Business Information Technology Series

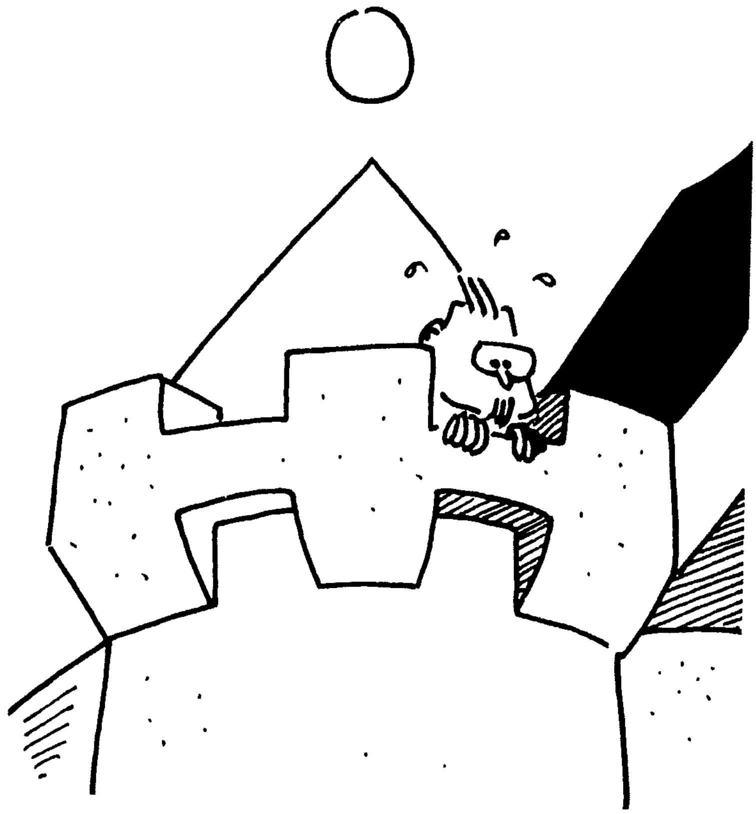
PROFESSIONAL SYSTEMS DEVELOPMENT

EXPERIENCE, IDEAS AND ACTION

NIELS ERIK ANDERSEN
FINN KENSING
JETTE LUNDIN
LARS MATHIASSEN
ANDREAS MUNK-MADSEN
MONIKA RASBECH
PÅL SØRGAARD

2 Situation and action

Death to standard methods



Many system developers have the idealistic notion that projects can go according to standards and given guidelines. Section 2.1 illustrates why this is not always the case in real life projects. In some cases the system developers can rely on routine, but in other cases they need to experiment or make a mental effort to be able to understand and handle the situation. Section 2.2 describes three types of situations, each of which is characterized by different degrees of uncertainty, and discusses what is required to act and intervene in a project. Section 2.3 illustrates why system developers must actively establish and regulate a project and its relation to the surrounding environment: which aspects in the environment are important? What makes a project an organizational unit?

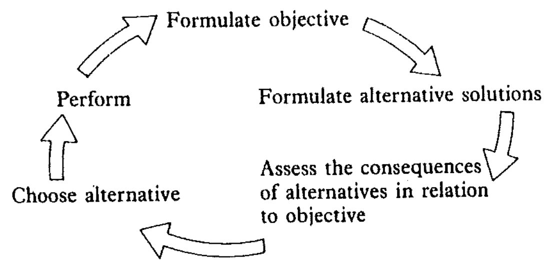
The relationship between the system developers on the one hand and the users and user organization on the other hand is important. Section 2.4 discusses the practical possibilities for establishing cooperation between the project and the users. In some cases users and their unions may demand active involvement, but this may vary considerably.

Section 2.5 argues that professional system developers must have the possibility, and the willingness to increase and change their repertoire of working practices. Only in this way is it possible to act sensibly in different types of situations. At the same time system developers have to develop their skills in accordance with technical and organizational changes.

2.1 Ideal and reality

To believe in rationality is both naive and irrational

Many of the methods employed in systems development are based on a simple rational ideal (see Figure 2.1). They assume that the involved parties in a given project share the same clearly defined objectives, that resources are available and plentiful, and that it is possible to identify the different design options and their consequences on the basis of analyses. Many prefer to see themselves as rational system developers. They also prefer to believe that they have a very clear notion of what it is they wish to obtain before they start the project. And they prefer to have good reasons for choosing a certain solution or procedure over another.



Assumptions:

- The project is based on clear and relatively stable objectives.
- The system developers know all about alternative system and project designs.
- The system developers can measure the consequences of the various designs in relation to the objectives.
- The system developers are willing to, and have the necessary resources to, maximize the fulfilment of the objectives.

Figure 2.1 The rational ideal and the assumptions on which it is based.

However, projects usually take quite a different course. They start off without a clearly defined objective. Design decisions are taken before the alternatives have been investigated, and before the consequences have been analysed. What is the reason for this discrepancy? Is it because system developers have not yet become competent enough? Would they be able to make reality accord with the rational ideal if they worked in a more disciplined manner, and followed the common standards more strictly?

There are many reasons why system development projects cannot follow a rational ideal:

- In most cases the user organization does not know exactly what it wants, or the users are unable to describe what they know.
- Even if system developers knew the requirements, there are many other facts they must know before they can develop a system. Many important details only become clear later in the process, and design decisions may have to be revised.
- Experience has shown that even if system developers knew the relevant facts before they started, the complexity is often difficult to handle. During the project the system developers must separate the important from the unimportant. This requires experiments and mistakes cannot be avoided.

