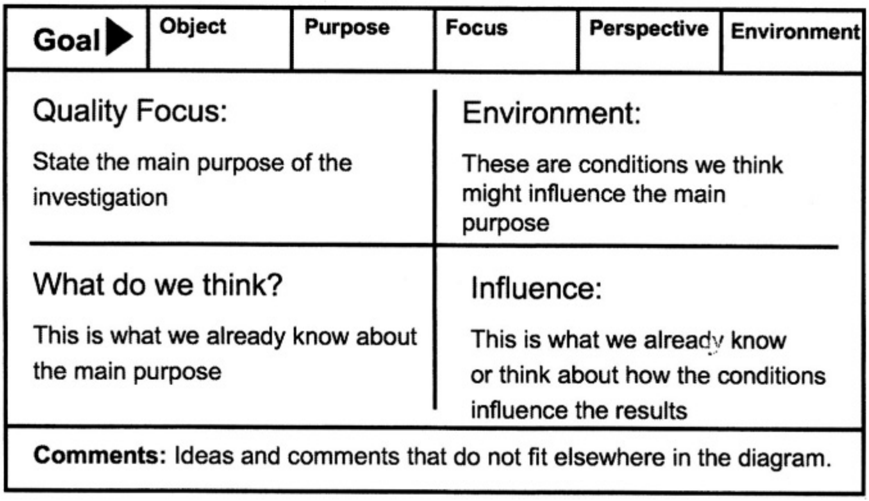
The two upper squares are filled with questions concerning focus and environment. The two lower squares hold the information that is available at the moment concerning values and influence.

In addition it is important that we document other ideas and suggestions that do not fit directly into the four squares on the sheet.

Why should we use the abstraction sheet?

The abstraction sheet is a central part of GQM. It has proved to function well during interviews, both with individuals and with groups.

When interviewing individuals we may organize the information in one sheet per person, and then merge all the information into one abstraction sheet per goal.



Goal ▶	Object	Purpose	Focus	Perspective	Environment

Quality Focus: State the main purpose of the investigation	**Environment:** These are conditions we think might influence the main purpose
What do we think? This is what we already know about the main purpose	**Influence:** This is what we already know or think about how the conditions influence the results

Comments: Ideas and comments that do not fit elsewhere in the diagram.

Figure 7.2. GQM abstraction sheet.

Alternatively this process could take place in plenum, which will save time and costs by documenting the results continuously in the sheet, for example on a board. We should therefore prefer this method if possible.

Interviewing the participants one by one and then merge the single contributions into a collective sheet are a much more time consuming and expensive process.

How do we use the sheet?

The working sheet is mostly used as a part of a process implemented by a group or a team. The purpose is to conclude with a defined improvement goal with the corresponding questions and metrics.

Step 1:

Define goals. The GQM process always starts with a defined improvement goal.

The first thing to do is to reformulate and specify this goal according to the GQM syntax.

"Analyze the <object> for the purpose of <purpose> with respect to <quality focus> from the viewpoint of <viewpoint> in the following environment: <environment>"

As soon as the goals are defined, we may start to fill in the four squares in the GQM abstraction sheet.

Step 2:

Define the quality focus question. In the upper left square we put all questions throwing light on the focus or linking focus to the purpose of the improvement goal.

The form of the questions depends on what we want to achieve. Generally it may be wise to ask specific questions; otherwise defining the metrics may take a long time. Yes/no questions should be avoided. We usually get the best questions when starting with what we believe about the issue to be measured.

The questions may very well be asked in such a way that you will need two or more metrics to find the answer.

Step 3:

Define the environment question. The environment is very often related to properties of the project, product, the company or the market. It is important to be critical when selecting questions related to the environment factors. The following points should especially be assessed:

- Factors that usually vary from one project to another.
- External factors representing a hazard to the project.

Step 4:
What do we believe? This square should hold what we at present think is the answer to the questions asked in the focus square. Filling in this square often raises useful discussions and gives new knowledge, often caused by:

- an inaccurate question – implying that the question needs to be expressed in other words.
- the project team has little or no experience in this area – use an estimate with a upper and lower limit, possibly also a middle value or an expected value.

Step 5:
How will the environment make an influence? The information we collect here is mainly a set of hypothesis on cause and impact. Additionally this square contains important information for the later data analysis.

We will seldom have enough resources to check on all possible relationships.

We assume that the relationships provided by the project members are the most important and we therefore want to check them first.

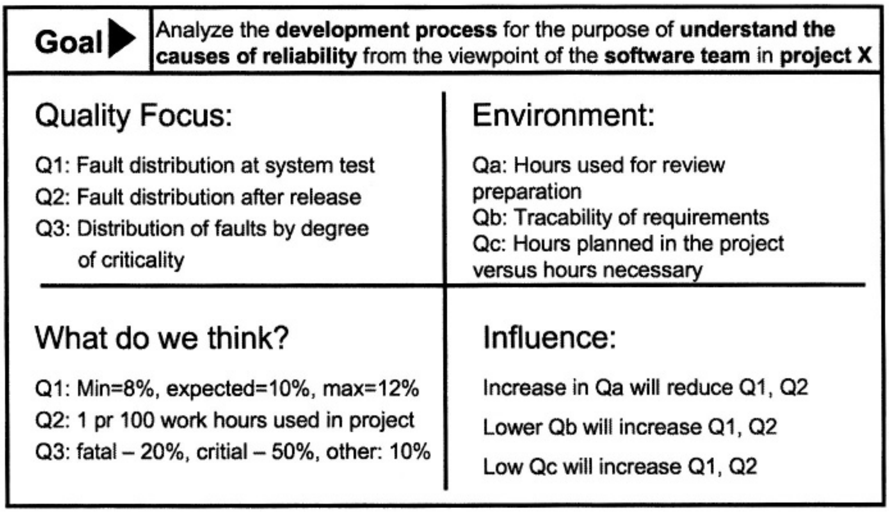
Step 6:
Include comments. This field is used to document ideas, input and other relevant items that are related to focus, environment or possible connections between these, but not fitting into any of the other four main squares in the abstraction sheet.

