An Integrated Procurement Process In Systems Development
- On Requirements Engineering in a Wider Perspective -

Peter Hökenhammar, PhD
Departement of Computer and Systems Sciences
Stockholm University and Royal Institute of Technology
Isafjordsgatan 39 Forum 100 S - 164 40 Kista
+49 8 16 16 52 hoek@dsv.su.se

Abstract
This work deals with information systems that support administrative processes and are developed by companies themselves, i.e. by a department or a function in the company formulating a specification and ordering the realisation of a system from the IT-department. Many systems development projects do have considerable problems, mainly caused by shortcomings in the initial phase, named the requirements phase. Thus, in a customer-/ supplier relationship the shortcomings are mainly related to the customer.

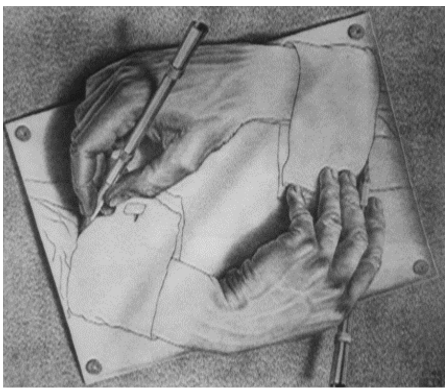
The perspective in the requirements process is expanded by integrating a wider range of phases in the development process, by widening the scope of analysed artefacts and by widening the circle of participating stakeholders. To reflect this, the concept of a procurement-process is introduced instead of requirements-process. An integrated way of working is defined, based upon an analysis model where different kinds of information in the procurement process are categorised and can be stored, related and analysed. In addition to ordinary system requirements, such as functional requirements, also underlying information is regarded, for example description of the work processes to be supported and overall objectives of the company.

In three case studies it is demonstrated that relationships in form of directed binary dependencies are extensive as they can be found amongst information in the categories in every combination. Based on this an integrated way of working is defined, meaning that all categories of information should be taken into consideration in parallel throughout the procurement process. We also describe how the integrated way of working could be applied in practical work by a computer based supporting tool constituting a workspace, in which the appropriate stakeholders can co-operate throughout the procurement process.

Based upon this we also shortly discuss possible contributions to some common goals in information systems development.

The paper is a summary of my Ph.D. thesis presented at The University of Stockholm 2001.

Keywords
Systems Development, Requirements Engineering, Requirement, Procurement, Viewpoint, Stakeholder, Interest, Collaboration, Specification, Integrated process



"Drawing hands" by M. C. Echer
The picture symbolizes idea in the work
- The influence is reciprocal in an integrated process -

Introduction

By information systems development we mean when companies by themselves develop and put to work IT that supports administrative processes, e.g. ticket-booking and stock-control. Somewhat simplified the development process can be divided into two main phases: specifying the system and realizing it. Phases that follow, which are not strictly development work, are putting the system into work and the maintenance of it. This work deals with the specification phase.

Why focus on the early part? Information systems development projects often face a lot of different problems, e.g. lack of quality in the system, cost overruns and time delays in the project. An extensive study [12] of US companies shows among other things that 31 percent of all projects are interrupted and that 52 percent have a budget overrun by more than 89 percent. The following explanations are mentioned, sorted descending by importance: Incomplete requirements, lack of user involvement, lack of resources, unrealistic expectations, lack of executive support, changing requirements & specifications, lack of planning, didn't need it any longer, lack of IT management, technology illiteracy. In another study on European software industry [6] the following areas are reported as a major problem, sorted descending: Requirements specification, managing customer requirements, documentation, testing, lack of quality system, project management, lack of standards, systems analysis and design, configuration management, installation and support, programming**.** The importance of specifications is also emphasized in an academic thesis [9]: "55 percent of all mistakes in information systems development projects were made in the specification phase and only 18 percent of them were noticed in this phase. 48 percent of the mistakes where due to obscurities and misunderstandings in connection to the specification". Thus in a customer/supplier perspective in information systems development the shortcomings are mainly related to the customer and to the specification work.

The specification document results from different activities, normally carried out in sequence, starting with objectives in the company and resulting in mainly functional requirements. Different kinds of stakeholders participate in the process, like managers and end users. Different areas are included in the analysis like company goals, business processes and technical and economic issues. The aim of this work is to argue for the advantage of an integrated and parallel process by widening the range of phases in the development process, by widening the scope of analysed artefacts and by widening the circle of participating stakeholders. The concept Requirements-process is replaced by Procurement-process in order to emphasize the expanded perspective.