Take your starting point in the situations

- Even if system developers could handle the relevant details, requirements still change in the course of almost all projects, and external relations even create conflicting requirements. Situations like that may require that new design decisions be made.
- Human error can only be avoided if we avoid human beings.
- The projects will usually be subjected to ideas from previous projects or from methods. Such ideas are not necessarily rationally linked to the requirements of the new system.
- Economy may force system developers to apply software or hardware which was intended for other purposes, or which is not yet finished and tested. The software and hardware is consequently considered adequate in light of the money saved, but does not necessarily meet the requirements of the project.

These are some of the reasons why system development projects cannot take a rational and unproblematic course. It would be professionally naive to think that system development projects can be made to follow a rational ideal. However, the above-mentioned fundamental characteristics of system development projects present an enormous professional challenge. If that challenge is not accepted, projects will inevitably slip off course because of random decisions or lack of intervention.

2.2 Situations and uncertainty

Take your starting point in the situations

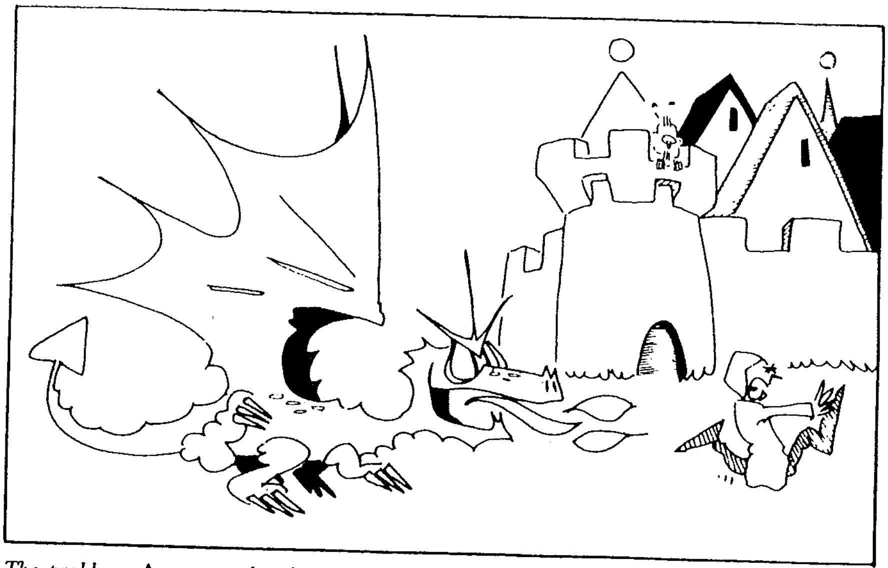
During the course of a project different types of situations will arise, each of which makes special demands on the system developers. We distinguish between three types of situations which we call routine, problem solving, and problem setting situations, respectively (see Table 2.1). The three types of situations correspond to an increasing degree of uncertainty in the situation. It is easiest to predict the course of, and estimate the required effort in a routine situation; and it is hardest to do so in a problem setting situation.

Situation \ Attributes in relation to	Assignment or problem	Working practice	Uncertainty
Routine	Known	Known	Small
Problem solving	Known	Unknown	Intermediate
Problem setting	Unknown	Unknown	Large

Table 2.1 Characteristics of three different types of situation which arise in system development projects.

Routine

The problem and the task are known in routine situations, and so also are the correct working practices to apply to handle the task. The situation is routine when a module which can recognize input of a given format is to be programmed. The task is known, and it is known that a solution can be structured with the help of state-transition tables. The situation is also routine when it is announced at a project meeting that a sub-activity is finished ahead of schedule. The management task is known, and it is known that the situation requires adjustment of the original project plans.



The problem. A system development project takes its starting point in a more or less well-defined problem in the user organization.

Routine situations are easy to handle because they are known, and so is the method of handling them. However, routine situations may still require an extensive and qualified effort.

Problem solving

System development projects require active and competent project cultivation

The task is known in problem-solving situations, but there is uncertainty or inadequate knowledge about which working practices to apply. A problem-solving situation arises when it is known that there are errors in a program. The task is clear, but it is not known exactly how to tackle the situation to find the error. A problem-solving situation also arises when one of the project group members is suddenly transferred to another project or leaves the company. The situation is clear, but it may be difficult to find an expedient way to intervene.

Problem-solving situations are open situations. They often require adoption of an experimental strategy. Problem-solving situations require mastery of different working practices to try to apply to the problem.

Problem setting

In problem-setting situations the whole situation is unclear to us. We know that we have to intervene in one way or another. But we do not know the problem or the task — and consequently we do not know what procedure to employ. If a project finds that the user organization rejects one design proposal after the other, the project group will know that something is wrong. But is it the design? Or is it because the user organization's original requirements do not reflect the current reality?

Another example is a project where the staffing of sub-tasks results in disagreements or bad feelings. This is also a problem-setting situation. Is the problem the way the assignments were allocated? Does the situation reflect personal conflicts in the group?

In problem-setting situations a number of symptoms become evident. From there the situation must be analysed to identify the problem or the task. There is no certainty, however, that it will be possible to define which problem or problems covers the situation. And if it is possible, there is no certainty that everyone involved will agree with the formulation of the problem. Various conflicting interpretations of the situation might emerge. There is a temptation to do nothing or try to cover up the situation, even though it is known or suspected that something is wrong.

System development projects contain, as described above, different types of situations, and this means that the system developer is facing two challenges.

System development projects require active and competent project cultivation

The first challenge is to be able to handle the various situations arising in a project. In problem-setting situations one may try to define the problem or the task, and in this way create a routine or problem-solving situation. In problem-solving situations one may make use of experiments to work out the appropriate working practices, thus creating a routine situation. System development projects cannot be formed merely on the basis of standards and given guidelines. They require active and competent project cultivation throughout the course of the project. System developers must be able to assess the situations in which they find themselves. They must be able to face problems and mistakes, and be open-minded with respect to the status of their own work and the project as a whole.