The ideas documented here may prove to be valuable during later feedback meetings. This is valid for example for suggestions for metrics that should be collected, or relationships that should be assessed or analyzed at a later stage.

A completed abstraction sheet is shown in the example in figure 7.3.

When the abstraction sheet has been filled in, we state the metrics related to each of the questions. Very often it will be possible to state one or several metrics providing answer to the questions. Where this is impossible, we need to implement new interviews in order to redesign the question. The new information that is provided may have the effect that the original abstraction sheet needs to be changed.

When we have agreed on the selection of metrics and how to define them, we may establish the projects' measurement plan, defining how the data collection should take place, by whom, when and how (se chapter 4).



Goal ▶ Analyze the **development process** for the purpose of **understand the causes of reliability** from the viewpoint of the **software team** in **project X**

Quality Focus:	**Environment:**
Q1: Fault distribution at system test Q2: Fault distribution after release Q3: Distribution of faults by degree of criticality	Qa: Hours used for review preparation Qb: Tracability of requirements Qc: Hours planned in the project versus hours necessary
What do we think?	**Influence:**
Q1: Min=8%, expected=10%, max=12% Q2: 1 pr 100 work hours used in project Q3: fatal – 20%, critial – 50%, other: 10%	Increase in Qa will reduce Q1, Q2 Lower Qb will increase Q1, Q2 Low Qc will increase Q1, Q2

Figure 7.3. Example of GQM abstraction sheet.

TECHNIQUE 3: MIND MAP

What is a mind map?

A mind map is a note written in a form that provides a good overview and that concentrates a lot of material in little space. The method exploits the knowledge of how our brain is organized and makes use of both halves of the brain. The right half of the brain focuses among other things on fantasy, daydreams and colors, while the left part of the brain focuses on words, logics, numbers, sequences, analysis and lists.

Why use mind maps?

Mind maps are useful to provide an overview; additionally it is a quick way to make notes for later reading, as all unnecessary information has been removed. Furthermore is easier to remember what we write down in a mind map because we combine it with the use of text, colors and pictures that activates both halves of the brain.

Mind maps are well suited for making notes from meetings, organizing material for presentations, as well as a technique for creative problem solving.

How do we make a mind map?

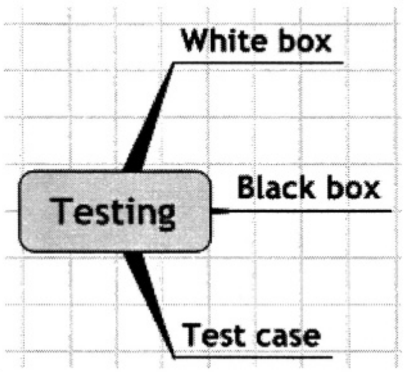
The mind map is made in the following way.

Step 1:

Main subject in the middle. Write down the main subject for the mind map in the middle of a sheet – and use few words for the description. It may for instance be testing in the project. You should willingly use colors and shading!

Step 2:

Make branches with sub-subjects. Write the sub-subject with upper and lower case letters (do not use handwriting!) on one branch from the main subject. Describe the sub-subject with keywords, not complete sentences. Use pictures and colors where possible – that makes it easier to remember!



Hints when you make a mind map

- Use large sheets of paper for large subjects – A3 or "wall sheets".
- You may very well use software tools to make mind maps. The mind maps in this book are made using *MindManager*.
- Use arrows, symbols and numbers to underline or organize sub-subjects.
- Mind maps may be used for problem solving by brainstorming over the subject. You may for example make a note of everything that was good and bad about the testing in a closed project. Then you draw a mind map of the possible solutions.
- Vary the size of the writing according to the importance of the subject.
- If you use a mind map to generate new ideas: Add new lines or ask questions if you run out of ideas.

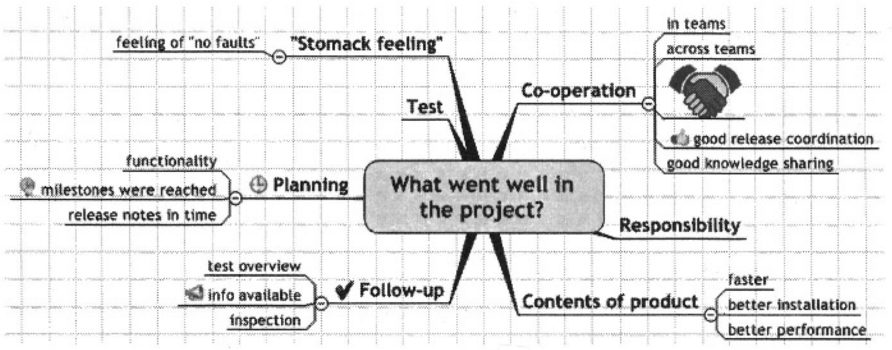


Figure 7.4. Example of mind map.

TECHNIQUE 4: BRAINSTORMING

What is brainstorming?

Brainstorming is a technique used by a group to bring forward the ideas of each individual and then present them in a structured way for the rest of the group. The key factor is to create an environment of creativity and unlimited exploration of ideas and solutions in such a way that self censorship and "accepted truths" do not control the production of ideas.