Starting from some theories in development and requirement work, shortly presented at the end of the paper, an analysis model is developed, containing different categories of requirements and other types of information, related to the specification process and to different kinds of stakeholders. The model is used in arguing for the necessity of cooperation throughout the process and is also intended to be used as an analyze tool in a development project.

This paper is mainly a summary of my doctoral thesis [15] " An Integrated Procurement Process in Systems Development - On Requirements Engineering in a Wider Perspective -", presented at Stockholm University 2001.

The information system - its development and its role in the company

Information systems development can not be looked upon as an isolated discipline as it is normally a part in a wider change process in the company. Or according to Hällström [4]: "The aim is not longer to produce a detailed requirements specification but to quickly get a basis for decisions on different IT-investments enabling dramatically changed business processes". Figure 1 shows the role of the information system as we see it.

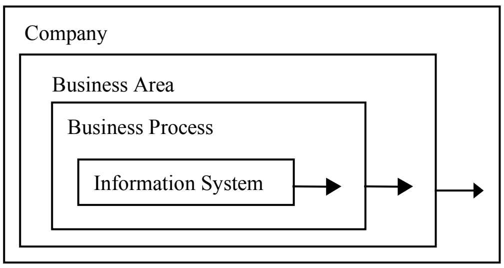


Figure 1. Information system and its context

Business-area means a part of the company, e.g. finance, and Business-process means work that is performed and supported by IT, e.g. bookkeeping in the finance department or a workflow from customer order to shipment. An arrow means Supports.

What is an information system? As we see it the main parts are application programs, system programs (Op erating- and communication system etc.) and computer hardware. Additional artifacts such as documentation, operational procedures and training may also be included. 'IEEE Standard Glossary of Software Engineering Terminology' [7] gives the following definition of a system: "A collection of components, organized to perform a special function or a set of function".

The specification (contract) of a system handed over to the IT-department for realization is a result of a process normally containing activities such as analyzing objectives in the company, defining business processes, defining objectives of the new system, defining functional requirement and quality attributes of the system and finally analyzing feasibility of different kinds such as technical-, economic- and project issues. This concerns stakeholders of different kinds such as top managers, managers of business processes, system users, system developers and project leaders.

The specification often is called the Software requirements specifications and should according to *'IEEE Recommended Practice for Software Requirements Specifications'* [8] among other things contain background information to the requirements - e.g. goals in the company and its business processes, wanted functions in a system, design restrictions, non-functional requirements, e.g. quality attributes in the system like usability and robustness, and hardware requirements.

This section is meant as a background to what should be taken into consideration in the procurement process and to the parts in the analysis model, presented in the next section.

The analysis model and its categories of information

The procurement process is of a creative nature and is performed in collaboration among different kinds of stakeholders. Information systems development normally is a part of a broader process of change in the company where business processes and IT should be developed together in a businesslike way in accordance with objectives, strategies and restrictions of different kinds in the company. This implies that different kinds of information must be taken into consideration in the procurement process like general business restrictions in the company and functional requirements. This motivates the concept Statement instead of Requirement. Figure 2 shows the categories, named aspects, used to categorize statements made in the procurement process. The statements are stored as objects within aspects. A statement could be defined by identity, name, description, classification, reference to other documents, estimated cost and time, owner and date-created. The shadowed part is the main content of the requirements specification to be realized.

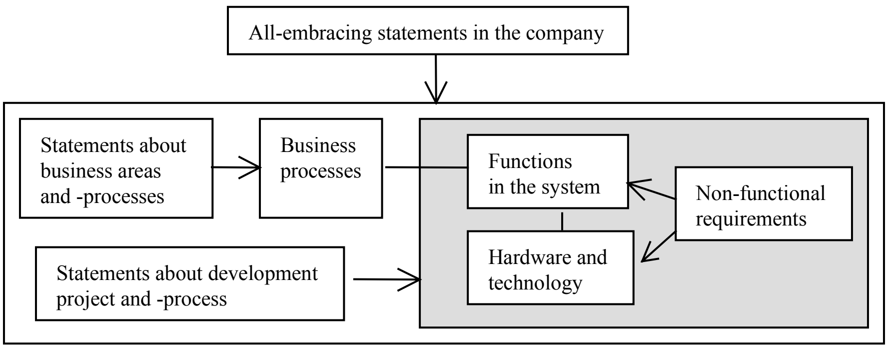


Figure 2. The analysis model

The aspects are shortly described below. The intention is that the first three should be developed and refined to a requirements specification, while the remaining ones should be looked upon as underlying causes, from the viewpoint of an information systems developer.