System developers must be prepared to break with traditions

Another, more fundamental, challenge is to develop and change the repertoire of working practices. The degree of uncertainty in the project situation is determined by the experience and qualifications of the system developers participating in the project. For instance a situation might be a problem-setting situation to an inexperienced person, but a problem-solving (or even a routine) situation to a more experienced person. System developers must master a repertoire of working practices and must be willing and able to introduce new working practices in the project. They must not get stuck in traditions. New technical and organizational conditions make new demands, and system developers must therefore change and develop their repertoire of working practices.

Exercise 2.1

Discuss project situations

Choose a number of typical situations from your last project. Some of the situations should be concerned with actual project performance, others with project management.

Characterize each situation as either a routine, problem-solving, or problem-setting situation.

Discuss the working practices applied, and whether seeing the situations as either routine, problem-solving, or problem-setting situations helps you to consider more expedient working practices.

There is a constant interplay between situations, working practices, and conditions

2.3 The project and its environment

It is necessary to know many aspects of the project and its environment to be able to understand and handle the different types of situations. Firstly the relations between the situations, the working practices and the conditions of the project are discussed. Secondly the mutual

3.2 Performance and management

We need method-independent descriptions of systems development

What components is systems development made up of? What type of work is involved? All systems development methods contain an answer to that, but the answers are always linked directly to the guidelines in the books. A method-independent understanding of systems development is needed. How would it otherwise be possible to understand and explain the actual situations in a project? Or how would systematic comparison be possible between different methods?

There are various possibilities for characterizing the main components of systems development:

- The work itself can be described, or the services and products resulting from the work can be described.
- What actually happens can be described, or what is intended to happen.
- The descriptions can be made more or less abstract.

Here the work itself is chosen as the starting point. Figure 3.2 illustrates some basic concepts for this purpose. Function, task, activity, and process correspond to more or less abstract ways of looking at work, and they describe both what actually happens, and what is intended to happen.

Process and activity

A process is the most concrete description of what actually happens. Processes take place in time and space, and they can be split up into sub-processes. A systems development project contains a process limited in time, resulting in specific products and services.

An activity denotes selected parts of a process which are seen as belonging together, for instance interviewing user groups, programming a module, or revising detailed plans. Activities denote what is actually done in a systems development project. However, a description of an activity is more abstract than a description of the process itself. In an activity specific aspects of the elements in the process have been selected.

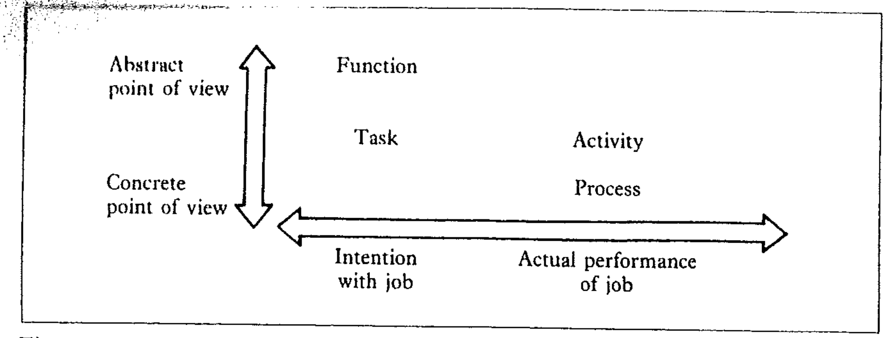


Figure 3.2 Some basic concepts for describing systems development.

The name of the activity typically indicates those elements in the process which is focused on. A project group meeting may be seen as a process where various events take place simultaneously; presenting viewpoints, discussing, passing on information, drinking coffee, etc. The meeting may be viewed in a more abstract way, namely as an activity: a review. In that case the interest is in the specific aspects of the process linked to this activity — others are ignored.

Function and task

A function is the most abstract way of looking at systems development. As opposed to processes and activities which express what is actually happening, a function denotes the intended result of one or more processes regardless of how the processes actually take place. Analysis, design, and planning are examples of functions which in practice correspond to specific activities. Functions, like activities, are delimited by content and thus activities typically contribute to the execution of the corresponding function. A function is performed through one or more activities. One may talk about the function of planning or about the activity of planning. The relationship between functions and processes is more vague because processes are not delimited by content, but rather denote the totality of what actually takes place in the course of a project. Thus a process typically contributes to several functions simultaneously. A process might, for instance, contribute to both analysis and design activities.

A task is a more concrete description of an intention. A function can typically be concretized into a number of tasks directly indicating the desired result. The planning function may, for instance, be concretized in: working out the next baseline, working out a detailed plan for the next phase, revising an overall project plan, etc. The tasks may correspond directly to the equivalent activities.

