17 Process Improvement

Process Improvement

Introduction

It seems almost like a truism to state that the product quality is by and large determined by what we do and how we do it - i.e. the development process. From this, it follows that if we want to improve the quality of our product and our productivity, we must improve the development process.

The development process can be improved in at least two ways:

- By improving the activities that we already perform in our process
- By changing the process, e.g. removing some activities, introduce new activities or change the sequence that we perform the activities in now.

In addition, there are three approaches that can be taken, namely:

1. Improve the process in order to reduce the development costs and try to keep the quality stable - improved productivity
2. Improve the process in order to improve the quality and try to keep the costs stable - improved product
3. Improve both productivity and quality.

Even though it is kind of surprising to many people, alternative three is quite realistic. The reason for this is that the focus of quality improvement is to do it right the first time every time, thus reducing costs and improving the result.

Whatever improvement strategy we choose, it is important to bear in mind that improvement is not a project, since a project has a start and a finish date. Improvement, on the other hand, is a continuous process: any advantage that we can gain over our competitors is temporary!

Since this book is about methodology, we first look at issues concerning Introducing new development methods in a company (p.17-5). This is a collection of experience and advice we have gained over the years that the SISU project has run.

The remaining part concerns process improvement in general.

If we shall improve something, there must exist one or more goals for the improvement. This is discussed in Introducing new development methods in a company (p.17-5), where we present The Decision Process (p.17-13), The Top Level View (p.17-14), The Influence Factors (p.17-15) and the Input to improvement (p.17-16), and also discuss Process Improvement Methods (p.17-18) and Selecting Technical Improvement Goals (p.17-28).

All change carries with it a certain amount of uncertainty. How to deal with this is discussed in Risk Assessment and Control (p.17-33), where we look at Risk in General (p.17-33) and A Company’s Attitude to Risks (p.17-33).

In order to assess which improvement steps to land on, one should perform Change Cost Analysis (p.17-42); we present two methods: Leverage (p.17-42) and Return on Investment (p.17-42).

Aspects of performing measurements of a software process can be found in the related theme of Metrics.

Lastly we look particularly at The Effect of introducing new Tools and Methods (p.17-45) into a company - an important chapter for those contemplating on introducing the methodology described in this book into an organization. We dwell on How to measure effect (p.17-45), How we can state our Hypothesis (p.17-45) of the expected effect of introducing them, and How we can Assess the Effect of new Tools and Methods (p.17-46).

We also provide a few Examples (p.17-49) to some of the methods presented here.

Introducing new development methods in a company

Ten rules

Ten simple rules to be obeyed when changing a development process:

1. Analyze your current development practices. Involve your developers in this analysis
2. Search for a change that will have measurable positive impact on major business issues of the company. Involve management in this search
3. Evaluate the risk of the change: how bad can things get, and when can you know how bad things are going?
4. Find a real development project to implement the change in - unless the risk is too great and you are able to perform an experiment instead
5. Determine how success will be measured, and when it can be measured
6. Don't overestimate your digestion - eat according to your age!
7. Don't underestimate the resistance to change - meet it face to face as soon as possible
8. Be prepared for blood, sweat and tears
9. Be patient and realistic. Changing behavior is difficult and takes time.
10. Don't forget: nothing great was ever done without enthusiasm!

Introduction

The methodology presented in this publication is extensive. An appropriate question asked by any person who contemplates introducing it into their organization is "how easy is it to introduce into my company - and what are the benefits?".

Improving the development processes in industrial companies has been one of the key issues in the SISU project from which this methodology has spawned. We have thus gained some experience in which problems occur, what some of the pitfalls are, and why some attempts have been more successful than others.

In the following we present issues, problems and advice that may seem matter of course, and indeed are, but as the obvious so often is unmentioned we include them here for your benefit. The context within which we present this, takes the view of a company with ongoing development projects, where introducing a new way of working (changing the development process) is being contemplated. We speak of change agents, which are persons that are eager to see the change happen, and we speak of an improvement program undertaken in the context of an ongoing or planned development project.

Balance is all

Changing the way we develop systems is a risky undertaking in any industrial context. We have to balance many conflicting issues:

- Investing in the future while staying alive today
- Giving time and resources to an experiment while being able to extinguish fires
- Defining a bureaucracy that keeps things tidy while supporting freedom to be creative and innovative

These are but samples of the challenges facing us. They may be overcome, but not without a fair share of planning, foresight - and enthusiasm. Nothing great has ever been done without enthusiasm!

Common pitfalls

A number of potential problems await any organization setting out to improve their development process:

Building Rome in a day

When we get ideas about improving something in our surroundings, how often do we not experience that thinking about the benefits takes so much shorter time than actually carrying out the improvement, and that reaping the fruits of our investment takes much longer than the time it takes to implement the change. And still we find ourselves getting surprised each time it happens!

Changing the way we think and work - which is what we are talking about - is probably one of the slowest changes to be carried out, possibly beaten only by changing values like tolerance to people of other cultures.

Never-the-less, we often observe unrealistic expectations in organizations concerning how quickly one expects to be awarded by changes in development methods. Experience has shown that it more often than not takes longer than expected to get a large group of people to change behaviour, and that the results are less revolutionary than one hoped for at the outset. Typically it takes several years before one can really state that people have changed their habits.

A common pitfall, even though one tries to be realistic, is to oversell - in fear of nobody buying ones ideas. “Why waste time introducing SDL into our development process if it only reduces development time by 10%?” is the question one is fearing to be asked - so why not try for 20%? But the changes are small that any single change in something as complex as a development process represents a significant influence.

This does not mean that changing is wrong! A few percent improvement each year quickly adds up to a considerable influence on the company’s bottom line. And meanwhile, your competitors are improving...

Underestimating the resistance

Inertia, or resistance to change, is a natural property for humans as it is in the world of solid objects obeying the laws of mechanics. It helps to keep things going despite the infinite possibilities that lie in alternative human behaviors.

Introducing change to the way in which humans behave in e.g. a development process <u>will</u> be met with resistance. One is often prepared for this, but not for the fact that this resistance will often be concealed, and appear in many disguises. Keen [121] calls such resistance against change counter implementation, and summarizes it as follows:

How to oppose a decided change without showing your face:

1. *Lay low*
2. *Rely on inertia*
3. *Keep things complex, hard to coordinate, and vaguely defined*
4. *Minimize the legitimacy and influence of the change agent*
5. *Exploit the lack of knowledge of the change agent*

Other forms of opposition against improvement are formulated in the Jante Law. The main message is that open opposition against change is quite unnecessary. It is better, and much less risky, to use more indirect means. Keen [121] gives several examples of games to play in this context.

Many change agents have made the mistake of relying on rational arguments to support their case. This has proved itself to be inadequate. To meet accomplished players of counter implementation one must use other means in addition. Keen [121] suggests that the change agent use counter counter implementation:

1. Make sure you have a contract for change
2. Seek out resistance; treat it as a signal to be responded to
3. Rely on face-to-face contact
4. Become an insider; work hard to build personal credibility
5. Co-opt users early

Following these suggestions, particularly item 2, can often be uncomfortable, as we naturally are apt to avoid conflicts. But experience has shown that delaying unpleasantness of this kind gives time and room for rumors and misunderstandings that can increase resistance greatly.

Mismanagement

There are two major pitfalls concerning the role of management in implementing change:

- forget to involve management
- leave everything up to management

All change requires resources. Introducing development methodologies is no exception. Many types of resources are needed: courses, tools, consultants etc., and most important: extra resources required to become prolific with the new way of working. This latter point is a “Catch 22”: the fastest way to get the job done today is to do it like you did yesterday - but then you never get to be any better!

This is illustrated in the following figure: every technology follows an S-curve; when first taken into use it is not as efficient as current practices, but gradually one becomes more prolific as one masters and improves the technique, up to a certain limit for each technology. This relationship we know to be true for basic technologies (e.g. wagons pulled by horse versus steam locomotives), and we believe it holds for software methods and tools too.

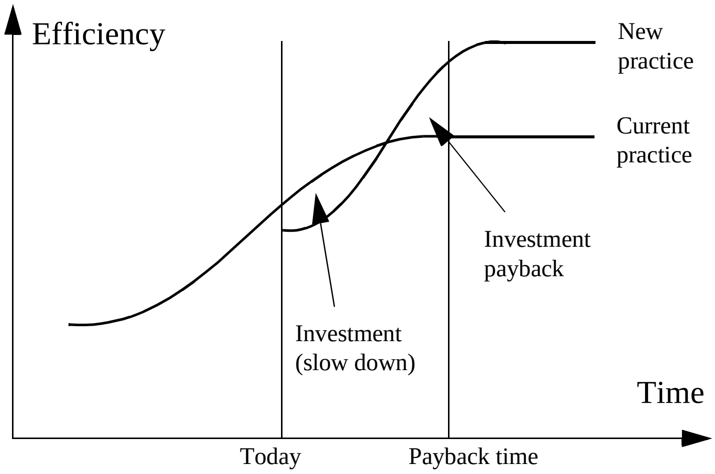

Figure 17-1: S-curves for introducing new methods

This is why management must be involved in improvement - they have to agree to slow things down and use more resources in the present, to be able to increase speed and reduce costs and defects in the future.

If management isn’t on your side, how are you going to fend off criticism and disbelief? How can you establish a contract for change without management commitment? These points are important in counter counter implementation.

The importance of management involvement and commitment - now obvious - sometimes leads to the opposite pitfall: leaving everything up to management, or implementing change top-down, that is a dictate from management not involving anybody performing the actual work. This is unlikely to be a success: management has neither enough capacity nor enough understanding of the everyday work to be able to take all the necessary initiatives.

The correct role of management is:

- Think in long terms
- Lay strategies
- Monitor and coordinate
- Support their workers
- Decide investments in personnel, training, consultants, tools
- Avoid detailed management

The individual projects and project workers must take the larger part of the responsibility for using the appropriate tools and methods!

SDL = Sex, drugs and what?

The improvement should address issues that are critical to the company's business (e.g. profitability, customer satisfaction), and that are tied to measurable goals (e.g. reduce warranty claims to less than 3% of sales), rather than issues that are only understood by software engineers (e.g. use SDL). Otherwise such process improvement efforts quickly lose the attention of management - and project workers!

Why waste time on improvements that aren't important to your company? Don't expect significant results from your improvement program if the aims are not measurable and unimportant.

Where is the code?

Introducing parts of TIMe can have significant effect on the amount of time used in earlier phases compared to previous ways of working. This is important for all to be aware of - including management. Moving from implementation-oriented to design-oriented development puts more weight on the design phases, while taking steps towards requirements-oriented development is a further shift "to the left" in the traditional phase model.

An organization accustomed to measuring progress in terms of lines of code, will not necessarily feel easy about a project working for half a year before the first line of code is written. We do not advise that such revolutions be performed overnight - but the point is to be aware that as emphasis changes, so does the types of output by which one can (and should) measure progress.

"Where is the code" could be rephrased to "Where is the domain dictionary?"

The role of training

In the turmoil of everyday work we never find the time or peace to get to learn new methods, tools and techniques. Only the few eager souls bring books and reports home for bedside reading. To change the behavior of a group of people, this is not adequate.

This is most commonly solved by using courses, that is an investment in terms of time and money to bring a larger group of individuals up to par. A few more or less obvious points regarding training:

- If the change is important and involves many people, don’t suboptimize by sending a select few; this is likely to result in a few missionaries that have seen the light - and a large group of disbelievers that will kill all enthusiasm.
- If the change involves something less concrete that a new tool for a known technique, then seriously consider the benefit of tailored in-house training. By selecting examples that are closer to the reality of the company, the added expense can very well prove to pay off. It also allows your developers to speak more freely about company issues.
- Combining consulting with holding courses is possibly the best way to implement more fundamental changes. The SISU project has had several successful examples of this.
- Timeliness of the training related to the project work to which is applies is vital. Training too long in advance of its practical use has been known to produce at best little more outcome than a refreshing break, and at worst frustration and demotivation in project workers.

Relying on tools

Software tools have long since become vital to many engineering practices, also for software developers. This is a notable exception from the cobbler’s children having no shoes. It seems obvious that a team designing a complex system using SDL must use a CASE tool that supports SDL diagrams - using pencil and paper just won’t get the job done well enough.

On the other hand, good tools are not enough; often people blame the tools for all problems encountered in a methodology, or worse: wait for some future version of a tool that perhaps will solve the problem, when the real reason is lack of understanding of the languages and techniques that the tool supports. Good tools alone will never give you the fundamental insight of the concepts that advanced systems development methodologies rely on.

Do not forget that pencil and paper are also very generic tools; so are good general drawing tools. You don’t necessarily have to wait for an MSC or UML tool to be up and running before you make your first attempt at defining the problem domain object model and its properties!

Relying on consultants

Bringing fresh people into an established setting is generally considered an advantage in connection with changing the ways of behavior in a group. In addition to hiring new people (especially newly educated), consultants - experts in the methods and tools that are to be introduced - are often seen as a ways of bringing about change.

If used rightly, consultants can indeed be a benefit, but look out for some of the traps:

- Don’t bring in consultants to “get the job done” instead of getting to grips with it your self. Some schools of management seem to rely on this type of consulting, believing that outsiders with the right competence more easily can get rid of old habits. More important possibly: leaving things to outsiders relieves the management of unpopular

or difficult decisions, that would force management to get into details they feel incompetent of handling.
We do not recommend this strategy; outsiders will never get to grips with the soul of an organization, and can at the worst mess up things that actually work. Management can use outsiders to get advice, or a second opinion, but should remain in charge themselves. This is the same advice given to organizations implementing a quality program like ISO 9001: you will not achieve quality by letting a consultant define your standards...

- Consultants are not as vulnerable to the Jante Law as insiders; this gives them more freedom to suggest new ways of treating problems. But counter implementation can easily be applied to them, particularly rule 4 and 5. Therefor management must implement counter counter implementation; the consultants can not do this!

As mentioned earlier, combining consultancy with training is probably the best use of external competence in bringing about change in a group of people.

Constipation blues - or when to introduce change

Not all times are the best to introduce change.

Sometimes one is forced to change all on-going development activities to satisfy some standard, e.g. in connection with certification according to ISO-9001; this is seldom easy or rational.

Experience shows that major changes are best introduced in the context of a major new development project. This applies particularly to small enterprises who are unable to perform experiments. The advantages are that:

- It gives everyone the opportunity of a fresh start, with all the enthusiasm this can bring about
- Change can be planned from the outset rather than introduced in the middle of a project, something that typically disrupts established plans
- It satisfies the requirement of being important to reaching company goals
- The project can be useful as a demonstrator

But beware of the risk involved! We have seen “fresh starts” which introduce new methods, new tools, new technology and new people, aiming at building a solution that surpasses its precedent in all important ways like functionality, performance, size, and price. When such a challenge falls slightly short of its ambitious aim in a few fields, it may not be the choice of methodology that is to blame. Rather one has swallowed more than one’s stomach can digest in one meal!