- *Functions in the system* is the core part in a specification and shall be refined to functional requirements, e.g. order-entry
- *Hardware and technology* is needed to run the programs (functions) and is seen as a part of the system. Technology means operating system, database handling system etc.
- *Non-functional requirements* means quality attributes in the system like usability and flexibility
- *Business processes* means work in the company to be supported by IT, e.g. the order-entry process or a wider work-flow.
- *All-embracing statements in the company* refers to statements that are superior to all activities in the company and also having relevance to the system and the development project. We do not use the more common concept Objectives as statements from management can be restrictions, recommendations and many other things. We make a distinction between general business issues in areas like business/economy, products and

services, management and organization, infrastructure and quality and between IT-related issues like infrastructure/architecture, technology and supplier issues. A general business related statement could be “Costs allover should be decreases by 20 % in the company” and a IT-related statement could be “All computer equipment should be bought from supplier X in order to get overall compatibility and a consistent technical infrastructure”.
- *Statements about business areas and -processes* means all-embracing statements on work performed in a business area like “Ticket-booking should be possible non-stop 24 hours per day”. Statements can address different areas like: profitability, effectiveness, safety and security, quality and accessibility.
- *Statements about development project and -process* bears upon the realization phase. Examples of statements are: “Overall secrecy roles should be paid attention to in the project” and “RUP should be used as the development process”.

As we use the general concept Statement instead of Requirement there is a need for some classification of statement objects. The ones used are: strength, accuracy and sign, are presented below

STRENGTH
- *Idea* is a statement representing a wish or something to investigate like “Defects in deliveries from our suppliers should be decreased”.
- *Goal* is a statement representing something to strive towards like “Defects in deliveries from our suppliers should be below 1 %”.
- *Requirement* is a statement representing something that has been decided like “Defects in deliveries from our suppliers to plant A must be below 1 % and shall be tracked by a statistical control procedure”.

ACCURACY
- *Name* is a statement representing a wish that is not yet defined like “Our company should examine incoming goods”
- *Specification* is a statement representing a wish that has been defined like “Goods from our suppliers should be examined by random sampling with algorithm x”
- *Product* is a statement representing a wish that can be bought or that already exists like “Information from examined arrived goods should be processed with the program StatView”

SIGN
- *Positive* means something we want like “Satisfied customers” or “A program for order-entry”
- *Negative* means something undesired like “Supplier X must not be contracted” or “Training of users should not be included in the undertakings of the project”

An example of a classified statement within the aspect *Functions in the system* could be: “A program with functions A and B (Specified in an appendix) shall be developed”. Classification STRENGTH = Requirement, ACCURACY = Specification, SIGN = Positive.

One main idea with the analysis model and its aspects is that all kinds of statements continuously can be recorded and successively refined. Thus a statement can change, e.g. from a vague idea to a defined an named product. Commonly there are different models in different phases in the procurement process like ideas, objectives, requirements and specifications [8, 11]. Well known concepts are Requirements definition, Requirements specification and Software specification. Or are they well known? Davis [2] observes the lack of consistent definitions of the concepts Requirement, Specification and Design*,* based on a study of 13 well known methods and standards. An advantage in working in one model, as we see it, is that dependency and influence from phases in systems development methods is reduced. To well defined phases may cause trouble in practical situations. Working in a more seamless way may allow interaction and dependencies among general statements, not yet developed, and already specified artifacts, and vice versa.

Relationships among different kinds of statements

Normally every statement in the analysis model is involved in some kind of dependency or relationship to one or more other statements, e.g. two technical components being incompatible, a business process demanding a certain IS-component or a technical approach being to risky. Thus the procurement process often must deal with negotiations and must handle different kinds of trade-offs. Below we define the binary relationships that are used in the analysis model. It is important to notice that they are directed in time, a fact that is used when the integrated procurement process is defined and motivated later on. A directed relationship from statement A to B implies that A should be known before B in the process, or at least at the same time if A influences B in some way. The following dependencies are used in the model and they are meant to be able to deal with dependency in time:

E = Existential dependency
This means that a statement in the model causes another statement to be born and added. Three situations are focused:

1. A *motivates/ demands* B, because B contributes to A, e.g. business process Order demands the IS-function Order-entry.
2. A *makes possible* B, e.g. Internet possibilities makes the business process International sales possible.
3. A *implicates* B (*logical conclusion*), e.g. if supplier A is chosen for computer products, then program X must be used.

D = Defining dependency
This means that a statement in the model influences how another object is defined and designed, e.g. the requirement 'Short response time' influences the design of a program. A may also restrict B, e.g. the capacity of a server may restrict the possible number of clients in the system.