Performance and management

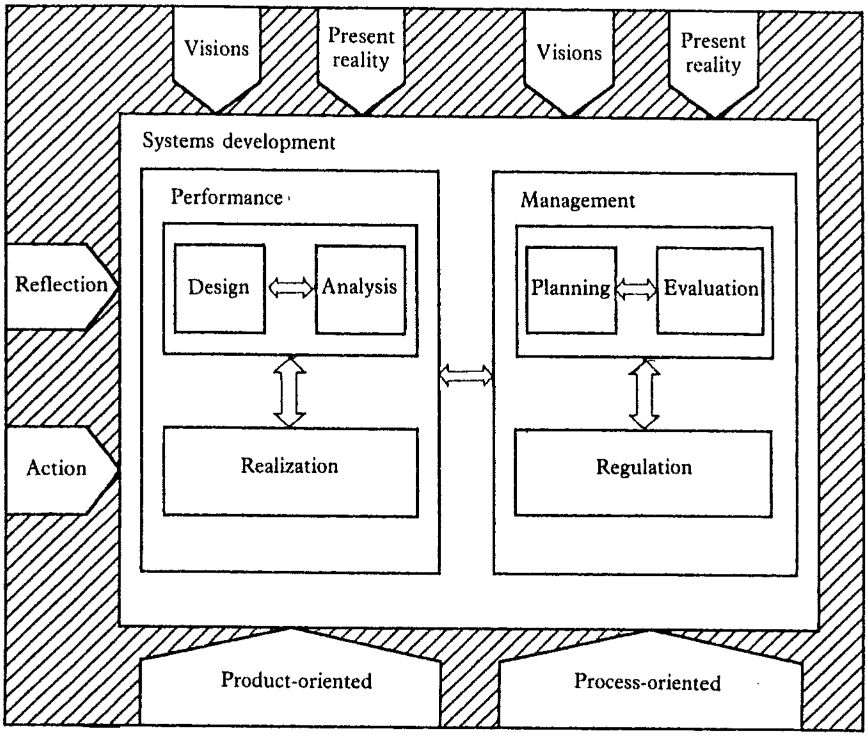


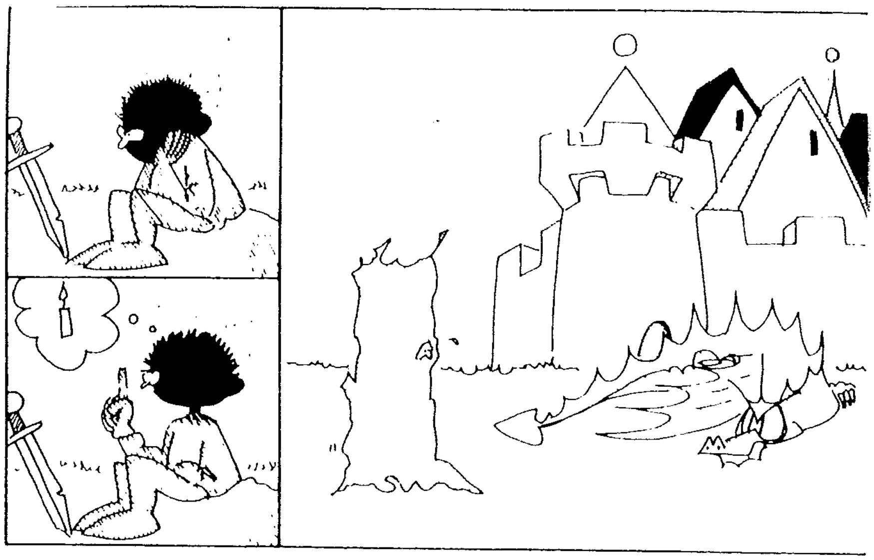
Figure 3.3 The main components of systems development. On the functional level the arrows indicate the relation between funtions. On the activity level the arrows indicate the interaction between activities.

It is time to leave the basic concepts and move on to what all this is really about: the main components of systems development. The description given in Figure 3.3 is based on three fundamental distinctions:

- Product orientation versus process orientation.
- Reflection versus action.
- Present realities versus visions.

Performance and management

System developers basically perform two types of creative activities. They create a computer-based system, i.e. a computer system and changes in the user organization. And they create a project resulting in the planned computer-based system. Systems development thus consists of two elements: a product-oriented element which we call


Reflection and action. During the work process the participants develop new knowledge about the process: its nature, possible procedures and possible tools.

performance, and a process-oriented element which we call management.

Analysis leads to an understanding of present realities

Reflection and action

Apart from this distinction a distinction is made between reflective activities and actions. Systems development requires an organizational, as well as a technical understanding of possibilities and conditions. However, the basic intention is to create and to change. The interaction between reflection and action takes place all through the course of the project, for instance when a module is to be designed and realized, or when an activity is to be planned and carried out.

Present realities and visions

Finally a distinction is made between two types of reflective activities: reflections directed at present realities, and reflections directed at future possibilities. To act consciously one must understand the starting point, and at the same time be able to visualize in what direction one wishes to move.

Design leads to visions of the future

Figure 3.3 illustrates the main components of systems development based on the above-mentioned distinctions. The components may be seen both as functions and as activities. Seen as functions the arrows indicate the relationship between functions. Seen as activities the arrows indicate the interaction between activities. Irrespective of what viewpoint is chosen, the figure stresses that the main components of systems development partly consist of analysis, design, realization, evaluation, planning, and regulation; and partly of the relation between these. In any project the specific content of the individual activities and their interaction is concretized and tied in time and space. Correspondingly a method contains a number of suggestions for how this should be done.

The advantage of this abstract description of the main components of systems development is that it is not connected to a specific project model. It emphasizes limitations and options, and it can be employed to evaluate and compare both specific projects and selected methods. The weakness of this abstract description is that it disregards the fact that systems development is enacted in an interaction between many people who, among other things, do not share the same interests, do not have the same resources at their disposal, and do not have the same qualifications. These aspects will be discussed later in this book.

Performance

The actual performance of systems development — the product-oriented part — consists of the reflective functions (analysis and design) and of the innovative function (realization).

The analysis function is directed at present realities — the user organization, technical options, and existing design proposals. The analysis function results in an understanding of the user organization, of technical possibilities, and of design proposals. The analysis function is typically carried out in activities such as the following:

- Interviewing users on their current practices.
- Describing working processes.
- Describing data flows.
- Visiting other organizations to study similar systems.
- Describing technical options.
- Evaluating design proposals.

The design function is directed at future technical and organizational possibilities. The design function includes formulation of one or several visions of a desired change in the user organization. The design function results in descriptions of programs, computer systems, and working processes. The design function is typically carried out in activities such as the following:

- Meetings where ideas are generated.

- Working out tenders.
- Describing functions.
- Description of module structure.
- Overall description of modules.
- Determining system architecture.
- Describing working processes.