Why use the brainstorming?

Brainstorming is very useful when you want to generate a large number of ideas concerning problems that need to be solved, ways to approach the problems, or actions to implement. Brainstorming makes it possible for a group to free themselves from old ways of thinking.

How do we use brainstorming?

To make brainstorming both creative and productive, the group must agree on a set of rules.

Advantages of brainstorming

- New input on subjects. Brainstorming expands the way of thinking to include more aspects of a problem or solution.
- A large number of ideas are quickly produced
- Makes everybody get involved.
- Ownership. Participation results in a feeling of ownership, which again increases the chances that any decisions are followed up.

Select a chairperson for the meeting and follow the steps below:

Step 1:
Go through the rules. Present the process and make sure that everybody accepts the rules for the session.

Step 2:
Set a time limit. Select a person to watch the time, and one to document the results. Brainstorming should be a quick generation of ideas. 5-15 minutes is enough. If ideas still are generated when the time limit has been passed, it is possible to keep on for a few minutes more.

Step 3:
Make a question. Express the subject you want to address as a question. Write it down and make it visible to everybody all the time. Check that everybody has understood the question.

Step 4:
Collect ideas from everybody. Establish an unstructured or structured way of collecting the ideas.

- *Structured.* Everybody presents their ideas in turn. If a participant is not ready to present his/her ideas, he or she may wail until the next round. The maximum number of ideas presented by each participant per round must be decided before you start.
- *Unstructured.* The participants present their ideas as they turn up. This method requires close attention from the chairperson of the session. The rules must be followed and everyone must participate.

Rules during brainstorming

- Active participation from everybody. Everyone presents their ideas, even if they seem stupid of irrelevant.
- No discussion. No criticism or comments during the brainstorming.
- Use ideas generated by others as foundation for your own ideas.
- Set a time limit.
- Clarify the ideas. Go through the ideas to make sure that everybody understands them. Remember that you should clarify the ideas, not judge them.
- Combine the ideas. Check if two or more ideas that seem to be the same, may be combined.

Step 5:
Document the ideas. Document the ideas as they are presented.
This may be done by using stickers. The ideas must be visible all the time in order to avoid duplication of ideas, and to make it possible for others to use the generated ideas as a foundation for new ones. The result may be documented by using relevant software and a digital camera. Mind maps are suitable for such documentation.

Step 6:
Clarify the ideas. Go through all the ideas to be sure that everyone shares the same understanding. Point at the ideas in turn and ask if there are any questions. If there are questions, the person who contributed with the idea will explain.

Step 7:
Eliminate duplicates. If one or more ideas apparently are the same, try to combine them, or eliminate duplicates. Those who generated the ideas that are combined or removed must agree. If there is no agreement, the ideas are kept as separate ideas.

Hints for brainstorming

- In order to save time the participants may present all ideas during one round. This will, however, reduce the activity in the group because the participants are kept passive too long. Another disadvantage is that the person presenting his/her ideas last will often repeat what has already been said.
- The technique often produces apparently incomplete ideas, but these will often lead to new and genuine solutions to problems.
- The ideas should be visible for everybody all the time.

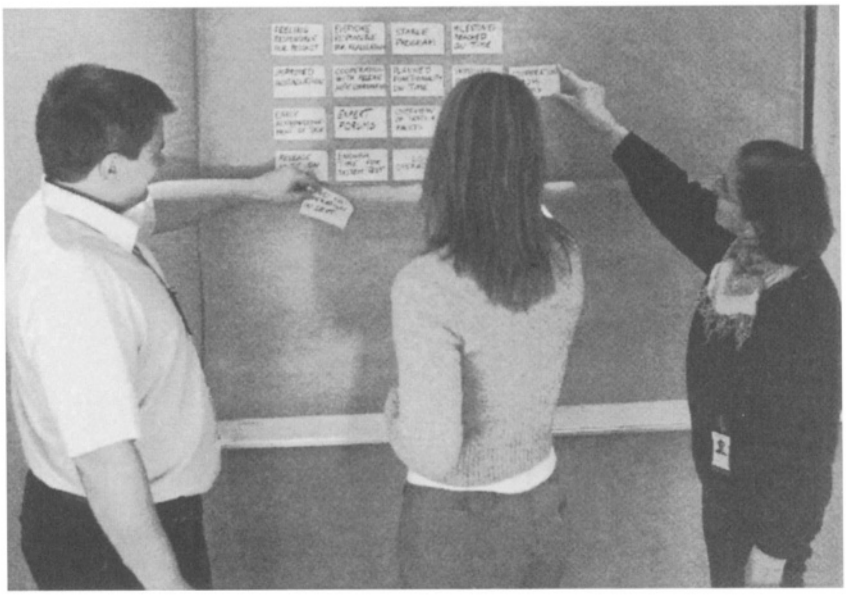


Figure 7.5. Going through ideas.

This page intentionally left blank

TECHNIQUE 5: KJ/AFFINITY DIAGRAM

What is KJ?

KJ, or affinity diagram, is a creative group technique used to organize and create relationships between apparently unrelated ideas. KJ is often used to group ideas that were presented during brainstorming. The name "KJ" comes from the initials of the Japanese anthropologist Jiro Kawakita.

Why use KJ?

KJ is used when a group wants to find an answer to one or more questions concerning the entire group, and where full agreement on the issue is wanted. An example of such a question could be: "What in the project was a success?"

How do we use KJ?