O = Other types of dependency
The meaning of this can be that two objects are contradictory, e.g. a desired database system does not fit with the operating system in use.

The objectification of statement and dependency is shown in figure 3

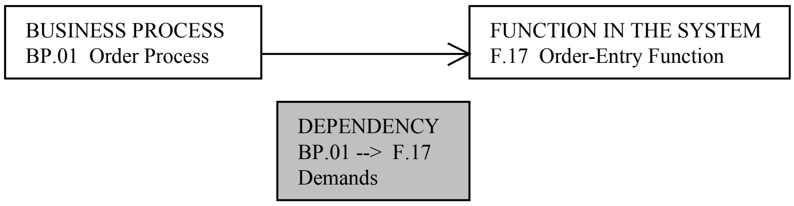


Figure 3. Dependency between two objects

The extent of dependencies

In my thesis [15] the requirement situation was reconstructed in three case studies of running systems. Statements in the seven aspects and dependencies among them where registered in the model. The matrix in figure 4 shows the possible combinations of dependencies. The direction in time of a dependency goes from vertical to horizontal. The shown arrow indicates a statement about a business process influencing a non-functional requirement, e.g. traffic control must work continuously and demands high availability in the system. This dependency is of type E1.

Dependency ----→	All-embracing statements in the company	Statements about business areas and -processes	Business processes	Functions in the system	Non-functional requirements	Hardware and technology	Statements about devel-opment project and -process
All-embracing statements in the company	X						
Statements about business areas and -processes		X					
Business processes			X		→		
Functions in the system				X			
Non-functional requirements					X		
Hardware and technology						X	
Statements about devel-opment project and -process							X

Figure 4. Possible dependencies

Some of the binary directed dependencies among the aspects may not be surprising, namely the ones on a traditional sequential line in the process, which might be in the following order:

1. All-embracing statements in the company
2. Statements on business areas and -processes
3. Business processes and non-functional requirements
4. Functions in the system
5. Hardware and technology
6. Statements on development project and -process

These may be called dependencies *with the stream*, e.g. a business process motivates a certain function in the system. But, in the case studies there is demonstrated that there are also extensive dependencies *against the stream*, which perhaps may be a little surprisingly. These dependencies against the stream are often contradictions and restrictions, e.g. preferred choices of suppliers in the IT-department may be contradictory to and influence the overall policy for collaboration with suppliers in the company. There can exist other dependencies against the stream, e.g. the architecture in the computer system might give unexpected possibilities for decentralizing the decision process in the company. These dependencies against the stream often causes iterations and rework in a project. One idea of using the analysis model is to reduce the numbers of iterations and rework in the procurement process. As can be seen in the matrix in figure 4, most of the dependencies below the marked diagonal could be regarded as *against the stream*, while those above can be classified as *against the stream*.

The dependency between for example a business process and a system function could be characterized as *direct* as they are adjacent in the sequential perspective, mentioned above. The question then arises if dependencies between two nonadjacent ones have to be classified as *indirect*. We claim that it is not necessarily so. An example could be an all-embracing statements in the company saying that a certain software supplier should be used, which may implicate that a certain software function has to be realized by a special product from the supplier chosen. Dependencies can also arise in different phases in the process. Early strategic reasoning can point out general ideas about for example external communication, which can be sketched at an early stage in the process. A business process, specified later may rise specific requirements on the communication. From this example we can also see that there might be several motivations to the existence and to the design of an object. Below are some examples of dependencies (arrow shows direction and type of dependency is marked within parenthesis) in the matrix in figure 4

Business processes --> All-embracing statements in the company (Against the stream)
A business process is dependent on other business processes, which may require common codes and concepts all over the company (E1).

Business processes --> Statements on development project and -process (With the stream)
A business process handling delivery and customs clearance may require special competence in the project team (E1). Another example is that the interaction between the operator and the system in a business process may require influence from users in the design work and that prototypes of the system are being used (E1).

Non-functional requirements --> All-embracing statements in the company (Against the stream)
Very high requirements on system performance may increase the costs, which may contradict all-embracing requirements on lowered costs (A).

Statements on development project and -process --> Statements on business areas and -processes (Against the stream)
The need for experimental learning in the development process may contradict business requirements on an early start of putting the system in work (A).

In three case studies in the thesis [15] there is demonstrated that binary directed dependencies among statements in different aspects are possible and probable in all directions and between all aspects, meaning that all fields in the matrix in figure x can be associated with meaningful information in a single project.