Realization leads to actual changes

The realization function is directed at the computer equipment and at the user organization. The realization function results in programs and computer systems, and in changed working practices, qualifications, and attitudes in the user organization. The realization function is typically carried out in activities such as the following:

- Coding programs.
- Integrating and testing programs.
- Implementing new or modified computer systems.
- Changing work organization.
- Training users.
- Conversion.

Performance principles

There are a number of basic characteristics linked to analysis, design, and realization. In the following these characteristics will be formulated in the form of a number of principles. The first principles are linked to the relations or the interaction between analysis, design, and realization (see Figures 3.4 and 3.5).

Principle P1: Analysis and design are mutually dependent, and should therefore be performed concurrently in order to support each other.

Understanding the relevant parts of the user organization and the technological possibilities is a prerequisite for being able to make design realistic. And visions of the technical and organizational change is a prerequisite for being able to delimit the area of analysis and formulate the criteria according to which the relevance of the analysis activity is determined. The logical sequence of first analysis followed by design is often preferred. In practice, however, it is impossible to perform analysis activities without having visions of the new system.

Leave room for moving against the main direction

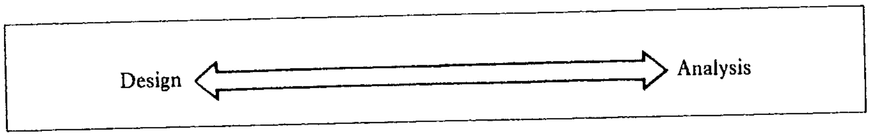


Figure 3.4 The relation between design and analysis.

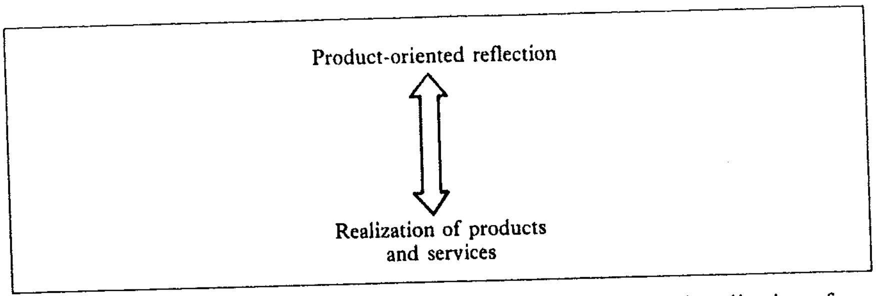


Figure 3.5 The relation between product-oriented reflection and realization of products and services.

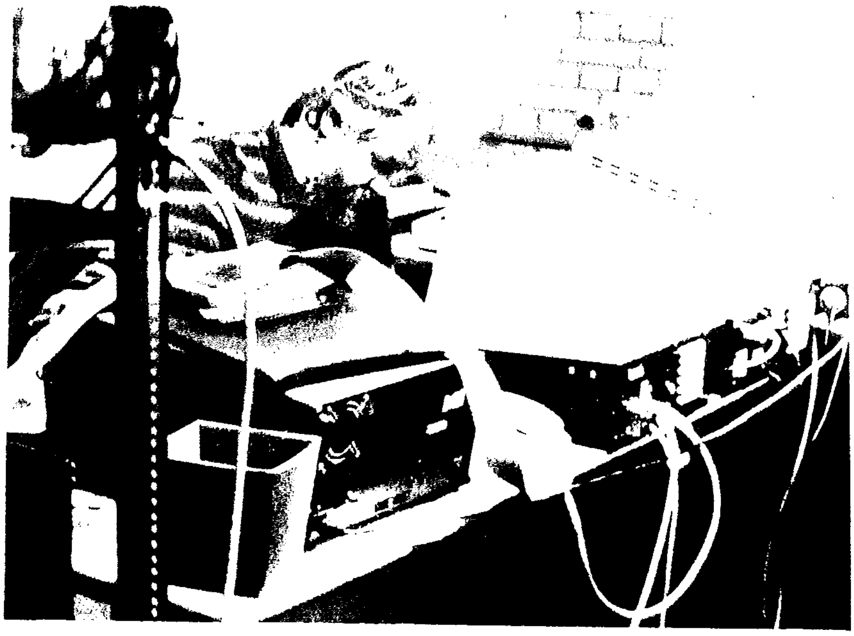
Principle P2: Product oriented reflection (analysis and design) and realization affect each other, and should therefore be performed concurrently in order to support each other.

Understanding the starting point and the desired change is a prerequisite for being able to organize and realize the actual change. And concrete experience gained through the realization activities creates a need for new analyses and designs. The logical sequence here is: first analysis and design, then realization. Test situations in practice will, however, always disclose problems which require renewed analysis or design considerations. It might be a question of errors in the system, but also of constructive suggestions for alternative solutions.

The logical sequence of analysis and design followed by realization can be observed as the main direction in practical systems development (see Figure 3.6). In the beginning of a project emphasis will naturally be on analysis, and as the project moves on, the emphasis will shift to design, and later to realization.

Leave room for moving against the main direction

The two principles mentioned above point towards a piece of practical advice: organize the project so that there is room for moving against the main direction in systems development. This will add insight which will increase efficiency and improve the quality of the product.



Analysis ⟹ Design ⟹ Realization

Figure 3.6 The main direction in systems development.

Planning a test, for instance, is not enough. New analysis, design, and realization activities emanating from the test must also be planned.

So far we have tried to understand the process. But what about the results? Table 3.1 gives an outline of the relation between analysis, design, and realization on the one hand, and the intended results and relations to project contracts on the other hand. In practice the big challenge is to realize these relations with due consideration to efficiency and quality.

Chapters 8, 9, and 10 will discuss analysis, design, and realization. These chapters discuss the following principles concerning the actual performance of systems development:

Principle P3: It is not possible to perform qualified analysis and design strictly according to given guidelines.