On the other hand, this is sometimes the way that Small and Medium-sized Enterprises have to introduce major change in order to meet their company goals; they simply have to take greater risks (see Risk assessment (p.17-35)).

Other times to introduce change can be:

- At major milestones in projects, when planning for the next phase;

- When major problems arise that could be addressed by the change - thus gaining management awareness;
- When new people with fresh eyes and new knowledge enter the scene.

The answer is blowing in the wind - the measurement problem

Elsewhere we speak much about Measurement based process improvement (p.17-24), measuring The Effect of introducing new Tools and Methods (p.17-45) and the Goal Question Metrics approach, and rightly so. Without measuring the effects of our change we cannot add facts to our conclusions.

But it may very well take 9 to 18 months - or even longer - from our decision to implement a chosen change and till we are able to measure its effect on strategic company goals such as customer satisfaction. This is a <u>very long time</u> in terms of management attention, and gives plenty of opportunity for counter implementation.

If you find this too long, and wish to avoid the risk involved, then you must find measurements or indicators that can determine how well you are doing at an earlier stage. This can for instance be done by measuring the number of findings in technical reviews or Walkthroughs - but remember then that such metrics must be analyzed, so that you have information and not simply data! Is it a good or bad sign that you record more findings in reviews? You must seek the information behind the numbers.

An other option may be to perform interviews with the developers to determine whether things seem to be improving compared to previous experience. The Wideband Delphi Method (p.17-31) is a technique that can be employed in this context.

A large company may reduce the risk of change by running experiments; small enterprises will generally not have this option.

New skin for the old ceremony: superficial change

Beware of superficial changes in behavior. It may for instance be that developers have migrated from the notations of Object Modeling Technique to the novel Unified Modeling Language - but are in their minds still thinking of Entity-Relationships, and not thinking of object types at all!

Setting goals

This chapter is about setting goals for improvement programs. It is based on experience with process improvement in Norwegian industry.

First we present a model of The Decision Process (p.17-13).

The top level of the improvement work, which sets the ground rules for improvement, is summed up in The Top Level View (p.17-14).

In The Influence Factors (p.17-15) we go on to discuss some of the environmental factors that will bear on which goals that are important, and which environmental constraints the improvement process must satisfy.

In Input to improvement (p.17-16) we discuss the data collection - the interviews - and the data analysis which will set the current improvement goals for the company under consideration, and present Process Improvement Methods (p.17-18), giving advice on How to Select an Improvement Approach (p.17-27).

Selecting Technical Improvement Goals (p.17-28) deals with coping with both quantitative and qualitative data, and presents two methods: The Influence Matrix (p.17-28) and Pair-wise Comparison (p.17-29).

We also provide An example of Setting Goals (p.17-51).

The Decision Process

The decision process can be illustrated by the following diagram:

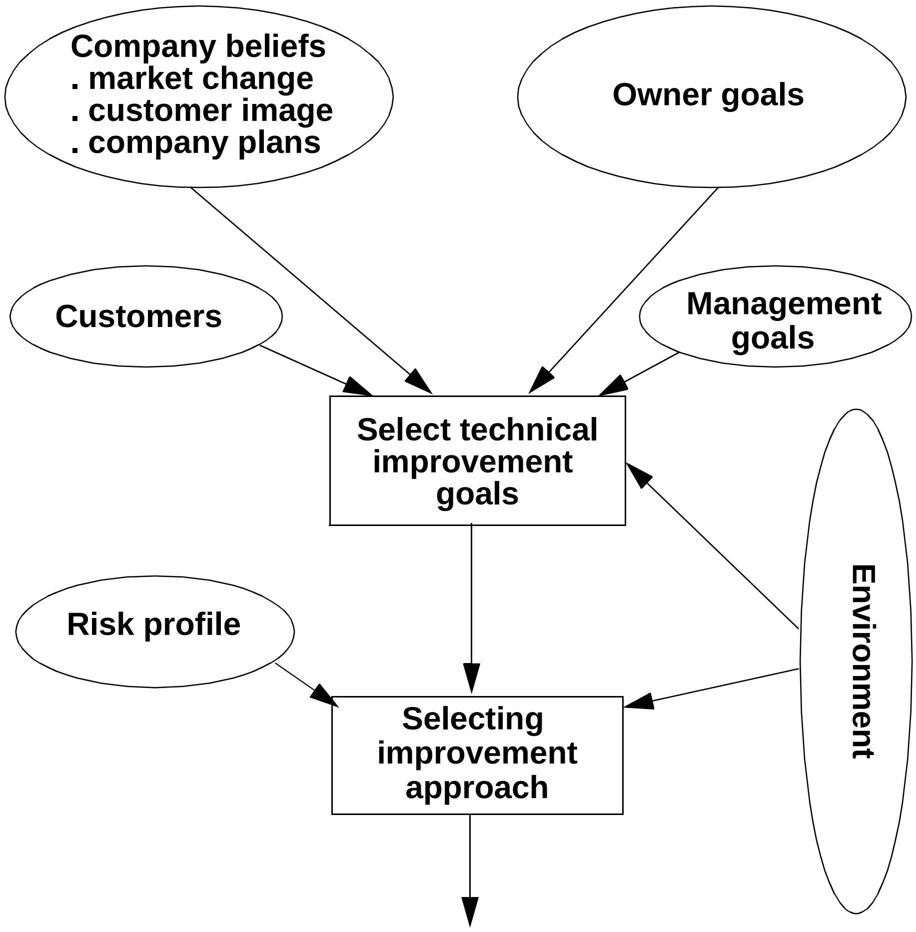

Figure 17-2: Decision process model

Each of the activities and information sources are described in the next chapters.

The Top Level View

If we, for the moment, ignore the influence of laws and regulations, the main players in the life of a company - software or other - are:

- The customers.
  They are the most important players in the field. The company's only reason for existence is to satisfy the customers' needs and wishes.
- The owners.
  The owners - seen as a group - have only one reason for being interested in the company - earning an interest on their capital. Since capital in modern economy is easy to move, it tends to flow to the companies which give the highest interest rate. The owners will, however, differ in their planning horizon - some are looking for long term security while others are looking for a quick grab.

- The management.
  The management's goals are complex since they try to satisfy two groups, namely the owners and the employees. In addition, they usually have goals of their own, such as company and personal prestige, marked share and public image.

When it is said that a company can have many goals beside that of making money and satisfying the customer, they are usually talking about the management's goals.

Since the management takes care of the day-to-day activities, their goals for the company tend to be the set of goals that enables the company to satisfy the customers and, at the same time, to satisfy their owners and in such a way that management has a good working relationship with their employees.

We are here mainly concerned with technical problems and their solution and implementation. We will thus take the company's top level goals - as they will be expressed by the company's management - as given. We should however, remember that these goals will be at a high level of abstraction. Typical goals are:

- Increase customer satisfaction by 40% within the next five years
- Lower the time to market by 30% for the next product development project, which starts in two years.
- Move into at least one new business area within the next three years.

Our job, as technical people, is to use these high-level goals to derive technical goals and develop means that can be used to implement them.

The Influence Factors

Those who shall decide on an improvement process will seldom be able to select their improvement activities only according to what they consider to be the best technical solution. They will also have to consider a plethora of external and internal influence factors that they can not influence or change - at least not within the short to medium time-frame. Some of these factors are:

- The type of company that we are working with.
  Is this a large company with large resources and many projects going on all the time? Or are we dealing with a small company that has only one project going at any time, which the company depends on in a critical way? Does the company depend on being flexible or can they operate with a static process or a few static processes?
- The type of customers and market that this company has.
  Are the customers technologically conservative or are they always looking for the latest and greatest? Is the marked filled with small, highly competitive and flexible companies? Or is it dominated by a few, big, stable or slow-moving companies? Are the customers' requirements stable, are they slowly evolving or are they changing in a more or less unpredictable manner - for instance heavily influenced by fashions?

- The technological development.
  How is the technological development in the marked where the company is operating? Is the area dominated by a large, slowly evolving body of technology, or is the applied technology moving in jumps? Is the focus on people, tools or methods and is the technology used people-intensive or capital-intensive?
- The company’s policy on innovation.
  Well-managed companies can usually be split into two groups, labeled as technology leaders and strong seconds. The technology leaders always look for the best and newest and try to stay ahead of their competitors by always trying to be first with new products. The strong seconds, on the other hand, relies to a large degree on competitor analysis and market analysis in order to go in where there seems to be an opening for a new product or service.
- The time-frame of operation.
  The time-frame of the improvement process is given by the time constants in the company’s operation. If the company’s environment is changing drastically, say every three years, the improvement cycle must be considerably shorter; this in order to influence the way the company works. If the environment - technology, products and customers - is changing continuously, then improvement has to be attempted on a level that is quite different from what we can attempt if the environment is stable or only changing slowly.

All these factors must be considered when we select improvement goals. If we ignore one or more of them, we run the risk of doing more harm than good. For a small company in a highly competitive market, the selection of improvement goals might be a question of survival or bankruptcy. Thus, the decision should not be taken too easily.

Input to improvement

We distinguish between the input related to the overall policy of the company, which we call The First Improvement Input (p.17-16), and The Second Input - the Interviews (p.17-17) with the customers and management.

The First Improvement Input

The most important input for an improvement program is the company’s ideas on the following questions:

- How is the company and its products viewed by its customers at the present?
- How do we expect that the market and the types of products that we sell will change during our next planning period?
- Where do we want to be at the end of the next planning period? This question concerns market area, technology, product type, price and quality segment.

All this will influence how much we are willing to invest in improvement for the next planning period. The investment will, among other things, depend on which other investments we foresee in the period and what the expected income generated from these investments is.

Part of this information is readily available, while other parts of the information needs to be collected. For a large company, this information is available from the marketing department, the company's management board and the company's economics department. For SMEs, however, this information is seldom readily available. Many of the companies in this category are much more market driven than plan driven. Thus, company plans are kept at the barest minimum. In these cases, the first improvement input will be rather short.

The Second Input - the Interviews

When the overall policy input to the improvement process is documented, we should start with the second part of the information collection, that is: the interviews. In our experience, this will supply the most important data for setting improvement goals.

Interviews with the Customers

If the company has a large and active marketing department, this part can usually be left out. Otherwise, it is of major importance. By interviewing several customers, it is possible to find out how “our” company's products are viewed in the market. Are they for instance considered to be reliable or failure prone, are the customers satisfied with the company's service level, pricing policy and so on?

The data collected in this way should be combined with data from interviews with those persons in the company that receive customers complaints. In addition, we should include information from the persons or companies and agents who sell or service the products we are considering.

The report that summarizes this information should be the first input to the company's management. This will help them to adjust their own view of the company's reputation in the market and increase the probability of selecting a set of appropriate improvement goals.

Follow-Up Interviews with Company Management

The management can now combine the information concerning customers and market, and combine it with their company's knowledge in important areas, as discussed earlier.

When all this information has been digested, we should do a follow-up interview with management. This is necessary in order to:

- Verify our understanding of the company policy. It is, for instance, important to understand what the company considers to be its main competitive advantage.
- Verify our thoughts and hypothesis on customer requirements at the present and at the end of the improvement period. This concerns, among other things, the marked and product stability.

The results of this will define the border conditions for our improvement goals. Without them, no real improvement is practically feasible.

Process Improvement Methods

There are several ways to improve any process. Broadly speaking we can split these methods into four groups, namely:

- The "Mean and Lean" Approach (p.17-18). This is based on a continuous hunt for activities that do not add value to the product. Such activities are removed in order to make a lean development process. This approach is strongly related to "Black Box" based Process Improvement (p.17-25).
- Capability Maturity Model (CMM) (p.17-19) has defined five maturity levels and a set of practices that must be in place in the organization before it can be considered to be at this level. Improvement according to CMM consists of an assessment - where are we - and an improvement - where do we go from here? The answer to this question is to implement the key practices that are needed to reach the next level in the CMM ladder.
- The Risk Management Approach (p.17-21) which builds on two items - an analysis of the gap between where we are and where we want to be and the change process needed to bridge the gap. For each step in the change process, the method attempts to identify and manage or control possible risks that can occur.
- Measurement based process improvement (p.17-24). Improvement methods that are put into this category are based on collection and analysis of data pertaining to the product and its development process. This improvement category can again be split into two groups - called "Black Box" based Process Improvement (p.17-25) and "White Box" based Process Improvement (p.17-25) respectively. The names refer to how you view the process.
  Closely related to the measurement based methods is the Goal-Question-Metric approach.

The choice of improvement method will depend on the stability of the company's environments and its time-frame of improvement and operation, as we explain in How to Select an Improvement Approach (p.17-27).

The "Mean and Lean" Approach

Not all methods require that you are able to perform measurements of your process in order to improve. The "mean and lean" method for process improvement has as its main principle to hunt for costs and activities that do not contribute to the final product and thus should be removed. By removing these costs and activities the process will improve. These costs can be split into two large groups as follows:

- Quality costs.
  These are costs concerned with error correction. Their name stems from the fact that these costs occur because we did not get the quality right the first time. Thus, such costs are also called rework costs. Ideally, they can be removed completely.

- Cost of unnecessary activities.
  These are costs concerned with work that should not have been done in the first place. The activities connected to these costs may not be un-useful per se - they just do not contribute to the value of the product.

The “mean and lean” approach takes as its starting point all activities that occur in a project. For each activity, ask the following questions:

- Why is this activity performed?
- How does this activity add to the value of the product?
- What will happen if the activity is removed from the process?

In some cases it will be quite clear why the activity is there and what will happen if it is removed. In other cases, the answers are not at all clear.

For some activities, although it is quite clear that the activity is needed - the question is instead how much resources it should use. This is for instance the case for code reading - most people agree that one cannot expect increasing benefits from any increase in code reading effort. The law of diminishing return applies here, as everywhere else.

The “mean and lean” approach to process improvement is most efficient on a medium scale of time frame and stability of environment. The approach can, however, also be applied in other situations, except where the time frame is short and the stability is low. In these cases, things will be much too unstable for any improvement.

It may be advisable to always look for improvement opportunities and make the “mean and lean” approach a way of life.

Capability Maturity Model (CMM)

The framework

The CMM is a framework describing an evolutionary improvement path based on five maturity levels, with each level providing a foundation for the next. Key process areas indicate where an organization should focus to improve its software process. To achieve a maturity level, all key process areas for that level must be met, as well as key process areas required for all lower levels. The underlying assumption is that, in general, organizations at level n+1 produce better software that organizations at level n. Figure 17-3 "Maturity framework" (p.17-20) shows the five maturity levels and their key process areas.