The integrated procurement process as a result from the extensive dependencies

As a result from the above mentioned extensive dependencies follows that statements in every aspect may be influenced from every other aspect. This leads to the conclusion that

All aspects should be considered in parallel throughout the procurement process.

This indicates a need for continuous collaboration. Thus the integrated process could also be formulated so that all the stakeholders, from top managers to IT specialists, should be engaged and co-operate throughout the procurement process. The main stakeholders could be: top managers, business process managers, system users, system developers, IT- strategists, project leaders and leaders of development processes. To make this continuous collaboration possible there is a need for a tool to register and store the statements and the dependencies and making them visible. This is covered below in the practical application. In a dictionary the concept Integrate is explained as: join, bring together to a whole, co-ordinate and make complete by incorporation.

Thus the procurement process should not be sequential, engaging different stakeholders in different phases. As a result of the integrated process requirements normally stated early in the process are visible, alive and could be taken into consideration throughout the process, e.g. all-embracing statements in the company. Inversely results down the process can continuously be related to earlier produced information. . Often in a sequential process the results from one phase are only looked upon in the next phase Also early awareness of requirements, statements and facts that normally arise later in the process can be interesting, e.g. hardware specifications, in case the existing hardware is to be used.

One might also ask if it is possible to make an early statement about something before all prerequisites are known. We claim that this is possible and preferable if there is an instrument for handling different levels of refinement, like the classification of statements described earlier. For example it could be noticed early on that a system entry via Internet is required, without knowing how it should be realized. This statement might motivate investigations about safety and security that might highlight risks and consequential costs. This in turn may raise doubts about an internet connection and furthermore doubts about the whole project. The advantage of the early statement in this case was that the threats and the doubts could be identified at an early stage, giving the opportunity to rethink the whole idea. This without having invested to much money in a blind track.

Another motivation for co-operation throughout the procurement process among all stakeholders is that binary dependencies among statements may occur in chains and that one statement may be related to several others in different aspects and thus concern a number of stakeholders. Figure 5 shows an example of a wholesaler wanting a faster flow of materials, which could be accomplished by delivery directly from supplier to customer. This requires system functions performing communication with suppliers. This might require computer capacity that is not there. The advantage in an integrated process in this case could be that major problems could be early detected and more realistic solutions could thus be investigated.

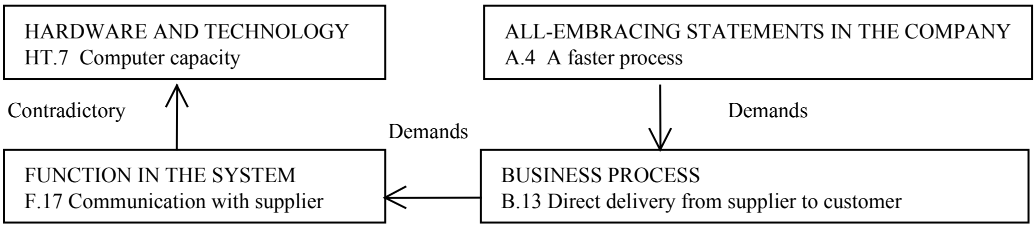


Figure 5. Chains of dependencies

A model doesn't have to contain fixed categories. Kotonya [26] discusses the advantage of flexibility in his work "Requirements engineering with viewpoints" and makes a distinction between direct viewpoints, mainly functional requirements, and indirect viewpoints for back-ground information. There can also be an interest of other features in a requirements model. Gotel [24] stresses the importance of tracking the evolvement of reguirements. In "Modeling the evolution of artifacts" [25] Rolland takes this further by recommending also to store information about decisions made in the requirements model.

The integrated procurement process in practice

The integrated process could in practice be implemented as a repository - parallel, coordinated and linked to information in the systems development method and -tool in use. The repository could work as a surface for collaboration, negotiation, balancing and trade-off where stakeholders can enter their statements of different kinds and relate them to statements by others, which can be seen in figure 6.