KJ is a process that is carried out by a group or a team, and it may take place in a group room with a whiteboard or wall covered with gray wrapping paper. Select a chairperson for the meeting and follow the steps below:

Step 1:
Present the subject for the idea generation.

Step 2:
Register ideas. Use brainstorming to generate a list of ideas and problems related to the selected subject. Write the ideas down on stickers – one sticker per idea. Write IN BIG LETTERS

Step 3:
Present ideas. Each participant presents their ideas and places their stickers in any place on a whiteboard or on the wall covered with paper (Figure 7.6). You are not allowed to criticize or argue against the presented ideas.

Step 4:
Group the ideas. When all the ideas are presented, the participants move and group the stickers according to how related they seem to be (Figure 7.7). Everybody participates in the process, but without discussion.

- Start by looking for ideas that seem to be related. Place them together – for example in a column on one of the sides.

- Look for ideas related to the ones that are already grouped and place them in the group.
- Look for other related ideas and establish new groups.
- The grouping should be based more on intuition than logical classification.
- Any possible "lonely wolves" are put in a separate place.

Step 5:
Make headlines. Make sure that the headline describes the ideas in the group. Take care to make the headlines understandable for people outside the team.

- Start by looking for stickers inside the group that may serve as a headline.
- Alternatively you may discuss and agree upon a covering headline.
- Look for relationships between two or more groups and arrange them in super-groups. The naming rules mentioned above also apply to the super-groups.

Step 6:
Draw the diagram. Review and clarify ideas, grouping and headlines. Draw any connection arrows between the groups and document the finished diagram.

Document the result by using relevant software and a digital camera. Mind map is well suited for such documentation (se example in the margin).

Step 7:
Sometimes it may be practical to set priorities on the ideas and/or groups. This may be done quickly and effectively by setting priorities.

Hints for KJ

- The time limit for a KJ depends on the number of participants and the number of stickers and ideas that are generated. One subject matter should not last for more that one hour. If you use more time there is the chance that creativity dies and the participants loose interest.
- To produce 30 stickers usually takes one hour. Three to fours stickers per person normally works well. If there are more than 10 participants it might be wise to place persons in groups and then let each group generate three or four stickers.

- If a sticker belongs to more that one group, you make a copy of it.
- KJ is often used along with other techniques. Then it is wise to leave the results from the KJ session visible as long as they are needed. You may well cover a wall with wrapping paper if there is no additional whiteboard.
- If several questions are going to be discussed it might be wise to change the sequence for the participants presenting their stickers.
- It is always less motivating to present your ideas last because there is a great chance that someone else already has used them.

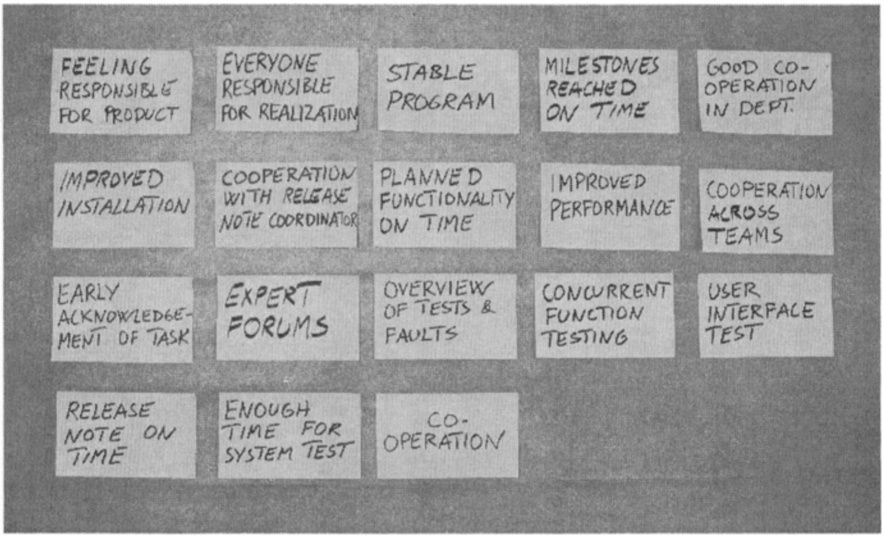

FEELING RESPONSIBLE FOR PRODUCT
EVERYONE RESPONSIBLE FOR REALIZATION
STABLE PROGRAM
MILESTONES REACHED ON TIME
GOOD CO-OPERATION IN DEPT.
IMPROVED INSTALLATION
COOPERATION WITH RELEASE NOTE COORDINATOR
PLANNED FUNCTIONALITY ON TIME
IMPROVED PERFORMANCE
COOPERATION ACROSS TEAMS
EARLY ACKNOWLEDGE-MENT OF TASK
EXPERT FORUMS
OVERVIEW OF TESTS & FAULTS
CONCURRENT FUNCTION TESTING
USER INTERFACE TEST
RELEASE NOTE ON TIME
ENOUGH TIME FOR SYSTEM TEST
CO-OPERATION

Figure 7.6. The ideas presented and placed on a wall.