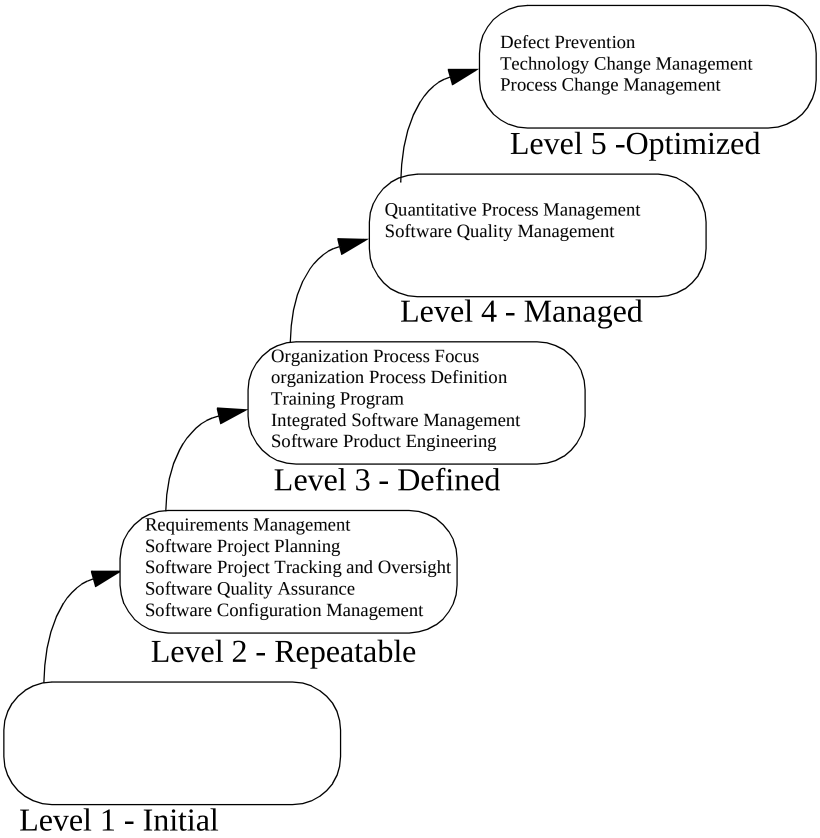

Figure 17-3: Maturity framework

Below, characteristics of each level are listed:

- Level 1: Initial
  Few Processes defined; capability is characteristic of the individual, not the organization.
- Level 2: Repeatable
  Basic project management processes established; earlier successes repeatable.
- Level 3: Defined
  Common, organization-wide understanding of activities, roles and responsibilities.
- Level 4: Managed
  Quantifiable and predictable; process is measured and operates within quantitative limits.
- Level 5: Optimized
  Continually striving to improve the range of process, the performance of projects.

Using the CMM

The CMM establishes a set of public criteria describing the characteristics of mature software organizations. These criteria can be used by organizations to evaluate their software capability and to improve their processes for developing and maintaining software.

- The software capability evaluation is focused on identifying the risk associated with a particular project or contract for building high-quality software on schedule and within budget. This evaluation is usually performed by a third party and can be related to an ISO-9001 certification. In the US, the CMM-rating is as important as an ISO certification is in Europe. For example, the US DoD requires organizations to be on level 3 before setting out contracts.
- The foundation for improving the software processes is a software process assessment which focuses on identifying improvement priorities within the organization's own software process. The result can be used to plan an improvement strategy.

The CMM establishes a common frame of reference for performing software capability evaluations and software process assessments. An important first part of this work is to complete the maturity questionnaire. Based on the responses to these questions, key process areas for further exploration are identified. The next step is to visit the site to be assessed or evaluated and perform interviews, reviews etc. The result of the work is a profile of satisfaction of the goals within the key process areas and findings that identify strengths and weaknesses in terms of key process areas.

The CMM identifies practises for a mature software process. It defines which key process areas that have to be met in order to achieve a maturity level. Note, however, that a key process area only defines "what" goals that have to be met, not "how" they should be met. The model is prescriptive and not oriented towards the organizations needs. This means that you do not need to collect data and establish a baseline before getting started. It also means, however, that whatever short-term needs you have, you have to satisfy those defined on lower levels before they can be considered.

Achieving higher levels of software process maturity is incremental and requires a long-term commitment to continuous process improvement. Software organizations may take ten years or more to build the foundation for, and a culture oriented toward, continuous process improvement. Although a decade-long process improvement program is foreign to most software companies, this level of effort is required to produce mature software organizations according to the CMM.

It should be noticed that the CMM does not address all the issues that are important for successful projects. For example, the CMM does not currently address expertise in particular application domains, advocate specific software technologies, nor suggest how to select, hire, motivate, and retain competent people.

The Risk Management Approach

The risk management approach to process improvement consists of the following steps:

1. Where are we now concerning the goals - for instance productivity, TTM (Time To Market) or operational reliability?
2. Where shall we be when this improvement process is finished?
3. How can we bridge the gap between these two states?
4. Which factors help us and which factors hinder us in reaching our goal?

The first two steps are more or less the same as we find in QIP (Quality Improvement Paradigm). The last two steps are, however, quite different.

The rest of this section is organized as follows:

First we shortly describe The Kiviat Diagram (p.17-22) and The Ishikawa Diagram (p.17-23). Then we describe a method for Handling uncertainties (p.17-23); see also Risk Assessment and Control (p.17-33) and a worked-out example, An example on Risk Management (p.17-57).

The Kiviat Diagram

The first step is the gap analysis. A convenient tool here is the Kiviat diagram. For our purpose, this diagram is constructed as follows:

- We start with a circle. All goals - where we want to be - are represented by points on the circle's perimeter.
- There is one axis per area for our gap analysis. On the axis, we mark the current status for each area. This can be based on either an objective or a subjective assessment.
- The points representing the status for each area are connected with a line in order to make is easier to grasp the total picture.

The following diagram shows a simple example with eight goals.

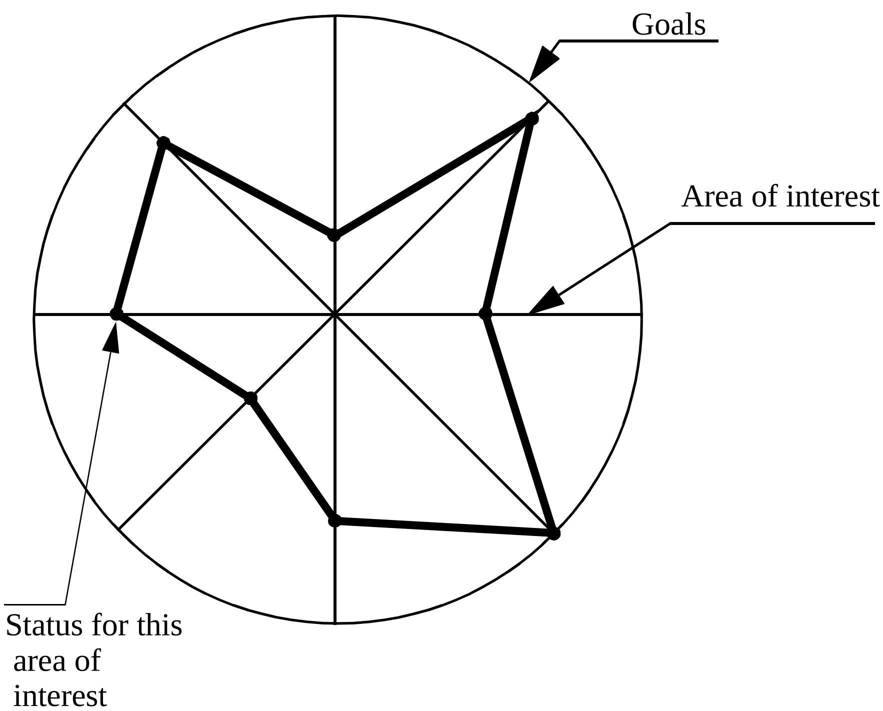

Figure 17-4: Sample Kiviat diagram

From the total view presented by the Kiviat diagram, we can select our top level goals. Once these top level goals are selected, they can be further analysed by using The Ishikawa Diagram (p.17-23).

The Ishikawa Diagram

It is common to use a fish bone diagram - Ishikawa diagram - for further breakdown of our goals. We will - for the sake of argument - assume that all improvement goals can be stated as: “Improve A”. This statement will serve as the goal for our diagram:

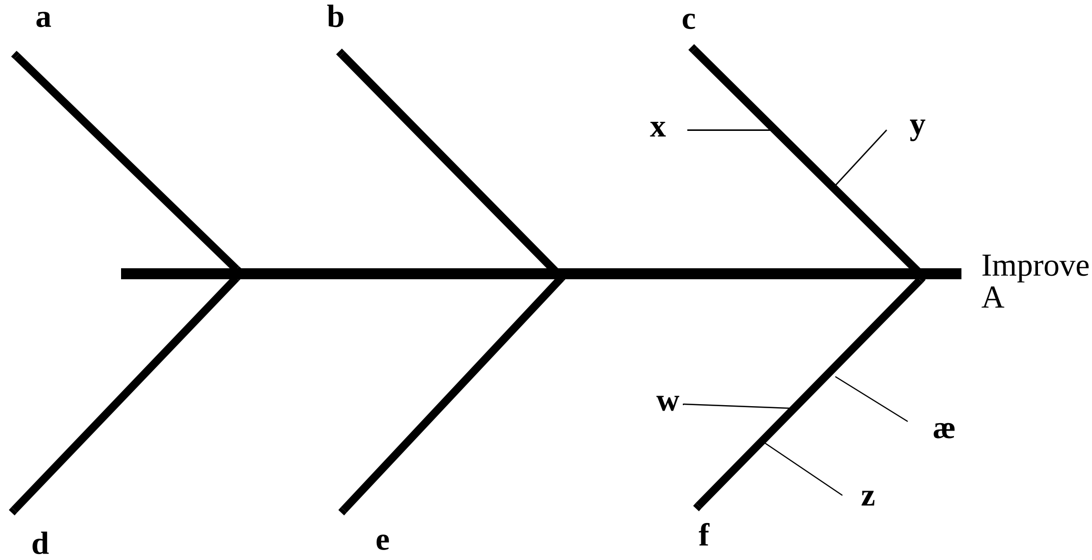

Figure 17-5: Ishikawa diagram

The Ishikawa diagram is - in the general case - a multi-level description of cause and effect. The diagram above should be read as follows:

- In order to improve A we must achieve a, b, c, d, e and f.
- In order to achieve c we must achieve x and y
- In order to achieve f, we must achieve w, z and æ.

The diagram is a way to organize a body of knowledge and ideas. Possible sources of knowledge and ideas are:

- Idea-generating methods, like brainstorming sessions and the affinity diagram method (KJ method) [Mizuno88].
- Interviews with developers in the company. The GQM (Goal- Question-Metric) work sheet can be used here.
- Literature search. This will give both information - what has worked at other sites - and ideas - what are the current theories in the area of software process improvement.
- Benchmarking - what are the market leaders doing, and how?

When we have an agreed-upon Ishikawa diagram, we can use pair-wise comparison to rank the main activities - denoted ***a*** to ***f*** in Figure 17-5 "Ishikawa diagram" (p.17-23) - according to their importance for the goal.

Handling uncertainties

Since the activities that we will use to achieve our improvement goal are not based on experiments, we need to control two types of uncertainties, namely:

- Uncertainties concerning the positive effect of each activity.
- Uncertainties concerning the negative effects connected to each activity- the risk.

In order to increase the probability of the positive effects and reduce the probability of the negative ones, we need to know the factors that affect the outcome of initiating the activity - the influence factors. The Ishikawa diagram can also be used at this level. The only change from its use in setting improvement goals, is that the main effect - denoted “Improve A” in the previous example - now will be for instance “Hinder X”.

When we have an agreed-upon list of influence factors, the factors must split into two groups as follows:

- Promoting factors and events:
  These are factors and events that will help us to get the maximum positive effect out of the proposed change. By increasing these factors we will shift the probability distribution of the activity to the right - towards better, more positive impacts.
- Hindering factors and events:
  These are factors that will inhibit or reduce the effect of an improvement activity. By reducing these factors or preventing the events from taking place, we can reduce the risk of low or negative effects and thus again shift the probability distribution of the activity to the right - towards better, more positive impacts.

The promoting factors will be part of our introduction strategy. The hindrance factors will be input to risk management.

If we are not able to obtain a satisfactory risk profile for our improvement changes after this, we must perform experiments. This implies that we - at least for the moment - leave the risk management approach and select a measurement based approach instead.

Measurement based process improvement

The following principles lay behind measurement based process improvement:

- Improvement must start by establishing a baseline. This baseline describes the current status. We need this in order to know where we are at the present. Without this, we can improve, but we would never know that we have improved or how much.
- If we shall improve something, there must exist one or more goals for the improvement. These goals must be stated in an objective manner. In the start of the process, it must be possible to measure where we are pertaining to these goals and later on, it must be possible to observe how we have changed.
- Before we improve something, we must understand it. Otherwise, our changes could wind up being little more than random manipulations, supported by a more or less imprecise set of gut feelings. Without understanding, our attempts to improve might be a haphazard affair with low efficiency and a small return on investments.
- We must have one or more methods that can be used to analyze our data. The important point here is as follows: If two observations of the same parameter show different values, how different must they be before we are willing to say that the difference is not caused by chance? Such a method is necessary if we shall be able to say that our improvement activities have had any effect.

The following sections will shortly describe two approaches to measurement based process improvement, “Black Box” based Process Improvement (p.17-25) and “White Box” based Process Improvement (p.17-25). Closely related to these measurement methods is the metrics method GQM.

“Black Box” based Process Improvement

Black Box based process improvement uses only measurements that are available from outside the project. As a result of this, the improvement activities must be based on general process knowledge, published papers etc. We provide an example of The “Black Box” Improvement Scenario (p.17-49).

Typical data to measure when doing “black box” improvement are for instance:

- Development cost per size unit delivered, such as number of SDL symbols, lines of code, lines of SDL/ PR or function points
- Number of problems experienced by the customer during the first year of operation
- Time To Market (TTM), e.g. measured as the calendar time elapsed from initial customer contact to customer system acceptance
- Life Cycle Cost (LCC), which includes all costs accumulated over the total life span of the system.

The improvement process will be as follows:

1. Decide on the variables that the company wants to improve - for instance the number of problems in a system when it is delivered to the customer.
2. Find the current status, for instance the value of one or more target variables, either as a single value - represented by the current gross average - or its distribution - represented by e.g. its mean value and the six sigma limits.
3. Set goals for the target variables and decide on improvement actions - including time limits for the effect to manifest itself. Typical actions can be to send all programmers on a course in language X, buy a new CASE tool or introduce new development methods. A typical time limit can be one year.
4. Check effect of changes by observing the parameters when the trial time has expired. If the changes introduced in step 3 were successful, keep the changes. Otherwise undo the changes. In all cases, repeat the process from step 1.

“White Box” based Process Improvement

White Box process improvement is based on process understanding. Thus, the work pertaining to modeling and data collection is larger. As a result of this, the initial investment is higher. On the other hand, the resulting improvement process will be more efficient.