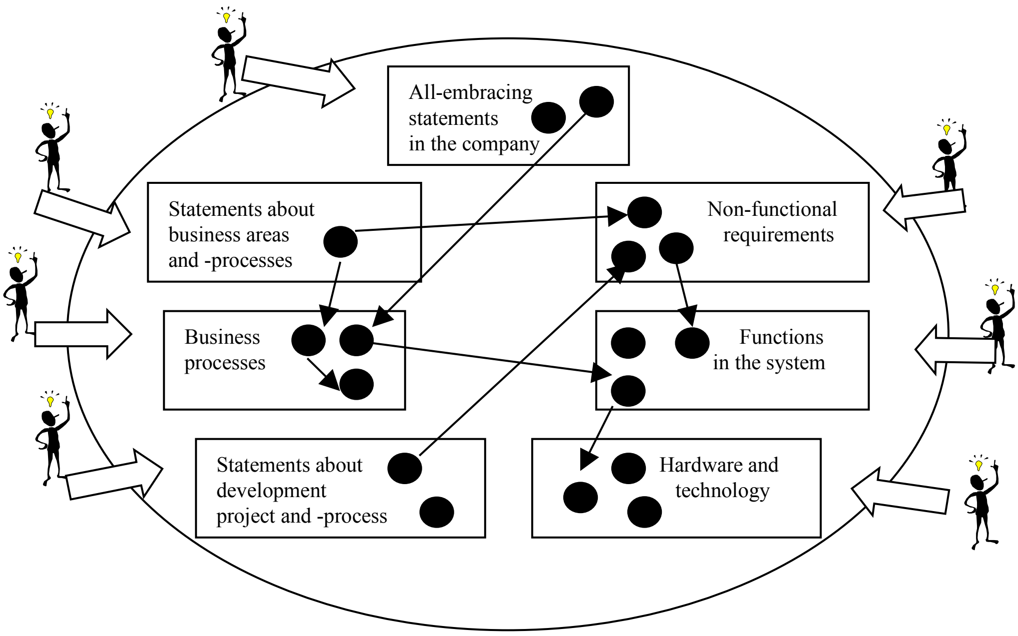


Figure 6. The analysis model as a surface for collaboration

Figure 7 shows a simple non-graphical input form to the repository.

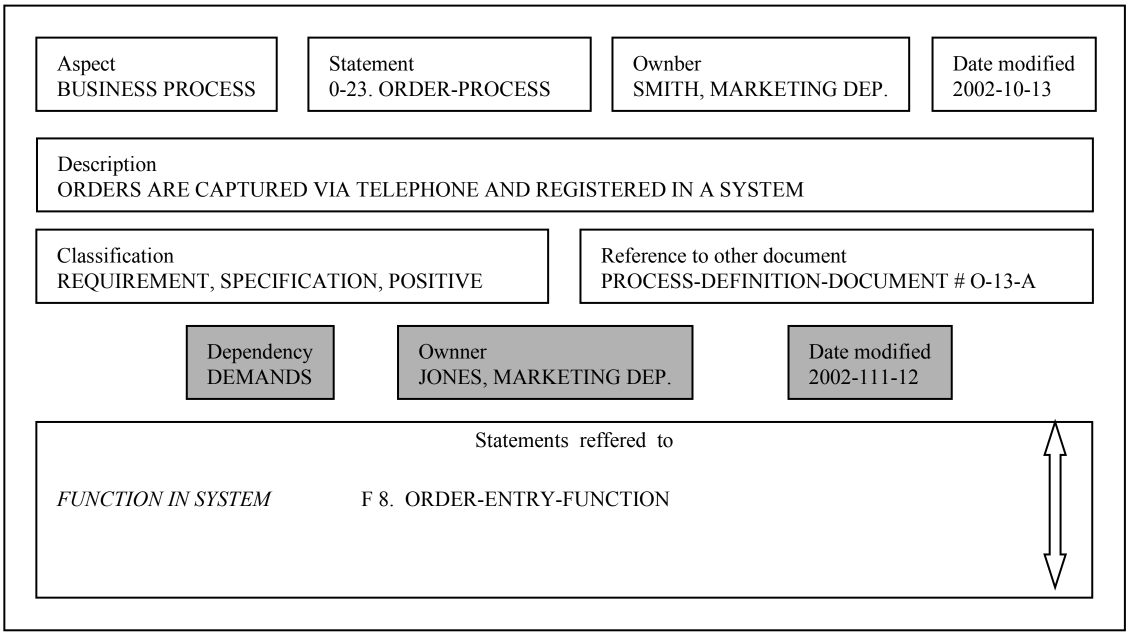


Figure 7. Input form to the analysis model

As output can different kinds of analysis be produced, e.g. "Which objects in chains do point to a specific object?" giving tractability or "To which objects in chain does a specific object refer?" which can be interesting in order to see the consequences of a change. The former case is illustrated in figure 8.