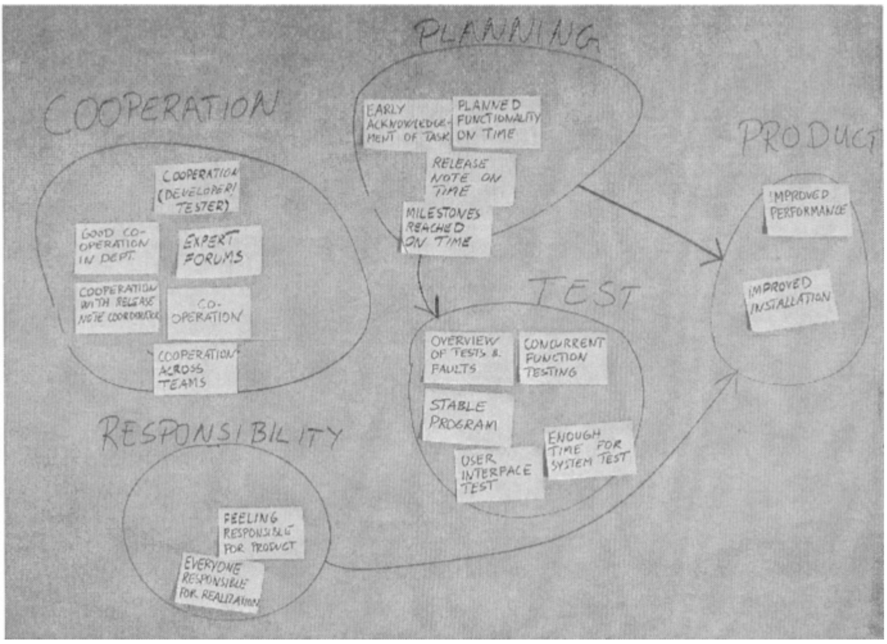

PLANNING
COOPERATION
PRODUCT
EARLY ACKNOWLEDGE-MENT OF TASK
PLANNED FUNCTIONALITY ON TIME
RELEASE NOTE ON TIME
MILESTONES REACHED ON TIME
COOPERATION (DEVELOPER/TESTER)
GOOD CO-OPERATION IN DEPT.
EXPERT FORUMS
COOPERATION WITH RELEASE NOTE COORDINATOR
CO-OPERATION
COOPERATION ACROSS TEAMS
IMPROVED PERFORMANCE
IMPROVED INSTALLATION
TEST
OVERVIEW OF TESTS & FAULTS
CONCURRENT FUNCTION TESTING
STABLE PROGRAM
ENOUGH TIME FOR SYSTEM TEST
USER INTERFACE TEST
RESPONSIBILITY
FEELING RESPONSIBLE FOR PRODUCT
EVERYONE RESPONSIBLE FOR REALIZATION

Figure 7.7. The ideas are grouped and headlines are made.

TECHNIQUE 6: PRIORITIZING

What is prioritizing?

Prioritizing is a technique to rank for example ideas, causes or actions defined by a group, according to one or several criteria.

In this section we describe a way to perform prioritizing as a democratic process in a group.

Why should we prioritize?

A process where priorities are set will help the participants to find out where you should keep up the work. It is for example seldom possible to keep up work on more that a few of the good ideas that are presented in a brainstorming. What is not given top priority in such a process may still be important and may be put aside for future work. As the entire group participates in the prioritizing, the group members will feel ownership for the results of the prioritizing.

How do we use prioritizing?

When you set priorities on a list of ideas or actions, the group must first decide the criteria the priorities should be based on.

This may be criteria such as costs, time and expected effect. An action that is simple and inexpensive to implement may be prioritized before an action that is more important, but requiring more calendar time and is more expensive.

Select a chairperson for the meeting and follow the steps below:

Step 1:
Decide on the priority criteria you should apply.

Step 2:
Agree on the number of votes each member of the group should have. This will depend on the number of elements to set priorities on, and how many participants there are. Three votes per participant usually work well.

Step 3:
Give your votes. This may be done by adding vertical lines behind the elements each participant wants to prioritize. If more that one vote is given, these may be distributed on one or several elements.

Step 4:
Make the count and present the results. Let the group comment on the results.

If there is disagreement or misunderstanding, step 3 may be repeated. If several elements are given the same priority and you need a defined priority, the final ranking may be decided by a new voting between the ones with equal votes. Alternatively the order may be decided by discussions.

Step 5:
Document. The result may be documented by using relevant software and a digital camera. Mind maps are suitable for such documentation.