Function	Intended results	Intended relation to project contract
Analysis	• Description of the user organization. • Description and evaluation of technical possibilities. • Evaluation of design proposals.	• Forms the basis of creating and changing the product contract. • Contains evaluation of product contracts.
Design	• Overall design. • Functional design. • Technical design.	• Results in product contracts. • Consists of clarification and refinement of the product contract. • Input to creation of process contracts
Realization	• Programs, computer systems. • Changed working practices, qualifications, and attitudes in the user organization.	• Subject to product contracts. • May lead to re-negotiations of product contracts.

Table 3.1 The relation between subfunctions during performance on the one hand and intended relations to results and product contracts on the other.

The activities require experience, intuition, imagination, and reflection.

Principle P4: The project group's relation to the users is of decisive importance to the quality of the analysis. If direct cooperation cannot be established between developers and users, precise requirement specifications and knowledge about the application situation are needed.

Principle P5: Good design is a question of flying high — and keeping both feet firmly on the ground. New visions and unconventional solutions must be created. It must at the same time be possible to introduce them into the organization.

Principle P6:	Technically oriented analysis and design may result in perfect solutions to the wrong problems. Qualified analysis and design require technical, organizational and social competence.
Principle P7:	Qualified analysis and design require that different perspectives are applied. The choice of techniques and tools, and the choice of how to organize the process depend upon the situation.
Principle P8:	Neither the analysis nor the design activities can move unidirectionally from totality to detail. Knowledge of details and of the concrete conditions are prerequisites for obtaining a comprehensive view of the situation.
Principle P9:	Tests do not solve any problems. Plan with time for repairs, thus preventing permanent fire-fighting.
Principle P10:	Qualified realization requires thorough planning of the conversion. The overall design should include the conversion plan.

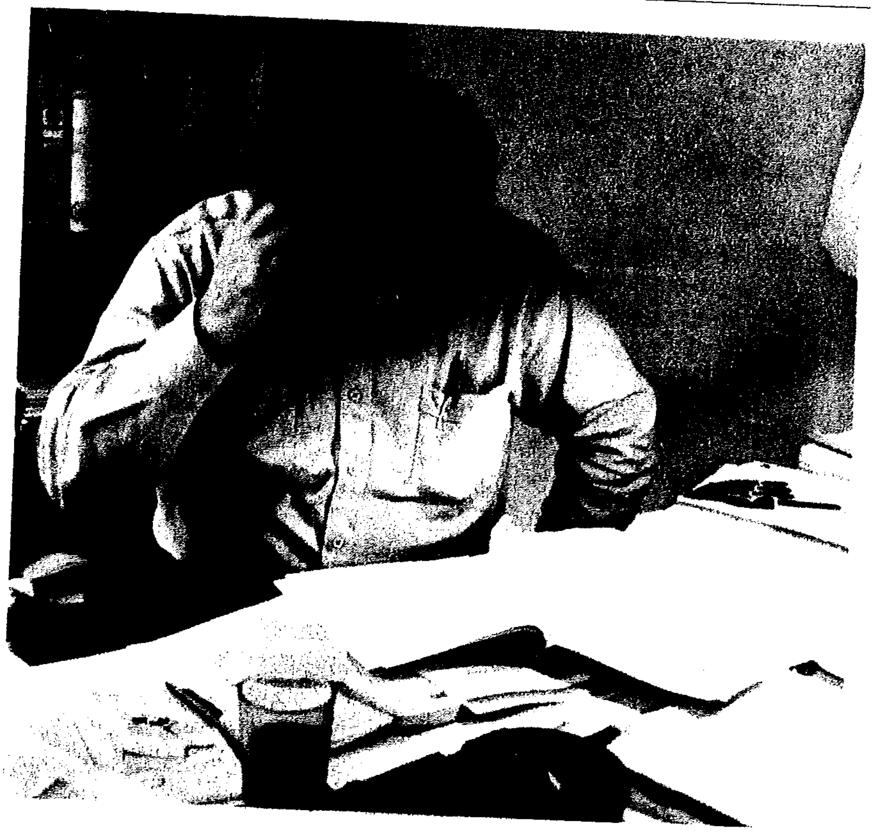
Management

Consider now the management of system development projects — the process-oriented part — which consists of the reflective functions: evaluation and planning, and of the innovative function: regulation. See Figure 3.3.

Evaluation leads to understanding of the situation

The evaluation function is directed at the process itself, and at the current plans. The evaluaton function results in an understanding of the applied working practices and the prevailing conditions of the project. The evaluation function also leads to assessments of the distance between the project's current status and current plans, and it leads to identification of errors, problems, and conflicts linked to the project. The evaluation function is typically carried out in activities such as the following:

- Assessment of status.
- Evaluation of plans.



- Evaluation of meetings.
- Evaluation of progress.
- Reporting to the development and user organizations.

Planning leads to visions about the course of the project

The planning function is directed at future working practices and future conditions of the project. The planning function results in plans on various levels, and in a description of the conditions constituting the prerequisites of realizing the plan. The planning function is typically carried out in activities such as the following:

- Establishing the project.
- Overall planning.
- Detailed planning.

Regulation results in changes

The regulation function is directed at the process in a broad sense. The regulation function is directed at the participants' expectations, attitudes, and qualifications, at working practices in the project, and at the

conditions of the project. The regulation function results in change in the process, and is typically carried out in activities such as th following:

- Establishing the project.
- Project group meetings.
- Negotiations with the development and user organizations.
- Evaluation activities.
- Courses and training.

Management principles

A number of basic principles are linked to management of systen development projects. The first of these deal with the relation o interaction between evaluation, planning, and regulation (see Figure 3.7 and 3.8), and they correspond to the equivalent principles o performance.

Principle M1: Evaluation and planning are mutually dependent and should therefore be performed concurrentl in order to support each other.