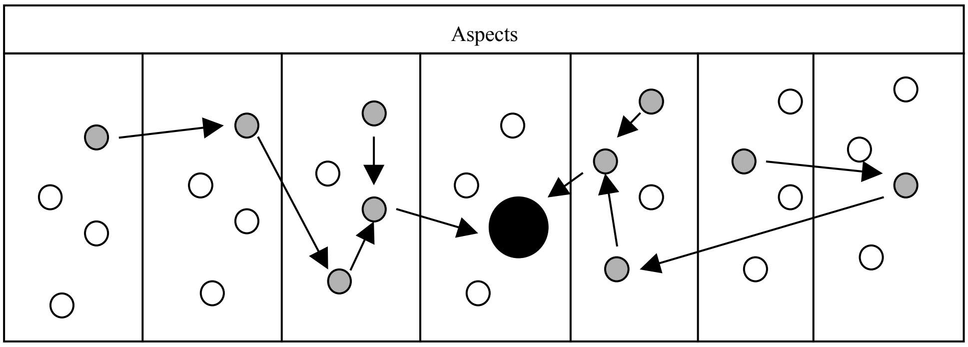


Figure 8. Chains of objects referring to a specific object

The integrated process fits to the following definition of Cooperative Work: "Cooperative work takes place when multiple actors are needed for doing the work and that they are mutual dependent on each others work and therefor must integrate and coordinate individual activities in order to get the work done"[10]. In "The Machine that Changed The World" [14] the following factors are pointed out for successful product development: leadership, teamwork across functional boarders, communication in a project, front-end-loading meaning that decisive problems are solved at the beginning and parallel development.

There are well established methods for collaboration like JAD (Joint Application Design) [17] based on structured meetings with users and IT professionals, Prototyping where system ideas are illustrated [23], RAD (Rapid Application Development) [18] with focus on development time and PD (Participatory Design) [19], focusing on harmonization between the social and the technical system. There are also supporting tools for the requirements process such as CASE (Computer Aided Software Engineering) [20], DOORS (Dynamic Objecto-riented Requirements System) [21] and RTM (Requirements and Traceability Management System) [22].

The effects of the integrated procurement process

In my thesis [15] there is argued that the integrated process may contribute to the requirement process in some different ways: to apply a *comprehensive holistic view* by the extent of the different aspects, to facilitate *communication among stakeholders* by working on a common surface, to facilitate *continuity among different phases in the development process* by the possibility to make statements with different levels of refinement and accuracy, to *add knowledge* by the extensive engagement (and even better: commitment), to *facilitate understanding* by confronting actors with statements and fact from others, to *add objects and dependencies* by the extensive participation in the creative process and finally to *facilitate decision making* in a project by the structured information in the repository giving a good foundation for decisions.

Furthermore it is argued that the integrated process could contribute to quality criteria, mentioned in 'IEEE Recommended Practice for Software Requirements Specifications' [8] : correct, nonambiguous, complete, consistent, ranked for importance and/or stability, verifiable, modifiable and traceable.

Based on the contribution to the process and to quality attributes it is argued that the integrated process could contribute to some goals related to systems development: the system should fit the company, the system should fit the business and its processes, advantage should be taken of possibilities given by IT, non-functional requirements on the system should be focused, the project should meet up to agreements on time and cost and finally should the process be handled in a business like way by addressing profitability calculation, risks, trade-offs etc.

Does user involvement guarantee a successful project? Barki [3] for example did not find a strong correlation and he stresses the distinction between participation and involvement

Theoretical background

Three works mainly constitute a theoretical background in the thesis [15]:

The first is *the F^3 theory* (From Fuzzy to Formal) [16] based on an *Enterprise Model* with the following components: Objectives model , Concepts model , Activities and Usage Model , Actors model , Information system requirements model and Information systems model. Within these models objects and dependencies of different kind are stored. F^3 focuses on the transition from vaguely formulated requirements and goals towards more formal specifications and that the requirements process should be grounded in an understanding of the problem domain. The F^3 theory contributes to the extensive perspective and to the collaboration with help from different models.

The second is the concept of *Concurrent Engineering* (CE) coming from Winner[13]: "Concurrent Engineering is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset, to consider all elements of the products life cycle from conception through disposal, including cost, schedule, and user requirement". Bergman and Öhlund [1] transform these ideas to systems development, called *Integrated systems development* looking upon systems development as a sort of product development. According to the authors it addresses the following dimensions: *Parallel development* in order to decrease time, *Integrated problem solving* by a creative and parallel process with problem solving and feedback between different areas of knowledge, *Co-ordination of knowledge* by cross functional teamwork to integrate knowledge from different specialties, *Information handling* with key words: uniform concepts, an integrated computer supported tool, distributed problem solving and accessibility, *Process handling* with key words: process tracking, -measurement, -control and -support, *Project management, Life cycle thinking* from the early beginning of the process, *Computer supported tools* for information handling, managing the project and the process, collaboration and design, *Customer focus and requirements engineering.* CE contributes to ideas on integrated and parallel work.