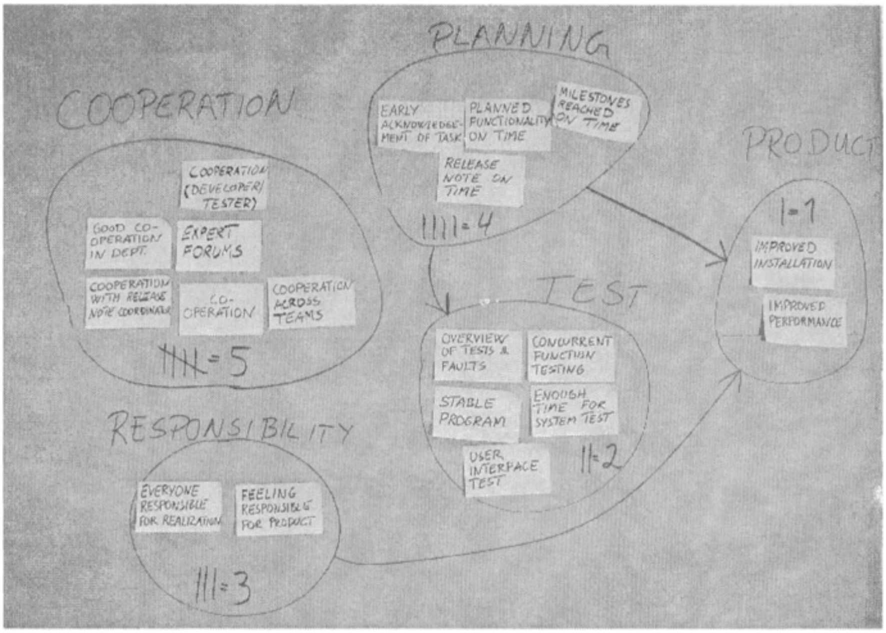

PLANNING
EARLY ACKNOWLEDGEMENT OF TASK
PLANNED FUNCTIONALITY ON TIME
MILESTONES REACHED ON TIME
RELEASE NOTE ON TIME
||||=4
COOPERATION
COOPERATION (DEVELOPER/TESTER)
GOOD CO-OPERATION IN DEPT.
EXPERT FORUMS
COOPERATION WITH RELEASE NOTE COORDINATOR
CO-OPERATION
COOPERATION ACROSS TEAMS
||||=5
PRODUCT
|=1
IMPROVED INSTALLATION
IMPROVED PERFORMANCE
TEST
OVERVIEW OF TESTS & FAULTS
CONCURRENT FUNCTION TESTING
STABLE PROGRAM
ENOUGH TIME FOR SYSTEM TEST
USER INTERFACE TEST
||=2
RESPONSIBILITY
EVERYONE RESPONSIBLE FOR REALIZATION
FEELING RESPONSIBLE FOR PRODUCT
|||=3

Figure 7.8. Priorities set by the groups describing why a project was successful – cooperation was given the highest priority.

TECHNIQUE 7: TIME-LINE

What is a time-line?

Making a time-line is a technique to place main events in a project in time – in order to se relationships and discover positive and negative experiences – and to relate these to time moments. The technique is based on the fact that a group remembers better that a single person – and may be also different events.

Why use time-line?

Time-line is used to create a common understanding in a group on what were the main events in a project and when these events occurred. The time-line can be used to make the participants look back at what happened in a project or a process, for example at the beginning of a learning meeting.

How do we use the time-line?

Time-line is a process that is carried out by a group or a team, and it may take place in a group room with grey wrapping paper placed across one of the walls. Select a chairperson for the meeting and follow the steps below:

Step 1:
Divide the wrapping paper in chronological order, with project start to the left and project end to the right Start may for example be "spring 2002" and end may be "spring 2003". Divide the time-line into suitable intervals.

Step 2:
Identify the main events. Along the time-line Distribute stickers where people may write down main events. You may very well use circular stickers for marking milestones in the project and square stickers for marking deliveries and other events.

Step 3:
Ask the participants to place their stickers on the time axis. Let one participant at a time place a sticker, and make him or her explain why this was a major event or milestone.

Step 4:
Let the participants make comments on the time axis – look for different opinions of what events were important and possibly if the participants have placed the same events differently in time.

Step 5:
Go through the project in a chronological way to find the most interesting events. Ask people what were the major events or what was must surprising. Is there any pattern in the events?

Step 6:
Use KJ on interesting events. If some events were of particular interest, you may use KJ for these. Try to establish positive and negative experiences from the major events and try to explain why some people emphasized special events or placed the events differently in time.

Step 7:
Use Root cause analysis on interesting events. Run a Root cause analysis on the most interesting events, or on why some events have been placed on different times.

Hints for using time-line

- Prepare the meeting by placing a long piece of grey wrapping paper along one of the walls, and provide enough stickers and tape.
- If there are more than seven participants, some will easily become passive. Then you should run the discussion on the events in groups.
- Take a break when the time-line has been placed on the wall to get an informal discussion started on how the project was carried out.

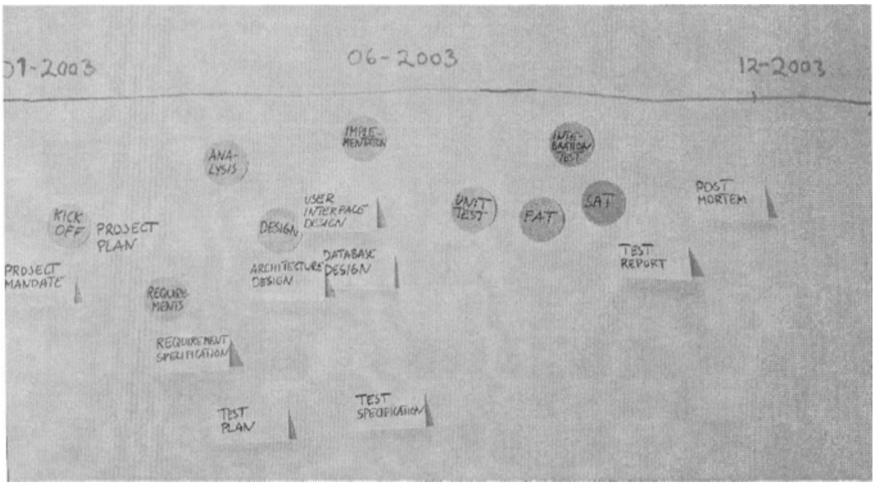


Figure 7.9. Time-line in a project with sticker for events.

This page intentionally left blank

TECHNIQUE 8: ACTION LIST

What is an action list?

An action list is an overview of measurements to be implemented for example in an improvement program. An action list is an important part of an improvement plan. It usually holds information on actions, resources, time schedules and responsibilities.

Why make an action list?

All improvement work depends on improvement actions to be carried out

These initiatives are planned and described in the action list. These initiatives are most likely to be carried out if it is easy to follow them up. It is also more likely that initiatives are carried out where there is a distinct description of them.

How do we make an action list?

An action list should be set up by a group or a team. It is important that the people participating in the actual implementation of the actions are involved.

Select a chairperson for the meeting and follow the steps below:

Step 1:
Generate an action. This may be for example achieved in a brainstorming in a feedback meeting, a KJ session or during a root cause analysis. Make sure that the actions are distinctly described.