The White Box improvement process has the following basic steps:

1. Characterize the development process. This includes customer relations, management influence and external influences. In addition, it includes the process under consideration, the customer’s requirements, the project personnel available and the tools and methods used.

2. Understand the development process. This is best done by making hypothesis pertaining to the process, collecting relevant data and using one or more statistical methods. The main challenge with this approach is the treatment of the factors discussed in step 1, which can not be controlled but only observed.
3. Package and store the information obtained in steps 1 and 2 in the company's experience base. Note that it is the combined information that makes sense. Thus, all knowledge is stored as “For a project with such and such characteristics, such and such relations are known to hold”.
4. Set improvement goals. In the general case, these can be set related to external measures such as product reliability or project productivity. They can, however, also pertain to project internal characteristics, such as the amount of resources used for reviews or the efficiency of the integration testing activity.
5. Define the changes that we need, based on the understanding obtained in steps 1 and 2 and the goals stated in step 4, in order to reach - or at least approach - our improvement goals.
6. Check the effect of the changes in step 5 to see if we are moving in the right direction - as defined in step 4.

It is important to remember that all the understanding achieved in step 2 in the general case only is relevant for projects that match the characterization obtained in step 1. Thus, the question “How relevant is these data to our project?” depends on the answer to the counter-question “Is the characterization of your project close enough to the characterization of the project that we have used as a basis for our understanding?”.

In order to answer this question, we must be able to assess the closeness between two or more development projects.

How to Select an Improvement Approach

The improvement approach that we select is mainly a function of the two variables "company time frame" and "stability of market and technology". The role of the suggested approaches can be placed in a two-dimensional diagram as follows:

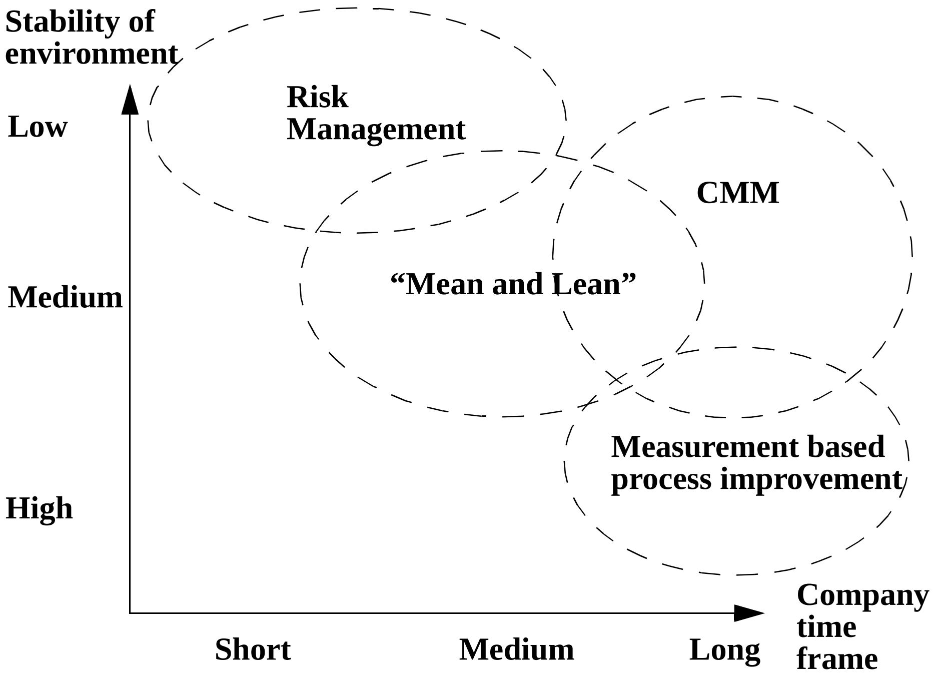

Figure 17-6: Improvement approach depending on time frame and stability

In order to select an improvement approach, we need to decide:

- What is our time frame? Can we wait four years for results or do we need to improve tomorrow?
- How stable is our environment - market and technology? Will we be operating in the same market and using roughly the same technology in four years time or do we foresee large changes?

Based on the diagram above, and the answers to the two questions, we can select an appropriate improvement approach.

Selecting Technical Improvement Goals

From the steps described in the earlier chapters, we get a potentially large number of possible improvement goals at the technical level. In order to run experiments, collect data and so on, we need to have just a few improvement goals, say two or three. The main problem confronting us when doing this selection, is that we have none - or just a few - quantitative data. The rest of the information available for the decision is qualitative. Thus, the methods that we can use at this stage of the process must be able to cope with both quantitative and qualitative data.

We will discuss three methods - The Influence Matrix (p.17-28), Pair-wise Comparison (p.17-29) and The Wideband Delphi Method (p.17-31).

The Influence Matrix

The influence matrix shows the coupling between decisions and influence factors - effects of the decisions.

Influence factors are areas of concern to the company. They are called influence factors since they will influence the well being of the company. Influence factors can be for instance software quality, productivity, time to market or market share.

The influence matrix is a way to combine several types of information, namely:

- Possible alternative decisions, denoted by *As*
- The factors of a decision that we want to consider, denoted by *Fs*
- The influence of one or more facets of that decision on the company or a selected part of it, denoted by a marker. The marker can take the following values:

  ++: A highly positive effect on the factor under consideration

  \+ : A positive effect on the factor under consideration

  0 : No effect on the factor under consideration

  \- : A negative effect on the factor under consideration

  -- : A strongly negative effect on the factor under consideration

The selection of influence factors is critical. If somebody does not like the conclusion it is always possible to point to an influence factor that has not been included and which will influence the decision.

In order to get a clean and open decisions process, the influence factors should be decided before the rest of the process. If the factors will be weighted in order to reach the final decision, the weights should also be agreed upon on beforehand.

The influence matrix is described as follows:

Table 17-1: Influence matrix

	Influence factors			
Decision alternatives	F1	F2	...	Fn
A1				
A2		marker		
...				
Am				

There is no algorithm that can help us to decide which marker to use where and under which circumstances. This will depend entirely on the knowledge, insight and intuition of the persons who fill in the markers in the matrix.

In addition, there is no standard method that can be used to sum up the resulting influence of selecting a certain alternative. Thus, the decisions has to be made based on the impression of all the information presented.

We provide a small, but realistic example, see An example of Setting Goals (p.17-51).

Pair-wise Comparison

The pair-wise comparison method does not require us to split the effects of each decision up into influence factors. Instead, each decision is compared against all other decisions based on a total impression of the effects of the decision.

The table used for evaluation and ranking the possible decisions is shown below:

Table 17-2: Pair-wise comparison

Alternative	Comparison counts					Sum
A1	Σ(A1 bt Ai)					S1
A2	A2 bt A1	A2 > i				S2
A3	A3 bt A1	A3 > A2	A3 > i			S3
...	...	...	...	...		...
An	n > 1	n > 2	n > 3	...		Sn
Sums	n - 1	n - 2	n - 3	...	1	n(n - 1) / 2

The notation “Aa bt Ax” means that alternative Aa is better than alternative Ax. If we just look at the first of the comparison columns, we have that:

- “Σ(A1 bt Ai)” denotes the number of alternatives that is assessed as being less attractive than A1
- “A2 bt A1” is set to one if alternative A2 is more attractive than alternative A1 and is set to zero otherwise
- “A3 bt A1” is set to one if alternative A3 is more attractive than alternative A1 and is set to zero otherwise.

The sums on the right-most side of the table - the S-values - are the sums of all entries on each line. The alternatives are ranked according to their S-values. The sums at the bottom of the table are just used to control that the table has been filled in properly.

If two alternatives are considered to be equal, it is customary to give a score of 0.5 to both alternatives. For instance, if A1 and A2 are considered equal, then “A2 bt A1” is set to 0.5 and “Σ(A1 bt Ai)” is incremented by 0.5. This will not influence the control sums at the bottom of the table.

A small example to clarify all this.

Assume that we have three alternatives A1, A2 and A3. Let us also assume that A1 is just as good as A2, that A1 is better that A3 and that A2 is better than A3.

Table 17-3: Example of Pair-wise comparison

A1	1,5			1,5
A2	0,5	1		1,5
A3	0	0	0	0
Sum	2	1	0	3

The Wideband Delphi Method

Data Collection

The Wideband Delphi method [Boehm81] is a method that can be used to obtain subjective estimates. In Table 17-4 "Wideband Delphi estimation form" (p.17-31) an example of an estimation form is shown which can be used to estimate a lowest value (L), a most probable value (M) and an upper value (U) of a parameter with the Wideband Delphi method.

Table 17-4: Wideband Delphi estimation form

Estimation of a parameter	Marker								
	0	20	40	60	80	100	120	140	160
Lower limit (L)		*X	!	XX					
Most probable value (M)			*	X	!	XX			
Upper limit (U)				*	X	!	XX		

We have used the following conventions for the markers:

- “*”: your estimate
- “!”: median estimate.
- “X”: other team members’ estimates.

The estimates are provided by a group of experts and a moderator. According to the Wideband Delphi method the work proceeds as follows:

- The group of experts should first meet and discuss the estimated parameter and any related issue.
- Each expert then makes an individual estimate of the unknown parameter and fills in the form.
- The moderator compiles the results and distributes new forms to the experts, now with the other experts’ last estimates marked.
- When these forms are distributed, the experts should meet again and discuss their results. When this is done, the experts, again anonymously, make new estimates and fill in forms again.
- This procedure is iterated until the estimates do not converge any further.

There are two different versions of the Delphi method, the Delphi method and the Wideband Delphi method.

- Wideband Delphi:
  The new estimates are discussed in a meeting following each round of estimation.

- Regular Delphi:
  Each expert must state why he thinks that the parameter should have the proposed value and why higher - or lower - values are incorrect.

In both cases, it is important to make all arguments visible to all participants so that they can understand the reason for the other experts' estimates and use these examinations to build a better mental model for themselves and thus improve their estimates.

Presenting the Results

The results from the Wideband Delphi process can be presented in several ways. However, we suggest that you use one of the two methods shown below:

Diagram presentation

The diagram below shows the probability distribution of the variable X.

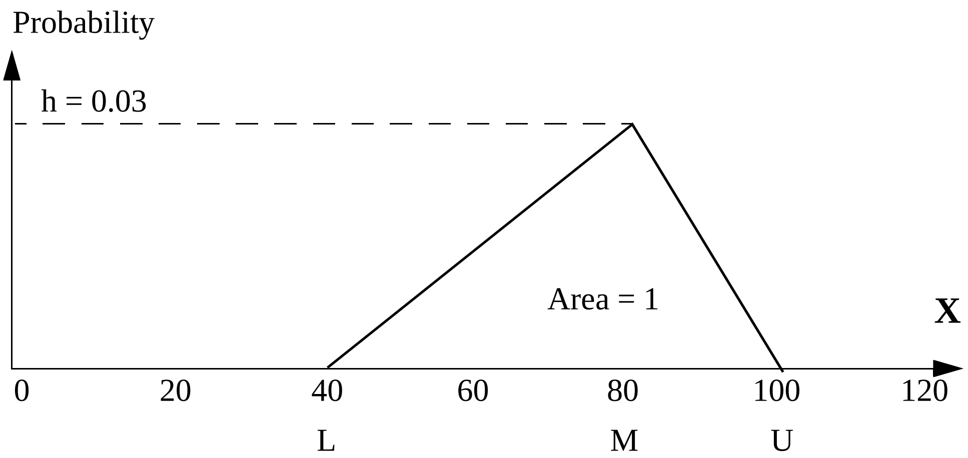

Figure 17-7: Wideband Delphi example

The maximum probability - h - is computed from the relation h*(100 - 40) / 2 = 1, that is, the area of the triangle shall be one.

Statistical characteristics

Here we estimate the mean value (μ) and the standard deviation (σ).

$$\mu = \frac{Upper + 4 \bullet Median + Lower}{6} \qquad \sigma = \frac{Upper - Lower}{6}$$

For the example in Table 17-4 "Wideband Delphi estimation form" (p.17-31), this gives us μ = 77 and σ = 10.

In the general case, we can not expect the Delphi process to give a symmetric distribution. Thus, if we need upper or lower, say 5%, limit we must use the method shown in How to use the Triangular Distribution (p.17-41).

Risk Assessment and Control

In this chapter we shall look at Risk in General (p.17-33), A Company's Attitude to Risks (p.17-33), Risk assessment (p.17-35) and Risk control (p.17-38).

Risk in General

All improvement implies changes and all changes imply a certain amount of uncertainty, which implies risk. On the other hand, by the time an improvement change is fully researched, well documented and understood, there is little competitive advantage to be gained.

Changes related to such improvement must, however, still be done, in order to stay in the market and be at least as good as "everybody else". In Japanese, the word for risk is translated to "dangerous opportunities", which describes the idea pretty well.

In the general case, risks can be split into two groups, namely:

- Internal risks.
  These risks are introduced through our own actions and decisions and can, at least in principle, be understood and controlled given enough resources and time.
- External risks.
  These risks are introduced by our environment, such as our customers, our competitors and the government. These risks are by and large completely outside our control.

Another way to look at risks is to split them up according to how we plan to handle them. As shown in Figure 17-8 "Difference in risk handling profile" (p.17-34), we can split them into:

- Risks that can be removed by understanding them. This requires extensive experimentation, data collection and data analysis which will result in risk avoidance activities, such as process changes.
- Risk that can be controlled through a set of risk controlling activities.This might also require a certain amount of data collection and analysis, which will be used to establish process control limits.
- Risks than can neither be removed nor controlled. Such risks can either be accepted and managed or transferred, for instance to an insurance company or to a subcontractor.

A Company's Attitude to Risks

In general, a company's attitude to risk is determined by its environment. The environment will decide both the risks that the company needs to take and the possibility of doing something to reduce the risks. The following characteristics are important:

- Type of company:
  Large companies with ongoing projects can run many experiments and obtain a large amount of data, making it possible to obtain statistically significant results. Small companies will have few projects and can thus obtain few data which again implies only vague information.
- Type of customer:
  Large, stable customer implies that the set and type of requirements will stay fairly stable. This increases the reusability of data and experience. For companies that have customers that focus on "latest and greatest", flexibility and personal service, many kinds of data and experience will be outdated rather quickly.
- Technological development:
  Slow, predictable technological development will make it possible to keep the development process stable or at least close to stable, while a quick, almost erratic technologically development may make most experience and data irrelevant and outdated in two to four years.
- Operational time-frame:
  The time between major organizational and technical changes will influence the lifetime of experience. Slowly evolving companies will have a much larger possibility to build up large amounts of valid experience and data.

The following diagram can illustrate the situation for two typical combinations of characteristics.