Having insight into the project and knowing the relation betwee the project and current plans is, on the one hand, a prerequisite fo

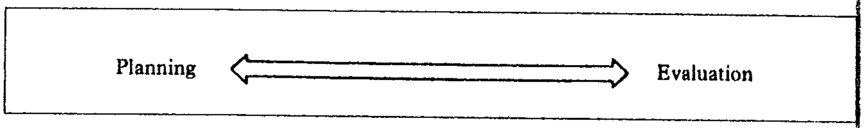


Figure 3.7 The relation between planning and evaluation.

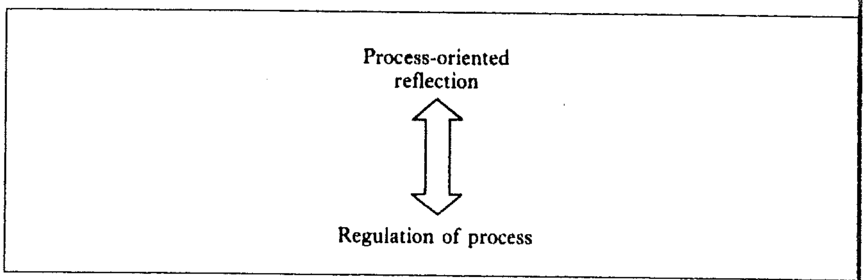


Figure 3.8 The relation between process-oriented reflection and regulation of the process.

being able to make realistic plans. Plans, on the other hand, are a prerequisite for being able to delimit and direct the evaluation of the project. In other words: idealistic plans merely collect dust on the shelves, and project evaluations with no relation to the objectives of the project seldom find a willing ear.

Principle M2:	Process-oriented reflection (planning and evaluation) and regulation affect each other, and should therefore be performed concurrently in order to support each other.

Understanding and accepting both the planned and the actual course of the project is a prerequisite for being able to regulate the project in the required direction. And concrete experience with regulating the course of the project creates a need for new evaluation and planning activities. In other words, it is unwise to act in the dark, and it is naive to think that it is possible to see all the options and consequences from the outset when the project is planned.

Table 3.2 illustrates the interaction between management functions on the one hand and the intended results and relations to the project contracts on the other.

Chapters 4, 5, 6, and 7 will discuss management of systems development projects. This is an extensive and important field of study, and the chapters do not attempt to give an exhaustive presentation. However, the selected subjects will emphasize and illuminate the following principles concerning management of projects:

Principle M3:	Systems development is characterized by a high degree of uncertainty. The most important prerequisite for qualified management is therefore transparency in both processes and products.
Principle M4:	It pays to establish the project systematically.
Principle M5:	Baselines and checkpoints are better than traditional phases. Traditional division into phases confuses time and content, thus making dynamic regulation of the project difficult.
Principle M6:	Project plans must facilitate evaluations. They must be in writing and contain evaluation criteria and procedures.

Function	Intended results	Intended relations to project contracts
Evaluation	• Understanding working practices and conditions. • Evaluation of distance between plans and status. • Identification of errors, problems, and conflicts.	• Forms the basis of creating and changing the process contract • Evaluation of the process contract
Planning	• Plans on different levels. • Description of conditions required by the plan.	• Based on the product contract. • Results in the process contract. • Consists of clarification and refinement of the process contract.
Regulation	• Changed working practices. • Changed qualifications, attitudes, and expectations of participants. • Changed conditions for the project, including changed contracts.	• Subject to the process contract. • May result in re-negotiation of the process contract. • Affects internal contracts.

Table 3.2 The interaction between management functions on the one hand and the intended results and relations to the project contracts on the other.

Principle M7: Only the system developers know enough to make realistic plans and to evaluate the status of the project.

Principle M8: It is important that all participants in a project understand and accept the plan.

Principle M9. It is necessary to apply several estimation techniques. The plan should be based on a probable estimate, and express the degree of uncertainty in the estimate.

Principle M10: It is necessary to plan with management activities. Management typically constitutes 15% to 25% of the effort.

Management and performance

Systems development is characterized by a high degree of uncertainty

The last, but perhaps most important, component of systems development is the relation between management and performance (see Figure 3.9 and Figure 3.3).

System development projects are typically characterized by a high degree of uncertainty. This is expressed in the following fundamental principle:

Principle PM1: A system development project should be organized in a way which ensures direct and close interaction between performance and management activities.

Unforeseen and problematic situations typically occur during a project. Managing the project effectively therefore requires up to date and thorough insight into the project. Effective management can neither be realized through directives or standards, nor by people who do not themselves participate in the actual performance of the project. In other words: it is impossible to manage system development projects effectively with traditional bureaucratic means.

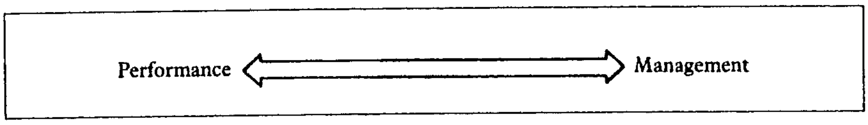


Figure 3.9 The relation between performance and management.

Principle PM2: The most important intermediate products are the plan and the overall design.

There is evidently a strong analogy between performance and management. The plan is the key document in management. The overall design is the key document in performance. Poor management is characterized by poor plans and poor evaluation — and this leads to projects characterized by randomness and confusion. Poor performance is characterized by poor analysis and design — and this easily leads to unnecessary firefighting. If the analogy is carried a little further, it will be seen that project evaluation must identify those areas in the project which require new plans and regulation. Analysis must correspondingly identify those areas in the user organization and in the new system which require new design reflections and intervention.

This analogy may seem artificial on first sight. What is expressed, however, is merely a fundamental view on the elements entering into understanding and changing processes. Performance and management both aim at processes: management at the process in the project, and performance at the process which is the users' work. From this point of view it seems natural to look at performance and management in the same manner.