Step 2:
Related actions are groups in "projects".

Step 3:
Set priorities to the action list. This may be done by using prioritizing. All actions may not be carried out at once, so the most important must be picked. This may become useful the next time an action list is made.

Step 4:
Describe the actions. For each action or project: short description, goal, costs, resources, time schedules and milestones.

Step 5:
Define a person who is responsible for the actions. This is the person who should follow up that the actions are implemented, and need not be the one actually doing the implementation.

Step 6:
Implementation plan. Make a plan for how the actions should be implemented in the company.

Hints for implementation of action list

- Combine short term and long term actions. The long term actions are important in order to work strategically, the short term actions are important to get quick results. Experiencing improvements actually taking place is important for the motivation.
- An action list must be realistic. Unrealistic ideas of what could and should be achieved will only be a demotivating factor for the improvement work.
- The plan must be reflected in the budgets.

Table 7.1. Example of action list.

#	Name	Description	Resources	Milestones	Responsibility
1	Implement procedure for prioritizing of tasks	Define a procedure for prioritizing of tasks	2 persons + department manager	May 1st June 1st	John
2	Improved communication with marketing	The developers must have influence on delivery times and selection of functionality	1 person for 16 hours	May 5th August 8th	Mary
3	Functioning professional groups	Increase the professional level by making the professional groups function better.	4 persons – 10 hours per person	May 1st November 1st	Jude

TECHNIQUE 9: ROOT CAUSE ANALYSIS

What is a root cause analysis?

The root cause analysis is a technique used to find the main cause and any sub-causes for an effect, either for a problem, or for something good. This technique is also called a cause/effect diagram, fishbone diagram or Ishikawa diagram named after the Japanese professor who invented the technique.

Why use root cause analysis?

The root cause analysis is used to make a common understanding in a group of what is the root cause of effects or symptoms. The analysis can be used to make a foundation for actions that may correct the problems or keep successes going.

The root cause analysis is used when a symptom or an effect is established, but it is hard to find the reasons why this happened. After a KJ or a brainstorming, the root cause analysis is used to find the cause of occurring problems or successes. The root cause analysis is most suitable for smaller groups. In larger groups the work must be split up.

How do we use root cause analysis?

Root cause analysis is a process that is carried out by a group or a team, and may take place in a group room with a whiteboard. Select a chairperson for the meeting and follow the steps below:

Step 1:
Draw an arrow on the whiteboard with the symptom or the effect you want to find the cause for. Make sure that the effect is distinctly described, but the description must not be too long.

Step 2:
Ask the participants to come up with the main causes for this symptom.
If causes are hard to find, brainstorming may be used. Make arrows for each main cause you find, as shown in the figure. Put the cause into exact words.

Step 3:
Ask the participants to find sub-causes for each main cause. Make arrows for each sub-cause that comes up. Find precise words to describe the cause.

Step 4:
Make changes in main causes and sub-causes when that is necessary.

Step 5:
Find the most important causes: If there were too many causes in the diagram, you may for example vote to decide the three most important causes. Remove the causes that got few or no votes, or put weight on the causes by using thicker arrows for more important causes.

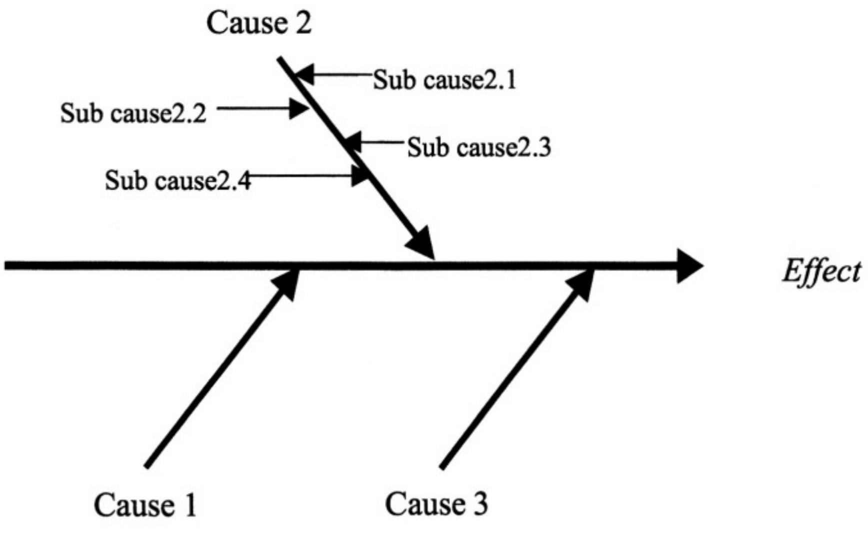


Figure 7.10. Notation in a root cause analysis with "symptom" in the middle and main causes as well as sub-causes over and under.

Hints for root cause analysis

- To find the causes, the chairperson may ask questions like: "What caused this <symptom>" and "What are the sub-causes for <main cause>"?
- If there are more than seven participants, some will easily become passive.
- You may then run a brainstorming på main causes and sub-causes in groups.

- Do not spend any time on actions when doing a root cause analysis. Actions may be found when the understanding of what caused the effect is good. Se the technique action list.
- If the work progress of finding causes stops – take a break and try over again!