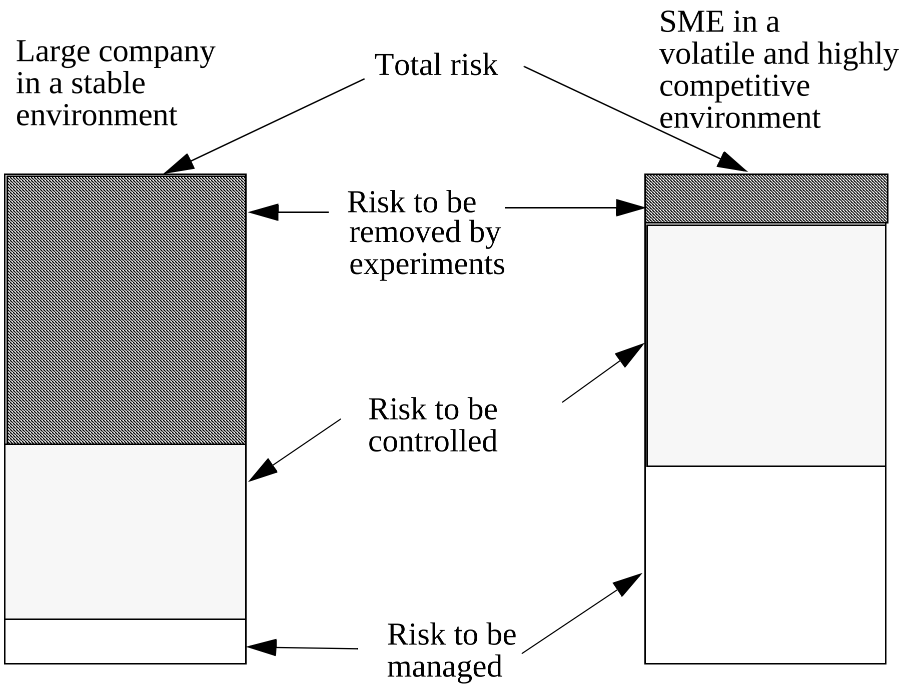

Figure 17-8: Difference in risk handling profile

The diagram above is based on the assumption that the two companies have the same total risk. In most cases, the situation is different, since the faster pace of change in SMEs will make the risk larger for this type of company.

Small companies inevitably end up taking more risk than large ones. For this reason they more often go bankrupt, while on the upside they are more innovative and break more barriers if they succeed.

Risk assessment

As mentioned earlier, the main problem with process improvement is an unstable environment or short planning horizon, is the need for data in order to remove risks.

The most important data during process improvement are data pertaining to the effect of a change. Usually we are interested in the effect of a process change on productivity, quality or time to market. The effect scale can be percentage change or an absolute change.

If it is difficult to asses numerical values to the effects of a change, we can use a qualitative "value" instead, such as "light", "low" etc.

There exist some methods that can be used to obtain at least approximate quantitative data. The most popular method in this area is The Wideband Delphi Method (p.17-31). Out of the Delphi method or any related method we can get a distribution of effects. In its simplest form it consists of just a triplet containing the lowest estimate, an estimate of the expected, most probable value and the highest estimate.

Graphically it is convenient to show this as a triangular distribution. This distribution is defined in the following way:

- The estimates of both the highest and lowest value are given probability equal to zero. This implies that we assign a zero probability to the two events "x > highest estimate" and "x < lowest estimate".
- The expected value is given the highest probability. Since we make a probability distribution, the distance from the x-axis to the top of the triangle is determined by the requirement that the area of the triangle shall be 1.
- The probability distribution diagram is drawn by connecting the lowest (min), expected and highest value (max) as shown in Figure 17-9 (p.17-36) below.

The probability of obtaining any effect can be computed from this diagram. See How to use the Triangular Distribution (p.17-41) for an explanation.

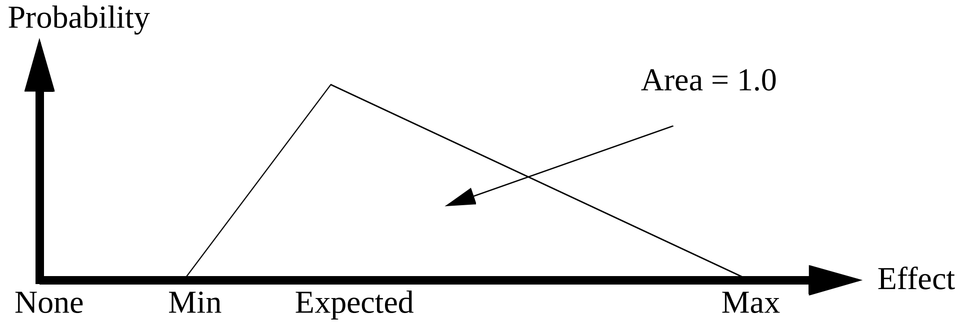

Figure 17-9: Example of a triangular distribution

The shape of the distribution and its relationship to the no-effect point will decide whether the change effect described by this distribution is appropriate or not. The following are some examples of the shape of the probability distribution of the effect w.r.t. a change activity in a process improvement setting.

In many cases, it will be difficult or downright impossible to assign numerical values to the effect of an improvement activity in a risk management based improvement scenario. In this case, the effect axis could be graded as high, medium and low - both positive and negative. This will give a diagram that for instance could be as follows:

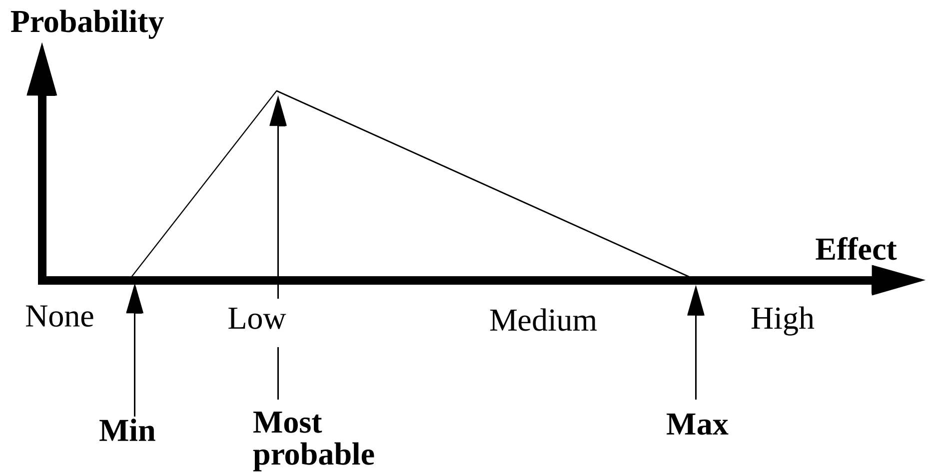

Figure 17-10: Example: A triangular

There are two ways to interpret the “None” point in the diagram. This point may be chosen as the point where the activity

1. Has no effect
2. Has an effect equal to the cost of implementing it.

The method is valid in both cases. We recommend that you try to assess the net effect of the change.

A small example will show you the general idea. Let us assume the following:

- We are studying a new tool for testing. The cost of this tool is 10000 $.

-We are running an average of three test campaigns per year, costing a total of 30000 $.

-The effect of the tool has been assessed through a Delphi process as follows.

A 33% reduction in cost will cause us to break even in the first year. By using a simple calculation, we can also find that there is a 28% probability of loosing money, while there is a 33% probability of saving $2000 or more.

Other examples are shown in the following figures:

The cause described in Figure 17-11 (p.17-37), Figure 17-12 (p.17-37) and Figure 17-13 (p.17-38) are related to changes where it is not possible to assign numerical values neither to the cost nor to the effect of the change. The shaded area shows the probability that the cost of the change is greater than the effect.

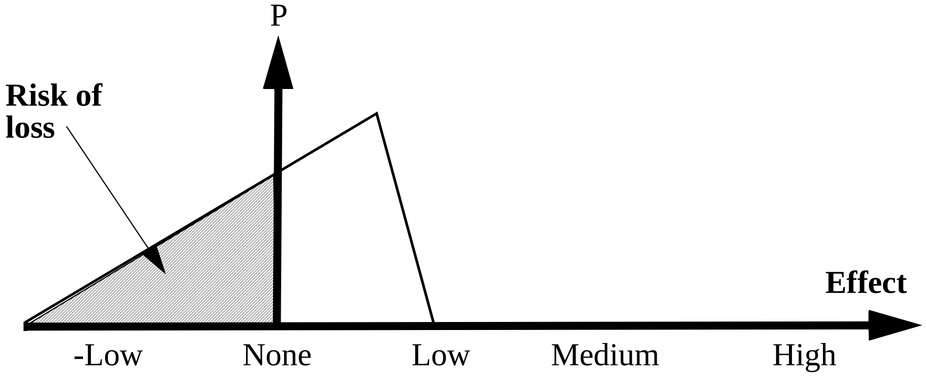

Figure 17-11: Example: Probably small effect, medium risk

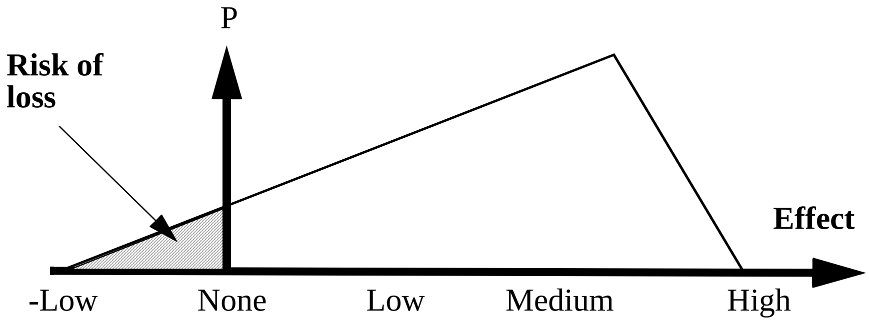

Figure 17-12: Example: Probably large effect, small risk of loss

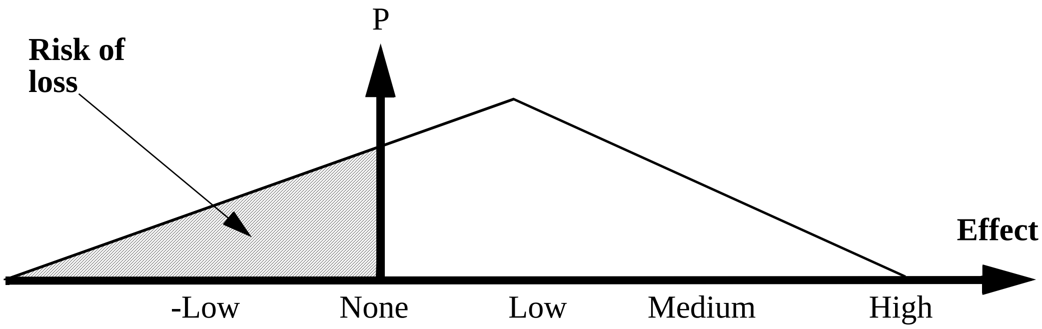

Figure 17-13: Example: Probably small effect, but large effects - positive and negative - are possible

Common for all these examples is that there are possibilities both for large - positive - effects and low or even negative effects, which will cause losses to the organization. In connection with this there are two problems that are important:

- How can we increase the probability of getting the best out of the change, i.e. moving the outcome towards the most effective results?
- How can we discover early that we are bound for a low or negative effect. In this case, we also need to consider contingency activities that will help us to cut our losses.

Some answers to these problems are discussed in Risk control (p.17-38).

Risk control

The risk control has three important activities:

1. Identify possible risks and their causes.
2. Assess their consequence and probability.
3. Identify possible responses for each risk. These responses will fall into one of three categories, namely:
   Responses that remove the risk from the activity.
   Responses that prevent the risk form happening.
   Responses that help us to control or contain the risk throughout the whole or critical parts of the process.

The information can conveniently be organized as shown in the table below:

Table 17-5: Risk Assessment and Control Table

Activity	Identi-fied risk	Estimates		Cause	Response
		Proba-bility	Conse-quences		
A1	R1.1	P1.1	C1.1		
	R1.2				
	R1.3				
A2	R2.1				
	R2.2				
	R2.3				

The following notation is used:

- A<i>: Activity identifier for activity number i.
- R<i.j>: Risk item number j for activity number i.
- P<i.j>: The assessed probability of the occurrence of risk R<i.j>. This probability can be quantitative - for instance 0.3 - or qualitative - for instance “Medium”.
- C<i.j>: The consequence if risk R<i.j> occurs. This consequence can be quantitative - for instance USD 10 000 - or qualitative - for instance “High”.
- Cause: A short description or a reference to one or more events that can cause the risk to occur.
- Response: This is a short description or reference to one or more contingency activities - i.e. activities that can be used to prevent or reduce the impact of the risk if it occurs.

Examples of a risk management tables are shown in the general example in the next section - see Table 17-6 (p.17-39) and Table 17-7 (p.17-40).

The risk connected to an event, defined as the product of the event’s probability and its consequences, can be assessed by using the following table:

Table 17-6: Risk assessment table

	Probability			
Consequence	Frequent 1.0 - 0.7	Probable 0.7 - 0.4	Improbable 0.1 - 0.4	Impossible 0.0
Catastrophic	High	High	Medium	Low
Critical	High	Medium	Medium	Low

Table 17-6: Risk assessment table

	Probability			
Consequence	Frequent 1.0 - 0.7	Probable 0.7 - 0.4	Improbable 0.1 - 0.4	Impossible 0.0
Marginal	Medium	Medium	Low	Low
Negligible	Medium	Low	Low	Low

For an activity connected to a project or to a change in our process, we can also assess the risk directly - that is: without assessing the probability and consequences first - by using the table below:

Table 17-7: Activity risk as a function of knowledge and experience

	Knowledge	
Experience	High	Low
High	This is the safe area. It contains activities where we have a large amount of experience and knowledge. (Low)	We have done this for a long time. The experience has, however, never been organized or researched. Thus, we are uncertain with respect to the effect in new areas. (Medium)
Low	This area contains our theoretical knowledge. Much of the information pertains to study reports, journals etc. The methods have never been used in practice at our site. (Medium)	This is the high risk area. (High)

How to use the Triangular Distribution

The following is a short description of how to use a triangular distribution for risk assessment. We take the diagram below as our starting point:

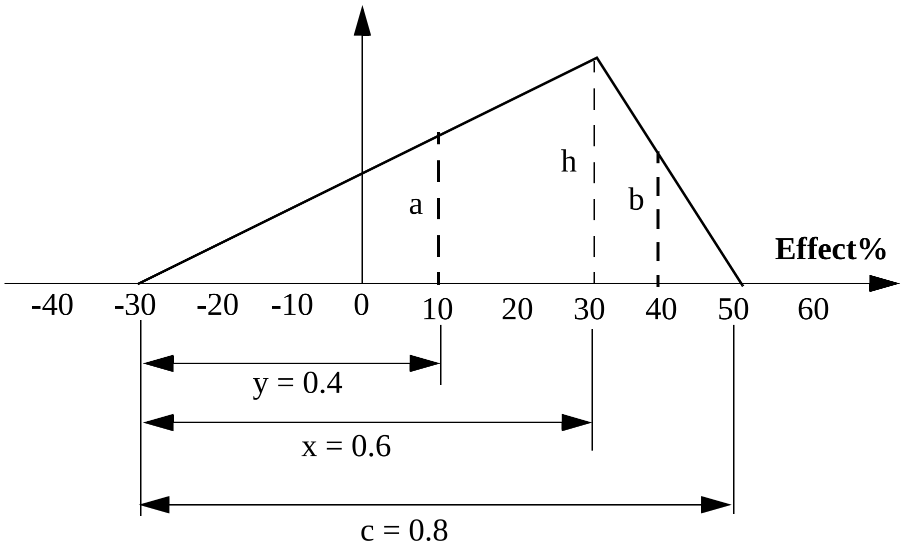

Figure 17-14: How to use triangular distribution

The maximum value - h - of the distribution is decided by the relation h * c / 2 = 1.0, which gives us that h = 2.5. Note that we here and in the following will use 0.8 instead of 80% and so on.