The third is my licentiate thesis [5] dealing with the contract between the customer/user and the supplier/constructor. It is noticed from some case studies that functional requirements, i.e. direct support for business processes, often are well treated whereas non-functional requirements, like performance and reusability, often are missing or treated in a slipshod way. This third part contributes to the understanding of the importance of completeness of the requirements specification.

Conclusions

The perspective in the requirements process should be expanded by integrating a wider range of phases in the development process, by widening the scope of analysed artefacts and by widening the circle of participating stakeholders. Based on this the requirements process should be carried through in an integrated way, meaning that relevant information of different categories should be taken into consideration in parallel throughout the requirements process. This can be realised with a computer based support tool that constitutes a workspace, in which the appropriate stakeholders can co-operate throughout the process. This approach may contribute to some common goals in information systems development.

References

[1] Bergman L, Öhlund S-E. *Integrated Systems Development - Lessons from Industry applied on Systems Development*. SISU Report nr 7. Effective IT. 1994. (In Swedish)

[2 Davis A M. *Software Requirements.* Prentice Hall. 1990.

[3] Barki H, Hartwick J. *Rethinking the Concept of User Involvement.* MIS Quarterly. Mars 1989.)

[4] Hällström M. *Business Process Reengineering.* Triad-report 24/94. SISU. 1994.

[5] Hökenhammar P. *Ordering Competence in Information Systems development.* Licentiat Thesis. Stockholm University. 1996. (In Swedish)

[6] Ibanez M, Rempp H. *European User Survey Analysis.* Report USV-EUR, version 2.1. 30 januari 1996.

[7] *IEEE Standard Glossary of Software Engineering Terminology.* IEEE Standard 610.12-1990. Software Engineering Standards Committee of the IEEE Computer Society. 1990.

[8] *IEEE Recommended Practice for Software Requirements Specifications.* IEEE Standard 830-1998. Software Engineering Standards Committee of the IEEE Computer Society. 1998.

[9] Karlsson J. *Towards a Strategy for Software Requirements Selection.* Licentiat thesis. Linköping University. No 513. 1995.

[10] Schmidt K. *Computer Support for Cooperative Work in Advanced Manufacturing.*

International Journal of Human Factors in Manufactu7, vol 1, no 4. 1991.

[11] Sommerville I. *Software Engineering.* Addison-Wesley. 1995.

[12] *Chaos, a Study of IT-projects in US.* Standish Group Int, 586 Old Kings Highway, Dennis, MA 02638. 1994.

[13] Winner R I et. al. *The Role of Concurrent Engineering in Weapons Systems Acquisition.* Report R-338. Institute for Defence Analyses. 1988.

[14] Womack J P, Jones D T, Roos D. *The Machine That Changed The World: the story of lean production.* Harper Perennial, New York. 1990.

[15] Hökenhammar P. *An Integrated Procurement Process In Systems Development - On Requirements Engineering in a Wider Perspective.* PhD Thesis. Stockholm University. 2001. (In Swedish).

[16] Bubenko J. *Extending the scope of information Modelling.* Stockholm Universitet, Royal Institute of Technology and SISU. Report Nr. DSV 93-034. 1993.

[17] August J H. *Joint Application Design: The Group Session Approach to System Design.* Yourdon Press, Englewood Cliffs. N. J. 1991.

[18] Martin J. *Rapid Application Development.* Macmillan publishing Company. 1991.

[19] Schuler D, Namioka A. *Participatory Design: Principles and Practices.* Lawrence Erlbaum Associates, Hillsdale, NJ. 1993.

[20] McClure C. *CASE is Software Automation.* Prentice Hall. 1989.

[21] *Quality Systems & Software Ltd.* Magdalen Centre. Oxford Science Park, Oxford, United Kingdom.

[22] Marconi Systems Technology. Virginia. USA.

[23] Naumann J D, Jenkins A M. *Prototyping: The New Paradigm for Systems Development.* MIIS Quartely 6. 1982.

[24] Gotel O C Z. *Contribution Structures for Requirements Traceability.* PHD Thesis. University of London. 1995.

[25] Rolland C. *Modeling the evolution of artifacts.* Proceedings of the First International Conference on Requirements Engineering. 1994.

[26] Kotonya G Sommerville I. *Requirements engineering with viewpoints.* Software Engineering Journal. 1996.