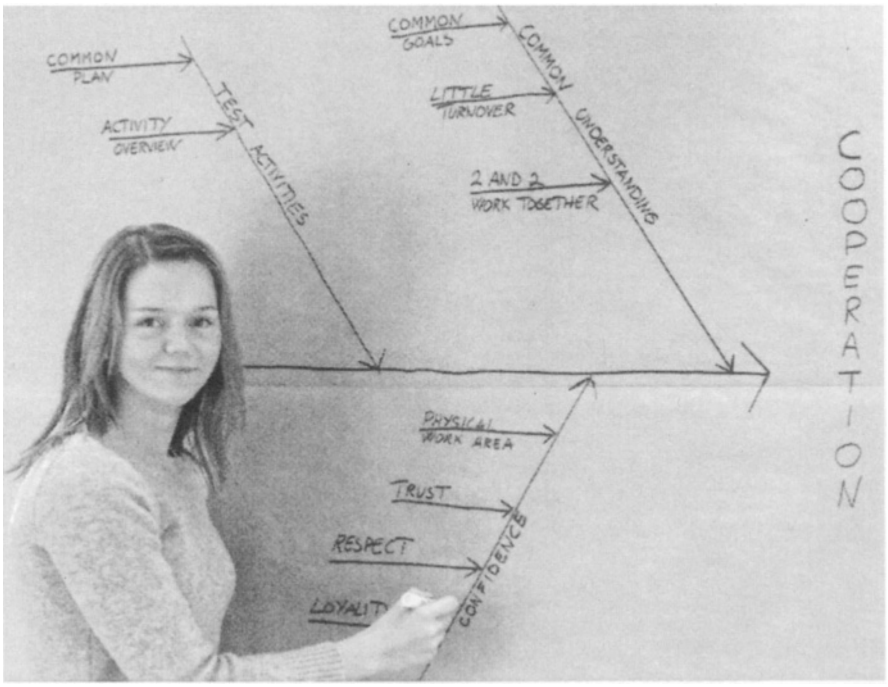


Figure 7.11. A chairperson adds a main cause to the root cause analysis.

This page intentionally left blank

Bibliography

- **Learning to Fly** – here you find a number of practical hints of knowledge management, written by people who have been employed at British Petroleum.

  Chris Collison and Geoff Parcell, Learning to Fly: Practical Lessons from one of the World's Leading Knowledge Companies: Capstone Pub, 2001, ISBN 184112124X.

- **Project Retrospectives** – the book provides an overview of different ways to organize learning meetings and post mortem analysis.

  Norman L. Kerth, Project retrospectives: a handbook for team reviews. New York: Dorset House Publishing, 2001, ISBN 0-932633-44-7.

- **Successful Software Process Improvement** – Practical book on how to succeed in process improvement... The book holds a lot of examples based on the author's work at Hewlett-Packard.

  Robert B. Grady, Successful Software Process Improvement. New Jersey: Prentice Hall, 1997, ISBN 0-13-626623-1.

- **A Quantitative Approach to Software Management** – gives a good introduction to how we may establish and implement a simple measurement program, based on Goal/Question/; Metric

Kevin Pulford, Stephen Shirlaw and Annie Kuntzmann-Combells, A quantitative approach to software management, Addison Wesley, 1996, ISBN 0-201-87746-5.

- **The Goal/Question/Metric Method** – the book provides a thorough introduction to the method of Goal/Qestion/Metric for measurement-based improvement.

  Rini van Solingen and Egon Berghout, The Goal/Question/Metric Method – A practical Guide for Quality Improvement of Software Development: McGraw-Hill, 1999, ISBN 007-709553-7.

- **Software Creativity** – the book discusses creativity in software development and gives a number of examples of techniques that encourage creativity.

  Robert L. Glass, Software Creativity: Prentice Hall, 1995, ISBN 013-147364-6.

- **A guide to the Project Management Body of Knowledge** – a book describing areas of knowledge in all phases of a running project.

  Project Management Institute, A Guide to the Project Management Body of Knowledge: Project Management Institute, USA, 2000, ISBN 1-880410-23-0.

About the Authors

Dr. Tore Dybå is chief scientist at SINTEF (the Foundations for Scientific and Industrial Research at the Norwegian Institute of Technology) and visiting scientist at the SIMULA Research Laboratory. He received his M.Sc. in Electrical Engineering and Computer Science from the Norwegian Institute of Technology (NTH) in 1986 and his Ph.D. in Computer and Information Science from the Norwegian University of Science and Technology (NTNU) in 2001.

Dr. Dybå worked as a consultant both in Norway and in Saudi Arabia before he joined SINTEF in 1994. He has been responsible for and worked in several large national and international projects concerning software and business process improvement, organizational learning and knowledge management, software quality assurance and measurement, and empirical software engineering. Dr. Dybå is the author of several publications appearing in international journals, books, and conference proceedings in the fields of software engineering and knowledge management

Dr. Torgeir Dingsøyr is a research scientist at the Department of Software Engineering at SINTEF in Trondheim, Norway. He wrote his doctoral thesis on "Knowledge Management in Medium-Sized Software Consulting Companies" at the Department of Computer and Information Science, Norwegian University of Science and Technology. He has worked on several large software process improvement and knowledge management in national and international projects as a researcher and consultant. He has published papers on knowledge management in software engineering, case-based reasoning and on software engineering education. In 2001, he spent half a year as a guest researcher at the Fraunhofer Institute for Experimental Software Engineering in Kaiserslautern, Germany.

Mr. Nils Brede Moe is a research scientist at the Department of Computer Science at SINTEF Trondheim, Norway. He joined the department in 1998. He holds a M.Sc. degree in Informatics from the Norwegian University of Science and Technology. He has worked with health informatics and software process improvement through several large projects. He has published papers on health informatics (telemedicine and homecare), user-centered design and software process improvement.

Index