We will now use the diagram to obtain the answer to two questions:

- Q1 - What is the probability that the effect of the change will be less than 10%?
- Q2 - What is the probability that the effect of the change will be greater than 40%?

These two answers can be found as follows:

$$\frac{h}{x} = \frac{a}{y} \Rightarrow a = 1{,}67 \qquad P(\text{Effects} < 10\%) = \frac{a}{2} \times 0{,}4 = 0{,}33 \approx 0{,}3$$

$$\frac{h}{0{,}2} = \frac{b}{0{,}1} \Rightarrow b = 1{,}25 \qquad P(\text{Effect} > 40\%) = \frac{b}{2} \times 0{,}1 = 0{,}06 \approx 0{,}1$$

The triangular distribution is just an approximation. The “real” distribution will for instance be a Beta distribution. Since our approximation is rather coarse, we should never build an argument of preferences on small differences. To avoid this, we should never use more than one digit after the decimal point.

Change Cost Analysis

When we have run our experiments - in our case projects - and analyzed our data, we will have to decide on the real improvement steps. In order to have a basis for our decision, we need to perform some kind of cost benefit analysis. We will have a quick look at two methods that are commonly used, namely Leverage (p.17-42) and Return on Investment (p.17-42). Both methods imply that it is possible to connect quantitative costs and benefits to each activity.

In no case can the change decision be built on one or more of these analysis alone. In addition, company policy, overriding political concerns and so on must be taken into account. These questions are, however, not within the scope of this methodology.

Leverage

The leverage for a change is defined as follows:

$$\text{Leverage} = \frac{\text{Profit} - \text{Investment}}{\text{Investment}}$$

The larger the leverage, the better the investment. In order to include the current or expected rate of interest, the future profit and later, necessary investments should be reduced to their net present value (NPV).

Return on Investment

A Return on Investment (ROI) analysis is based on reducing all future income and costs to the net present value and then compare them to the initial investments. If the investment is less than the NPV of the future incomes, the investment is profitable. The larger the ROI, the better is the investment. The diagram below shows a typical situation.

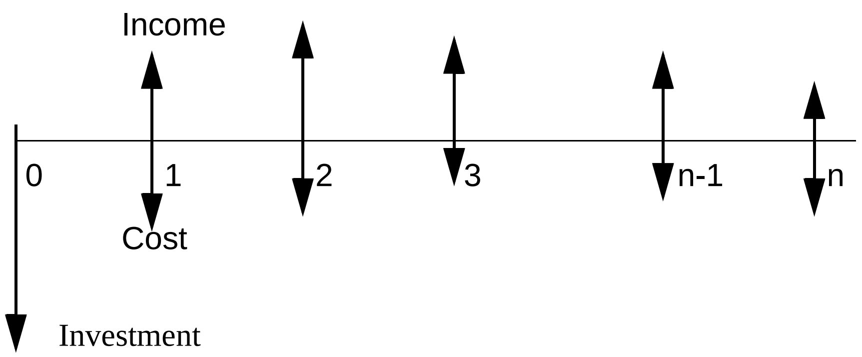

Figure 17-15: Income and costs per year for investment

In the diagram above, we see the investment in year 0, followed by a series of incomes and costs. The incomes are the positive results of the investments - for instance reduced development costs, while the costs are operational costs that are needed in order to defend the investments.

For the last year that we use the investment, we will also get a decommissioning cost. For equipment, this contains all costs concerned with getting rid of the equipment. It is, however, unclear if we have decommissioning costs for a method or for software tools.

We will introduce the following notations:

- I: Initial investments
- $C_{op,i}$: Operational costs in year number i
- C_{de}: Costs of decommissioning the investment
- REV_i: Revenue generated by the investment in year i
- r: Rate of interest. Many companies have their own, internal rate of interest for investments, while others use the current rate on bank accounts or on government bonds. As an example, the Norwegian government has as a rule that all investments must have a ROI larger than 1 with r = 7%. It is customary to cater for large risks by increasing the rate of interest.
- n: Expected life of investment. A typical value for equipment is five years. There is, at the present, no standard expected life time for a method or development process. However, for an area where we see a quick technological development and a large competitive pressure, n can be as short as two to three years.
- p: The probability of obtaining the stipulated revenue REV.

With this notation, we can write:

$$ROI = \frac{1}{I}\left(\sum_{i=1}^{n} \frac{REV_i \bullet p - C_{op,\,i}}{(1+r)^i} - \frac{C_{de}}{(1+r)^n}\right)$$

Note that in the general case, Leverage = ROI - 1.

It is usual to assume that operational costs are the same in each year, while the generated revenues will increase over time. The increase can be modelled as a learning curve or by assuming a simple, step-wise increase.

In some cases, we have that both operational costs and expected revenue are constants. For software and methodology we will assume that the cost of decommissioning is close to zero. In these cases, the expression can be considerably simplified, as shown below:

$$ROI = \frac{1}{I}(REV \bullet p - C_{op})\frac{1}{r}\left(1 - \frac{1}{(1+r)^n}\right)$$

The result of a ROI analyses depends strongly on the assumptions made concerning the change of the generated revenue over time. The following two diagrams show two starting points that can be useful in an analysis.

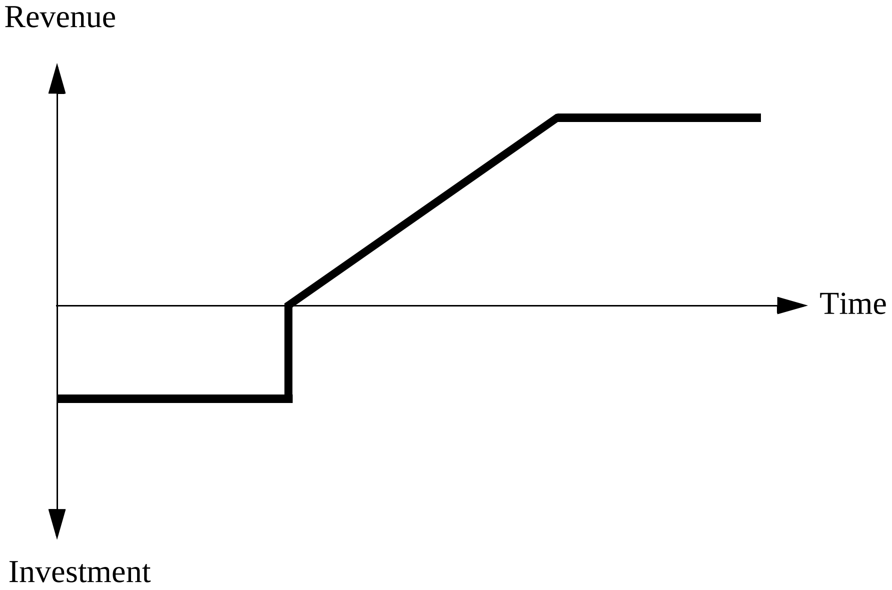

Figure 17-16: Stable return on investment after initial build-up

In the example we see a case where we - after the initial investment - gradually build up the organization to a point where the revenue reaches a stable level which can be sustained over a long time period.

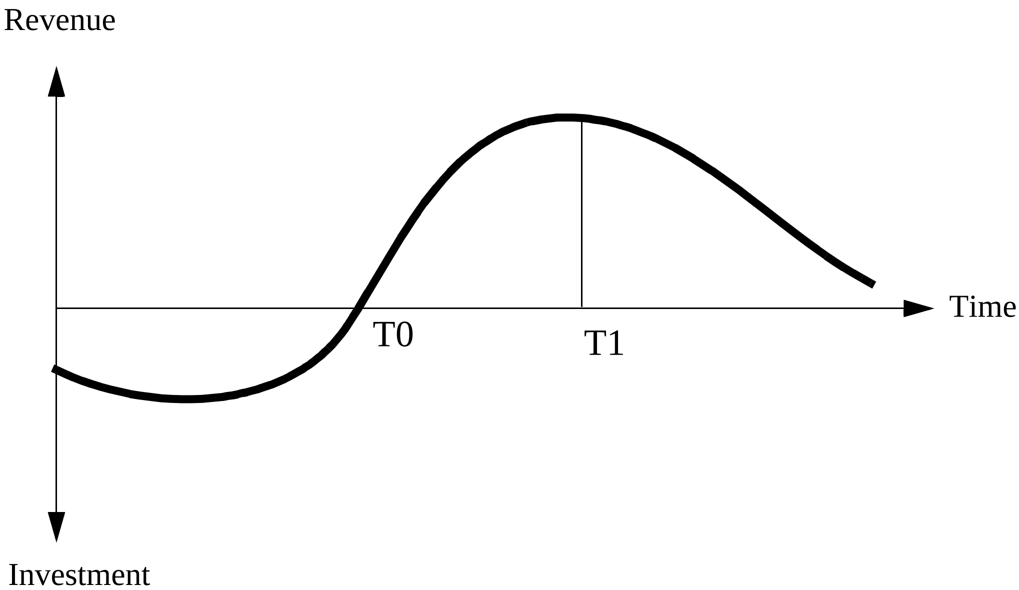

Figure 17-17: Decreasing return on investment

The example in the diagram above shows the typical development for an investment improvement in an area of strong competition. The point T0 is the time when we start to generate a revenue from our investments. T1 marks the time where our competitors start to catch up. From that time our revenue from that particular investment will steadily decrease as we are outsmarted by our competitors.

The Effect of introducing new Tools and Methods

This chapter will discuss how we can observe the effect of introducing new methods into an organization. If we shall be able to do a cost benefit analysis of the introduction of new tools and methods, it is important to be able to observe the effect and be able to decide if it has been beneficial.

The rest of this chapter discusses three related topics:

How to measure effect

The software industry, as all other industry, needs to have an economically based rational for the introduction of new tools and methods. Ideally, this should be based on a cost-benefit analysis. This requires, however, that we have complete knowledge on the effects of the new tools and methods before we start to use them. In most cases, however, only the costs can be assessed on before hand. The benefits are mostly left to be assessed by the gurus or promised by the salespersons, usually in rather exaggerated and undocumented terms such as “enormous improvement” or “large reductions in needed development resources”.

Since we, in the general case, cannot assess the benefits beforehand, we have to select the second best solution which is to state our expectations and decide how we will check if they have been met. The latter are our success criteria. In order to do this in an efficient and statistically sound manner we need to state the following:

- the expected effect. This is stated as a hypothesis
- our criteria for accepting or rejecting the hypothesis. This is stated as conditions on our observations
- what we shall observe and how we shall observe it. This is our software metrics definitions

These three points are discussed in full in How we can state our Hypothesis (p.17-45) and How we can Assess the Effect of new Tools and Methods (p.17-46).

How we can state our Hypothesis

A hypothesis has three components, namely The Conditions (p.17-46), The Hypothetical Relations (p.17-46) and The Acceptance Level (p.17-46). These three elements are described shortly below. An example on how to use Hypothesis (p.17-63) is given.

The Conditions

The conditions describe under what circumstances we expect our hypothesis to hold. For software tools and methods it is important to describe the projects where it will hold, our assumptions on project personnel and the customer's requirements.

By doing this, we can be reasonably sure that we only compare projects that are comparable. When we are using the "White Box" based Process Improvement (p.17-25) this is called characterizing the development process.

The Hypothetical Relations

The relations are the "thing" that we want to test or that we believe to be true. For the present context - introduction of new tools and methods - typical examples of relations could be:

- the introduction of the X-tool will reduce the error density in the finished product by at least 50%
- by using the Y method, the cost per delivered code line will be reduced by 40%

A relation must always be stated in such a way that it is possible to collect data that can be used to accept or reject that the relation holds.

The Acceptance Level

Many of the relations that we observe in the real world are not constant. Due to variations beyond our control, the observed data will have a natural variation. Instead of a single value, such data are presented by their mean value and an upper and lower limit. Such an interval is called a $(1 - \alpha)$ interval, since the probability that an observation will give a data value outside the interval is equal to α. This is also called the confidence interval.

Thus, when we say that we will use a 95% confidence interval of [1, 5] for X, this implies that there is a 5% probability that an observation of X will give a value less than 1 or larger than 5.

Alternatively we can state the mean value (μ) and its standard deviation (σ).

Our choice of confidence level will depend on the risk we are willing to take. This is discussed in depth in the section on Risk Assessment and Control (p.17-33).

How we can Assess the Effect of new Tools and Methods

This can be covered in two ways, namely

- after the introduction, to see if we have reaped the expected benefits
- before the introduction, so that we can do a initial cost/benefit analysis and set reasonable goals for improvement

We will cover both angles in the following section.

Initial Assessment

This assessment can be split into two major parts:

1. what will the tools or method do for you?
2. how will this affect your costs and benefits?

What will the Tool do?

It is usually straight forward to find out what the tool will do for you. We suggest that you try to relate it to one of the following categories:

1. help, in order to make a manual process step more efficient
2. partly automate a manual process step
3. replace a manual process step

In all cases the important number is the assessed effect of the new tool or method. This should be given as the portion of the work that will be removed - denoted S for savings. This is a measure of the efficiency of the tool or method. The next section will discuss this in more detail.

What are the Costs and Benefits?

Before you start on this assessment you need some key numbers concerning your current development process. Without them, the rest will be pure guesswork and probably a waste of time.

- How large a part of the company’s total budget is concerned with software development? The software budget is B_{sw}
- What is the distribution of resources over project phases or activities? The phase F_i uses a portion r_i of the resources.
- What is the distribution of inserted errors over project phases or activities? The phase F_i creates a portion n_i of all errors N that are found during development and operation
- What is your average error correction costs? The average total error correction cost is E_c

If you do not have this information already, getting it could be one of your first steps on the road to process understanding and improvement. Before you do it, however, be sure to know How to define Useful Metrics.

The following are the key questions that the organization needs to ask:

- How much resources are currently used on the activities that are affected by the new tools or methods?
- How much of the activity will be taken care of by the new tool or method? This results in the savings S.
- How will the new tool or method affect other processes or process steps? The work in phase F_i will increase by a portion g_i.

- What costs are connected to the new tool or method, such as cost of purchase, courses and maintenance?