Decision-making, communication and socialization

General functions

In addition to the characteristic components listed above, there are a number of more general functions. Here decision-making, communication and socialization will be considered. They are functions which appear as important components in most working situations. Figure 3.10 gives an outline of the total set of subfunctions of systems development.

Many decisions are made in a system development project. The decisions deal with both the process and the products, and they influence the project contracts in a broad sense. Some decisions are made internally in the project group. Other decisions involve external groups and people. Some decisions are a result of negotiations. Others are a result of cooperation. It is an important experience that decisions may take time,

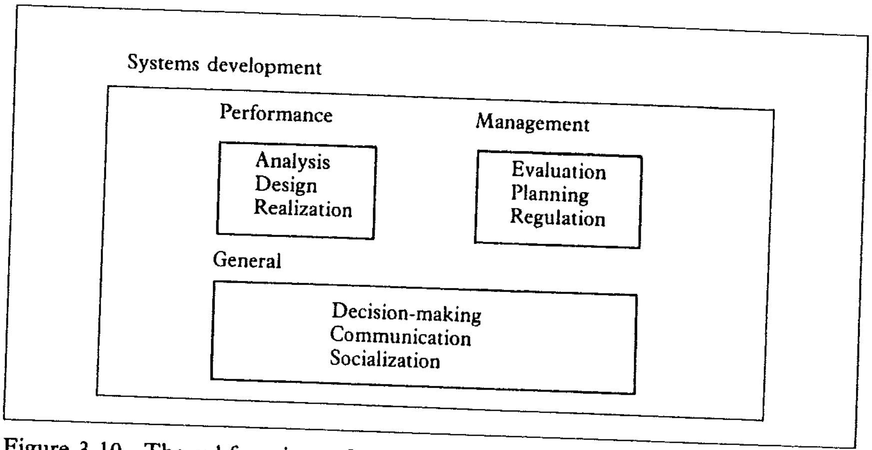


Figure 3.10 The subfunctions of systems development.

and that they may delay the project. A good plan takes this fact into consideration.

Communication is a necessary and decisive function simply because a system development project involves the participation of several people. Communication may be characterized by its content, form, and location in relation to the project. The content of communication deals with the project contracts in a broad sense — it deals with both the process and the products. The location of communication depends on the sender and the receiver: whether, for instance, the communication takes place between people or groups who both work in the project, or whether only one of the parties works with systems development. The form of communication depends on the language and the medium applied, and on whether the communication is one-way or a dialogue. It is an important experience that communication problems are often severe. One of the reasons for this is that computer technology requires formalized descriptions which differ from everyday language.

Socialization is also a decisive function simply because several people participate in the project. Socialization first and foremost takes place when the project is established, on project group meetings, and through the social processes in the project. The socialization function is primarily directed at the project's internal and informal contracts. It is a common mistake that the socialization function is given low priority. This may lead to serious problems in relation to both results and activities. Management activities must therefore be designed to contribute to the performance of this subfunction.

Exercise 3.1

Discuss the subfunctions in a project

Analyse the last project in whch you participated.

- How much emphasis was given to the various subfunctions?
- How did the interaction between the subfunctions work?
- Could some of the problems, errors, or conflicts in the project have been handled better by performing the subfunctions differently?

Exercise 3.2

Discuss the subfunctions in a method

Analyse one of the methods described in the literature or in the project handbook of your company.

- How are the various subfunctions weighted?
- How is the interaction between the subfunctions described?
- What types of problems, errors, and conflicts is the method poor at handling?

3.3 Views on systems development

So far this chapter has presented a number of concepts and fundamental principles for understanding and handling systems development processes. You could say that system developers often know a lot about what they are working with, but rarely very much about their work. That is why they find it difficult to discuss and exchange experience, and to let new methods affect their traditional working practices.

This section will look at the systems development process from various viewpoints. Each viewpoint interprets systems development as a specific kind of process. General knowledge about these different types of processes can serve as a base to increase understanding of the problems that typically surface in a project. And a more realistic attitude can be adopted to important conditions and possible working practices.

Construction, change, cognition, and policies

First, systems development can be looked at as a process of construction. The intention is to construct a computer system – and this system is to be integrated in the user organization. Systems development is a kind of organizational development, but it is the aspect of construction which gives the process its specific characteristics. The formalized descriptions necessary for realizing the computer system must live up to strict demands. Insight into the technical equipment to be applied, and into any other systems with relevance for the project, is a prerequisite. The complexity is often so great that systems development not only requires technical insight from its performers, but also that they have a solid talent for abstraction and systematization.

Secondly, systems development can be viewed as a technical and organizational change process. The fundamental intention of a systems development project is to change the technological, formal, and social structures in the user organization. The process itself also requires that structures are established (the project contracts) within the frames of which the process can take place. These structures must be maintained and changed throughout the project. Social innovation processes are basically directed at human beings and human relations. Technologies are, in this connection, only a means. Qualified innovative work requires a good deal of human and organizational insight.

Thirdly, systems development can be viewed as a process of cognition. On the one hand something must be learned about the user organization and about the technical and organizational possibilities for changing it. On the other hand something must be learned about the setting in which the process is to take place, and about the technical and organizational possibilities for designing the project. Processes of cognition cannot be reduced to routine work. They require experiments, an open mind, and inspiration. That is one of the reasons why systems development projects must be organized in a way which ensures close interaction between performing and managing activities.

Fourthly, systems development can be viewed as a political process. The various participating groups and individuals only to a limited degree share a common goal. They see system development projects as an opportunity to promote their own interests, maybe even at the cost of other people's interests. Conflicts which have been hidden so far may surface and affect the project alongside with the already acknowledged conflicts. The political process may take place in an atmosphere of cooperation, of negotiation, or of struggle, depending on what kind of process it is and on the strength of the participating parties. The professional system developer does not try to conceal the political process, but lets it take place in suitable environments.