For the case where the tool mainly affects one project phase, the assessment can now proceed as shown below.

Assume that the tool claims to reduce the work in phase F_a by a portion P and that the number of errors will be reduced by a factor q. Thus, the total savings in phase F_a are given as:

$$S = P \bullet B_{sw} \bullet r_a + q \bullet N \bullet n_a \bullet E_c$$

We see that the savings consists of two components, the saved work due to work reductions stemming from the tool and the reduced number of error corrections. We have here, for the time being, assumed that the errors are reduced in the same proportion as the work.

The total costs are given as follows:

$$\text{Extra costs} = \text{purchase} + \text{courses} + \text{maintenance} + B_{sw} \bullet \sum (r_i \bullet g_i)$$

As always, we need the savings to be larger that the costs. In addition, we must make sure that there are not any better ways to use the money that we now must spend to buy the tool and run our personnel through a new set of courses.

Post Festum Assessment

The post festum assessment of the introduction of new tools and methods is the same as the assessment of any other improvement-related change. Thus, the assessment can be done as described in steps 1 - 4 in “Black Box” based Process Improvement (p.17-25).

Examples

Here we present a number of examples to illustrate the methods defined in Process Improvement (p.17-2).

We give An example of an Improvement Scenario (p.17-49), An example of Setting Goals (p.17-51), An example on Risk Management (p.17-57) and An example on how to use Hypothesis (p.17-63).

An example of an Improvement Scenario

This section contains a description of a medium size software development company - called Company X (p.17-49) - and a scenario related to the improvement of the software development process: The "Black Box" Improvement Scenario (p.17-49).

Company X

Company X is a traditional software house that develops customized software products. Most of the systems that they develop are small, often needing less than one man-year of effort. The company has a staff of 20 dedicated developers who develop software using a traditional waterfall development process. Each project phase has a technical review at the end and the product is not allowed to leave one phase before the results are accepted by the reviewers.

The company has three categories of personnel, namely senior personnel who do the analysis and design, junior personnel who do the coding and sub-system testing and a separate test group who do the final acceptance testing.

The company has a resource registration system in place. In this way, they can find out how many person-days that are consumed in each project phase for each product. They keep logs from reviews and all tests, starting with subsystem tests. To keep an eye on quality, they register all error reported from the users. They want to keep the error rate observed by their customers below 1 error per 1000 lines of code delivered.

The "Black Box" Improvement Scenario

Improvement Instantiation

(see "Black Box" based Process Improvement (p.17-25))

The company has for a long time experienced a constant productivity measured as lines of delivered code per person-day. Three to five years ago, this was a quite respectable figure but the company has received signals from the marketplace that they are considered to be rather expensive. What still keeps them alive is that the product quality is quite good. However, their competitors are catching up on quality. In addition, more and more solutions are now offered by COTS technology, instead of expensive, custom-made software products.

Thus, as step 1 in the improvement process, the company decides that they need to improve their productivity while still being able to deliver the same, high quality that is their competitive edge.

By analyzing historic data, they find that their productivity figure is ca. 20 LOC / person-day. In order to improve their productivity they need to find out how the resource consumption is distributed over the project phases. This covers step 2 in the improvement process.

From the data from the three latest projects, they find the following phase-effort distribution:

- Analysis: 8%
- System Design: 4%
- Detailed Design: 17%
- Coding: 34%
- SubSystem Test: 17%
- Integration Test: 4%
- Acceptance Test: 16%

After having established the current status, the company has to decide on one or more improvement actions, according to step 3 in the improvement process.

After having discussed the collected data with their staff, the company's management and the developers agree to focus on the coding phase, since this is where the potential improvement are the highest. The company do not have data for each activity in this phase. Thus, it was decided to run a set of interviews with feed-back - a Delphi process - in order to find out how the resources were used in this phase. This process gave the following result:

- writing code: 30%
- compilation and fixing coding errors: 20%
- unit testing: 20%
- discussing and changing DD documents: 20%
- discussing and changing SD documents: 10%

A discussion of these data resulted in the following suggestions for improvements:

- compilation and fixing seems to take an unreasonable amount of resources. The main reason for this was a development approach that consisted of a continuous sequence of "cut and try" coding. A more disciplined approach should reduce the cost
- the coupling between the two design phases and the coding phase are not good enough. 30% of the resources spent in the coding phase are used to make the DD and SD documents fit for use. Participation of the coders in the design phases should increase the usability of these documents.
- there are by now ample evidence that unit testing is counter-productive and should be eliminated from the process

Based on these observations, the following changes were done to the process:

1. use a disciplined approach to coding. The code is written and the reviewed by two other programmers who check the code for logical and syntactical errors
2. coding personnel will participate in all reviews pertaining to design documents
3. all unit testing will be dropped

It was decided that these changes should be used for the next two projects. When the second project is finished, the data will be evaluated and they will make a final decision on whether they will keep the changes or go back to the way they developed software before.

As part of the improvement process - step 3 - the company also decided their success criteria. The changes will be accepted as beneficial under the following conditions:

1. the error rate observed by the customer must not exceed 1 error per KLOC
2. the productivity increase must be better than 20%

Follow-Up Activities

As required by the improvement process - step 4 - the company has to follow up on the effects of the changes. In order to do this, the company registered all the usual data for the next two projects. For the first project, they got the following results:

- productivity for project A: 27 LOC per person day
- error rate observed by the customer for project A: 0.8 errors per 1000 LOC

This looked promising, but the company also wanted to see the results form the next project in order to be more sure of their decision. For the second project, they got the following results:

- productivity for project B: 25 LOC per person day
- error rate observed by the customer for project B: 1 error per 1000 LOC

We see right away that both success criteria are met. The average production increase is 30%, while the error rate observed by the customer still is 1 error per KLOC or less.The company thus decides to keep the introduced changes.

An example of Setting Goals

This example is really two examples - one set of goals, but two ways to achieve it. The reason for this is that we will look at two different companies, called A and B. They are described in the table below, using the influence factors described in The Influence Factors (p.17-15). In order to better illustrate our points, we have chosen two rather extreme positions in the continuum of possible software companies. They are not, however, more extreme than that it should be easy to identify companies of both types in the European arena.

Owner’s Goals

The owners, represented by the management board, have decided on the following two goals for the company:

1. In order to improve the security of their investments, the company should increase its market share and, if at all possible, move into at least one new business area.
2. In order to improve the share value and the stock exchange value of the company, the company must increase the profit on their products.

Company Characterization

The factors used to describe the relevant company characteristics are taken from The Influence Factors (p.17-15).

Table 17-8: Company characterization

Influence factor	Company A	Company B
Type of company	Large company with many simultaneous development projects	Small company with only one development project at any time
Type of customers	Technologically conservative	Always looking for the “latest and greatest”
Type of market	Few, but large competitors of the same type as company A	Many, small - like B - and a few large - like A - competitors
Technological development	Large, slowly evolving body of capital-intensive technology	People-intensive technology that moves in skips and jumps
Innovation policy	Large technology survey department. Always looking for new technology to copy and adapt	Strong connection with universities and research institutes in order to pick up new ideas
Operational time-frame	New tools and methods every five - six years	No technology lasts longer than two years

Company Goals and Assessment

The next step in the improvement process is to obtain information of the relations described in The First Improvement Input (p.17-16). Interviews with customers, market departments and management gave the following results:

Table 17-9: Interview results

Factors	Company A	Company B
Customer impression	Good, state of the art technology. Service could be better	Excellent technology. Always in the forefront of development Always listening to customer requests and complaints
Expected market changes	Market will mostly stay the way it is. A slight, but steady decline is possible	Ever-changing market. Can not predict or influence market movements
Expected product changes	Will keep product line but replace technology as new technology becomes available from subcontractors	Will stay in same market segment and sell same type of product as before. Technology will change in an unpredictable way
Where will we be after this plan period?	Mostly stay the same. Will have to move into new areas in order to have a more secure basis for operation	Will try to stay in same market area as now. Will look for new ways to apply and sell the technology were we have a leading edge

Interviews with the company managers and marketing gave the next-level goals shown below. These goals are strongly related to the board's goals, but are closer to product, process and service. For company A, we got the following goals:

1. Increase product quality with special focus on the number of errors found in the product after installation
2. Lower development costs and thus increased productivity.
3. Shorter time to market in order to grab larger market shares.
4. Lower maintenance costs and thus lower life cycle costs, both for the company and its customers.

For company B, we got the following list of goals:

1. Increase product quality with special focus on the number of errors found in the product after installation.
2. Increased flexibility towards special requirements from single customers.
3. Increased service level and thus the company's standing in the market.

4. Lower development costs and thus increased productivity.
5. Shorter time to market in order to grab larger market shares.
6. Lower maintenance costs and thus lower life cycle costs, both for the company and its customers

Customer View and Management Goals

Two examples of the results of interviews with customers and management of different companies are presented: Company A - the large, stable company (p.17-54) and Company B - the Small or Medium sized Enterprise (SME) (p.17-54).

Company A - the large, stable company

This company has few but large customers. In order to get an idea of their customer relations we interviewed

- A representative from one of their customers
- People in the company’s marketing department who is responsible for the handling of customer complaints

The results from this data collection can be summed up as follows:

- The company is considered to be reliable and solid. It is preferred as supplier because they can deliver turn-key systems, not just software or software plus main processor(s).
- The customers are - by and large - satisfied with the quality of the company’s products but feel that they, at least in some cases, could have gotten at least the software cheaper somewhere else.
- The company is sometimes considered to be arrogant and stubborn in its customer relations. This becomes especially outspoken when the customer wants to change one or more of the product requirements.
- The customers foresee no big changes in their needs or requirements in the next three to four years.

The interviews with top and middle management focused on market and product stability plus the company’s current improvement time-frame. These interviews confirmed what we had summarized in table 4 above.

Company B - the Small or Medium sized Enterprise (SME)

Input to our customer view assessment was obtained from three sources:

- Data collection performed by the company’s marketing department.
- Interviews with four companies that marketed and sold the company’s products.
- Interview with two, randomly selected customers - one university department and one building inspector.

The results from this data collection can be summed up as follows:

- The company's products have a high quality. They are considered to be much more up-to-date on the latest development in their application area than their larger competitors.
- The company is considered to be extremely open to their customers' request for special modifications.

The interviews with the company's manager and the person who was responsible for marketing confirmed our impressions as summarized in table 4 above. Both persons put special stress on the need to be flexible and quick when it comes to handling customer change requests.

Towards Technical Goals - Second Step

In order to decide which goals to use, both companies decided to use the influence matrix method. We have used two tables, one for the goals of Company A - Large, Stable Company (p.17-55) and one for the goals of Company B - Small or Medium-sized Enterprise (p.17-56). In both cases, we have used a rather primitive assessment method by just using a score of 2 for "++", 1 for "+", 0 for "0", -2 for "--" and -1 for "-". Other, more sophisticated scoring schemes can also be used.

Company A - Large, Stable Company

The decision matrix for company A turned out as follows:

Table 17-10: Goals for Company A

Company goals	Reach goals	Personnel acceptance	Necessary investments	Lifetime of investment	Operational costs	Time to get results	Inherent risks	Summary
Increase product quality	0	0	--	-	+	+	+	0
Lower development costs	++	+	-	++	++	-	0	5
Shorter time to market	++	+	--	+	-	-	--	-2
Lower maintenance costs	+	+	0	++	+	-	+	5

Based on this assessment, company A decided to use two technical goals, namely:

1. Lower development costs.
2. Lower maintenance costs.

These goals are then approved by the board. We can now proceed to identify GQM-goals based on these two technical goals. We conducted a series of interviews with developers and middle management. During these interviews, the following ideas surfaced. In order to achieve the technical goals, it was important to find out:

1. Where do we spend our resources during development and maintenance - and why?
2. Where do we spend calendar time during development and maintenance - and why?
3. What will be the effect of introducing the new method X and a support tool for this method?
4. How large - in LOC - is a typical maintenance change?
5. How much resources and calendar time do we need for a typical maintenance action?

Company B - Small or Medium-sized Enterprise

The decision matrix for company B turned out as follows:

Table 17-11: Goals for Company B

Company goals	Reach goals	Personnel acceptance	Necessary investments	Lifetime of investment	Operational costs	Time to get results	Inherent risks	Summary
Increase product quality	++	+	0	-	0	-	-	0
Lower development costs	++	--	-	+	+	-	--	-2
Shorter time to market	0	++	--	++	--	--	--	-4
Lower maintenance costs	++	++	-	+	0	0	+	5
Increased service level	+	0	-	-	--	++	++	1
Increased flexibility	++	++	--	++	+	++	++	9

Based on this assessment, company B decide to use two technical goals, namely:

1. Increased flexibility.
2. Lower maintenance costs.

These goals are then approved by the board and the company can now proceed to identify GQM-goals based on these two technical goals.

Based on the goals as stated by the board of directors, we conducted a series of interviews with developers and middle management. During these interviews, the following ideas surfaced. In order to achieve the goals of the management, it was important to find out:

1. Where do we spend our resources during maintenance - and why?
2. Where do we spend calendar time maintenance - and why?
3. What will be the effect on introducing the new method X and a support tool for this method?

4. How large - in lines of code - is a typical maintenance change?
5. What are the customers' priorities?
6. How much resources and calendar time do we need for a typical maintenance action?

An example on Risk Management

The Ishikawa diagram below is an example of what might happen if we analyzed the goal "Higher productivity".

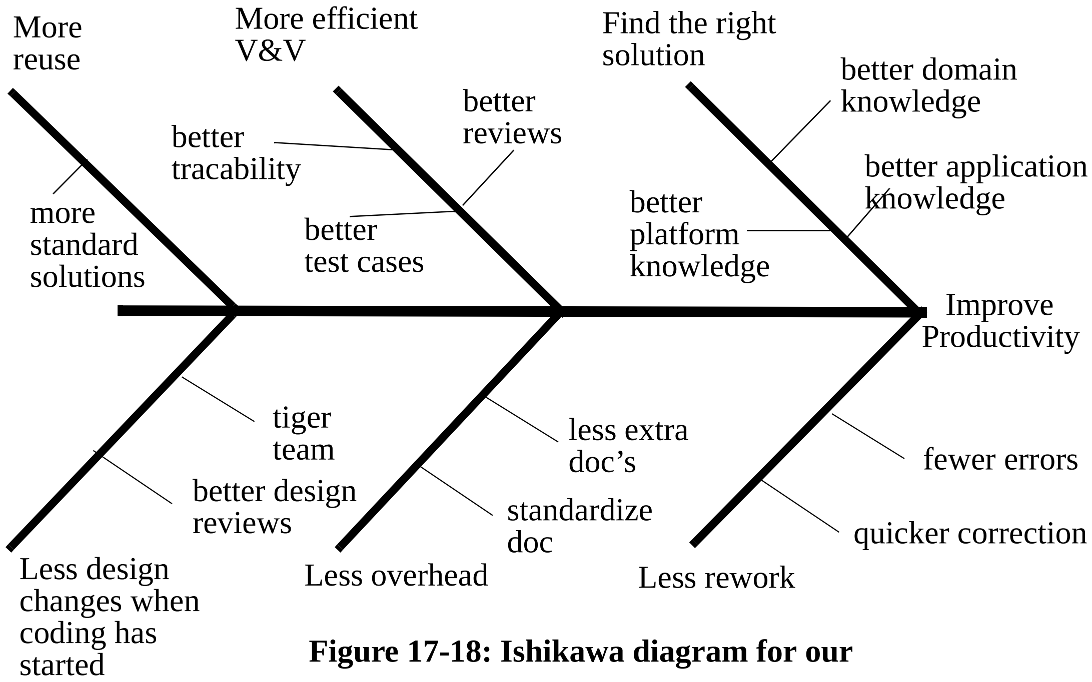

Figure 17-18: Ishikawa diagram for our

In the selection of increased reuse as an improvement activity, the company required that the reuse should be profitable already in the first project. This implies that the company can not first use several years to build up a library of reusable components but will have to rely on available components, at least in the start-up phase.

Since it would be impractical to try to use all these improvement changes, we next perform a pair-wise comparison in order to rank the six alternatives. See the section on "Selecting technical Goals". The pair-wise comparison process gives the result shown in the table below:

Table 17-12: Pair-wise comparison

Activity	Comparison count						Sum
More reuse	IIII						4
More efficient V&V		II					2
Find the right solution	I	I	III				5

Table 17-12: Pair-wise comparison

Activity	Comparison count						Sum
Less design changes				I			1
Less overhead					I		1
Less rework		I		I			2
Check sums	5	4	3	2	1	-	15

Based on this decision process, the company decided to try and change the process so that it helped the developers in:

- Finding the right solution.
  This implies that we need to obtain better domain knowledge, better platform knowledge and better application knowledge.
- More reuse.
  Increasing the use of already finished components in the development of new systems. This implies that we should seek to increase our use of standard solutions during development.

In order to decide on the risks and benefits of the selected actions, the company performed a The Wideband Delphi Method (p.17-31) analysis. They used four of their own developers plus two outside experts - one from the computer science department of a technical university and one from a consultant company.

The Delphi panel’s assessment of the effect of each activity on the goal “Improve Productivity” can be regarded as follows:

- More reuse:

Strongly divided panel. “Anything” can happen, from a 40% increase to a 10% decrease in productivity.

- Find the right solution:

Both better domain knowledge and better application knowledge were both assessed to have a maximum effect of 30% improvement, but could also have none at all.

There are a general agreement that better platform knowledge would have a small effect - if any - on productivity.

This resulted in the following diagrams:

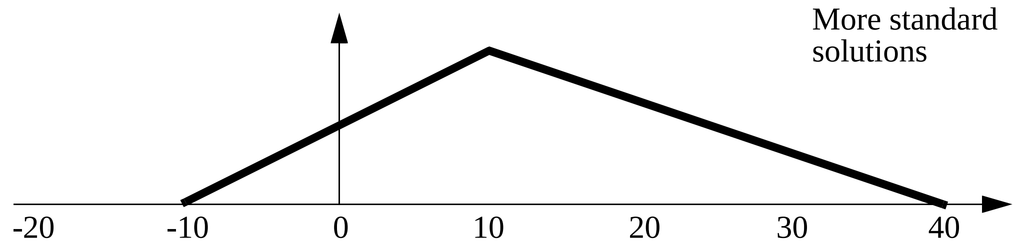

Figure 17-19: Wideband Delphi results for "More reuse"

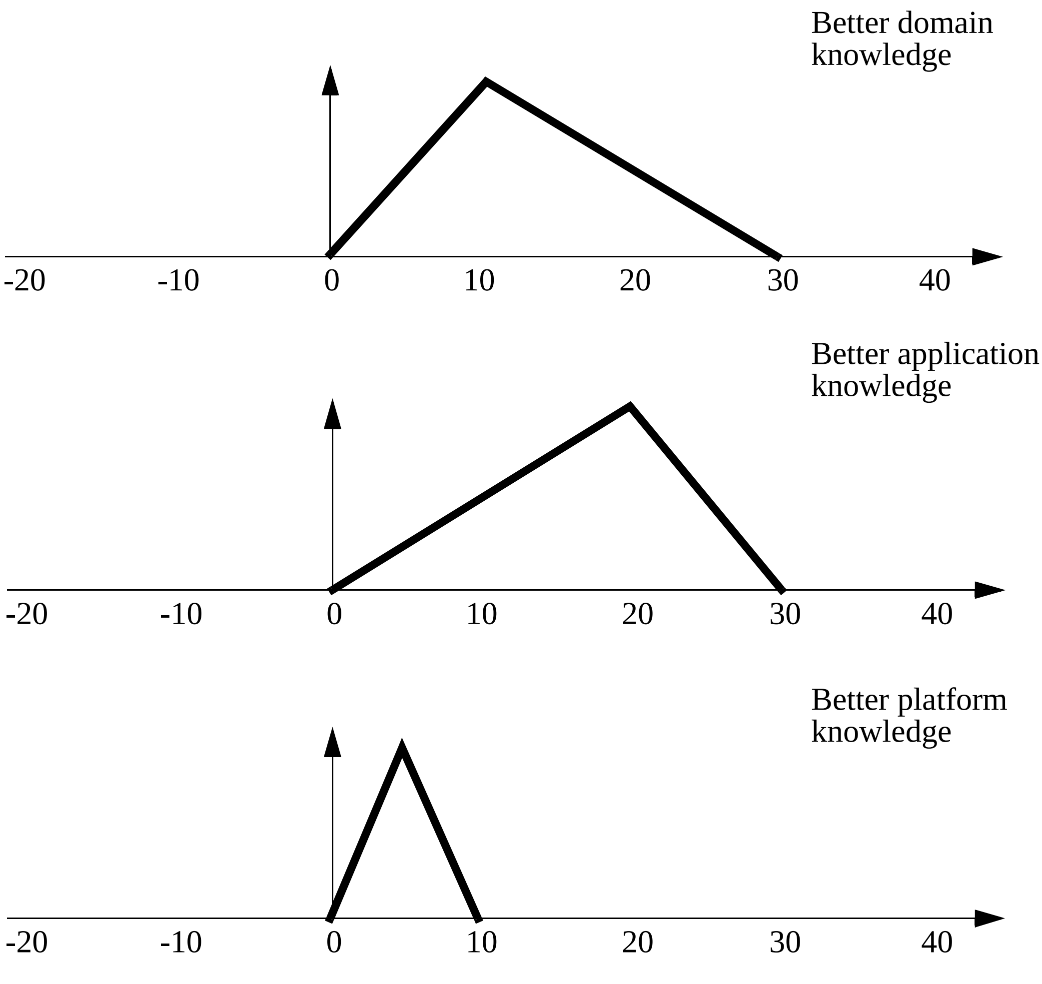

Figure 17-20: Wideband Delphi results for "Find the right

Based on this results, the company decided to do the following:

1. Drop activities related to "Better platform knowledge".
2. Start a course to increase application knowledge for all developers in the company.
3. Start a process to find out what promotes and hinders large impacts from "Better domain knowledge" and "More standard solutions".

4. Identify risks that can prevent us from reaping the benefits from better application knowledge (Table 17-13 (p.17-61)) and more standardized solutions.

This work resulted in identification of the following promotion / hindrance factors and risks:

More standard solutions

- Hindrance
  No component libraries available on acceptable terms.
  The reuse of imported components cause design problems.
  Customer resistance to the use of standard solutions.
  Rapid technology change makes the libraries outdated.
  Rapid changes in customer needs make the libraries outdated.
- Promotion
  Better application understanding
  Course in reuse of software components
  More “What” and less “How” during design

If the activities that are identified above are performed and the hindrance factors (risks) are controlled, the group of experts - in a new The Wideband Delphi Method (p.17-31) analysis - agree on the following result profile:

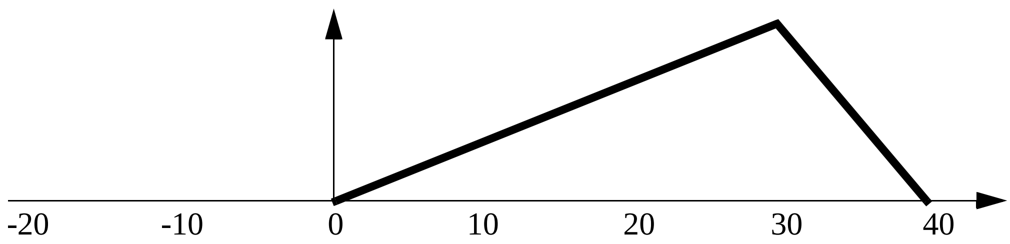

Figure 17-21: Wideband Delphi results for “More reuse” - second round

As a result of this, the company managers agree that the risk is acceptable and can be managed. They also agree to start with the activities necessary to achieve software reuse through the use of a commercially available library of standard software components.

The following risks are identified:

- Loss of component library supplier.
- Large, unexpected changes in the methods and techniques used in the application area.
- Large component price increases.
- Components do not live up the promised quality - reliability, maintainability etc.

Based on this, the following risk management table is defined:

Table 17-13: Risk Management table for software reuse

Activity	Identified risk	Estimates		Cause	Response
		Probability	Consequences		
More standard solutions	Loss of supplier	Low	High	Supplier closes down or goes bankrupt	Keep updated list of alternative suppliers
	Changes in application area	Low	Medium	Unexpected technological development	Subscribe to technology development surveys from university
	Price increase	Medium	Medium	Larger market or general cost increase	Accept
	Too low quality	Medium	High	Supplier develops substandard components	Run audits before signing contract with supplier Run test on selected components in received libraries

Application knowledge

- Risks:
  Bad application knowledge course
  Low motivation among developers
  Time pressure makes the developers unwilling to or not interested in course participation
  Large and quick changes in the application area.

Based on these identified risks, the following risk management table was defined:

Table 17-14: Risk management table for Application knowledge

Activity	Identi-fied risk	Estimates		Cause	Response
		Probabil-ity	Conse-quences		
Course to improve applica-tion knowl-edge	Bad course	Medium	High	Bad lectur-ers or bad contents	Check course. Ask for references
	Low motiva-tion	Low	High	Developers do not understand the need	Company motivation drive. Success stories
	High time pressure	High	Medium	Bad plan-ning Unexpected project problems	Better planning Plan with a margin
	Changes in the applica-tion area	Low	High	Technologi-cal break-through	None at the present

The two most important risk are “Bad course” and “High time pressure”. The “Bad course” risk is easy to deal with, as is indicated in the “Response” column. For “High time pressure”, the situation is quite different. The response - to plan with a margin - is quite costly and will put the company in an unfavorable position - at least in the short term.

Domain knowledge

- Hindrance
  Domain is unstable or is undergoing rapid changes
  Domain knowledge is difficult to systemize or understand
  Low developer interests in this domain
- Promotion
  Create domain interest through courses and workshops

Management decides to initiate courses and workshops to create interest in domain knowledge. In addition, they will arrange for visits to important customer sites and create a forum for the exchange of ideas between customers, marketing and developers.

An example on how to use Hypothesis

In order to make the ideas described in How we can state our Hypothesis (p.17-45) a little more clear, we will here work through a small example.

1. The hypothesis is stated as follows:

   For a project with customer requirements as described in document A, manned with personnel that meet the requirements described in document B and the development platform as described in document C, the productivity at the present has a mean value of 19 and a standard deviation of 1.3.

   Introduction of tool X and method Y will increase the productivity by at least 30%, i.e., it will have a mean value of 24.7 and the same standard deviation as before.
2. In order to test the hypothesis, we must run projects which fulfil the conditions stated in the condition part of the hypothesis. We run three small projects and get the following productivity figures: 18, 25, 28. This gives us a mean productivity of 23.7 with a standard deviation of 4.2.

   We will accept the hypothesis of improvement if the new average productivity is greater than the old average plus two times the standard deviation, which is the 95% confidence interval. This gives us the value of 21.6. Since the new average productivity is 23.7 we will accept that the new tool and method have improved our productivity.

List of figures

List of definitions

Standard deviation

The standard deviation, usually denoted by the greek letter σ, is a measure for the variation of a data population. A set of observations is a sample from this population. Together with the mean value, usually denoted by the greek letter μ, these two parameters describe many important characteristics of a statistical distribution.

If the data are symmetrically distributed around the mean value, then there is a 95% probability of finding a data point inside the range μ +/- 2σ and a 99% probability of finding the data point inside the range μ +/- 3σ.

As a result of this, if we observe a value that is greater than $\mu + 2\sigma$, there is a 5% probability that this data point belongs to the same population as the others - or in other words - there is a 95% probability that this data point comes from another population.

Counter implementation

[Keen 81]: *How to oppose a decided change without showing your face:*

1. *Lay low*
2. *Rely on inertia*
3. *Keep things complex, hard to coordinate, and vaguely defined*
4. *Minimize the legitimacy and influence of the change agent*
5. *Exploit the lack of knowledge of the change agent*

Must be met by counter counter implementation.

Counter counter implementation

[Keen 81]: *How to cope with* counter implementation:

1. Make sure you have a contract for change
2. Seek out resistance; treat it as a signal to be responded to
3. Rely on face-to-face contact
4. Become an insider; work hard to build personal credibility
5. Co-opt users early

The Jante Law

The Jante Law according to Aksel Sandemose:

1. Du skal ikke tro at du **er** noe. Thou shalt not presume that thou **art** anyone [of notice].
2. Du skal ikke tro at du er like så meget som **oss**. Thou shalt not presume that thou art as good as **us**.
3. Du skal ikke tro at du er klokere en **oss**. Thou shalt not presume that thou art any wiser than **us**.
4. Du skal ikke innbille deg du er bedre enn **oss**. Thou shalt never indulge in the conceit of imagining that thou art better than **us**.
5. Du skal ikke tro du vet mere enn **oss**. Thou shalt not presume that thou art more knowledgeable than **us**.
6. Du skal ikke tro du er mere enn **oss**. Thou shalt presume that thou art more than **us** [in any way]
7. Du skal ikke tro at **du** duger til noe. Thou shalt not presume that **thou** amount to anything.
8. Du skal ikke le av **oss**. Thou art not entitled to laugh at **us**.
9. Du skal ikke tro at noen bryr seg om **deg**. Thou shalt not presume that anyone cares about **you**.
10. Du skal ikke tro at du kan lære **oss** noe. Thou shalt not suppose that thou can teach **us** anything.

The Jante Law (Janteloven) is from the novel “En flygtning krysser sitt spor” (‘A refugee crosses his tracks’) by the Norwegian/Danish author Aksel Sandemose. The book takes place in an imaginary Danish small town called Jante, based on Sandemose’s hometown Nykøbing Mors. The book is about the ugly sides of Scandinavian smalltown mentality, and the term has come to mean the unspoken rules and jealousy of such communities in general.