

A Project Management Model Based on Shared Understanding

Ulf Cederling, Roland Ekinge, Bengt Lennartsson, Lars Taxén and Tommy Wedlund

Linköping University Post Print



N.B.: When citing this work, cite the original article.

Original Publication:

Ulf Cederling, Roland Ekinge, Bengt Lennartsson, Lars Taxén and Tommy Wedlund, A Project Management Model Based on Shared Understanding, 2000, Proceedings of the 33rd Annual Hawaii International Conference on System Sciences.

<http://dx.doi.org/10.1109/HICSS.2000.926940>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-111360>

A Project Management Model Based on Shared Understanding

Ulf Cederling, Växjö University. E-mail: Ulf.Cederling@masda.vxu.se

Roland Ekinge, Whirlpool Sweden AB. E-mail: Roland_B_Ekinge@email.whirlpool.com

Bengt Lennartsson, Linköpings Universitet. E-mail: benle@itn.liu.se

Lars Taxén, Ericsson Utvecklings AB. E-mail: Lars.Taxen@uab.ericsson.se

Tommy Wedlund, Linköpings Universitet. E-mail: tomwe@ida.liu.se

Abstract

Traditionally in industrial system development, the total project is decomposed into phases. The result from one phase, normally a document or a system component, is passed to the phase(s) to follow. The deliverables from the "phases" are often prescribed in standards or corporate guidelines. This metaphor, where understanding is packaged into documents, has been a cornerstone for our educational systems as well as for organizing engineering or social development projects. It is assumed that the understanding once achieved by the author(s) of the document will be transferred to its reader(s).

In three longitudinal case studies of industrial development projects, a new view has evolved. The analysis team develops a capability to answer questions occurring on the fly, rather than writing down answers to initially stated issues. Our aim is to define a model based upon developing and making available shared understanding. The paper presents a survey of the case studies. In a fourth project an embryo of the new system development model is being applied and evaluated.

1. Introduction

In Sweden there is a long tradition of developing complex technical systems. A number of companies have been remarkably successful in the development of systems in telecommunication, command and control systems, industrial process control, and the white-goods sector. This is the background for a set of longitudinal studies where we are aiming at identifying the major success factors. During the course of these case studies, we have noticed a common factor: in addition to the concepts and processes explicitly defined in the system development models used, the development of a shared understanding in the early phases has a key role. This aspect is so important that we are aiming at defining a new system development model more flexible and capable of adaptation on the fly to technological achievements and changes on the global arena in general. In this paper we will first present the four case studies. The first is on ship-borne command and control systems at CelsiusTech Systems by Ulf Cederling. The second focuses on the evolution of a company-adapted system development

model at ASEA Brown Boveri, ABB, by Tommy Wedlund. The third is on the steps towards an incremental development process within Ericsson, by Lars Taxén, and the fourth is on the introduction of a new management strategy at Whirlpool Microwave Oven Business Unit, by Roland Ekinge, and recently also by Tommy Wedlund. All four case studies are part of a coordinated effort conducted by Bengt Lennartsson. The main funder, besides the four companies involved, is NUTEK, the Swedish National Board for Industrial and Technical Development.

2. Arena CelsiusTech Systems AB: A product line management experience

The first arena we studied was CelsiusTech Systems, and there we got our first indicators of the need for a new and more flexible system development model.

2.1. Ship System 2000

From an early vision of reusing software at the module level, a more mature view of software reuse has evolved, where the building blocks are organized in patterns and are assets of a generic product line architecture. CelsiusTech Systems' product family Ship System 2000 [1,2] for the development of distributed embedded command and control systems is a good example of this.

When the project started at the end of 1985 the technical strategy was to establish an architectural platform that could profit from the rapidly emerging technologies concerning design of distributed systems. Since the specific customer products assembled from the platform varied from customer to customer, it was also necessary to build scalability into the design. Furthermore, it was of vital importance to choose a system structure that would allow commercial off-the-shelf (COTS) components to be integrated into the systems.

Previous Command, Control, and Communication project experiences at CelsiusTech Systems have shown difficulties to predict development cost and schedule due to changing requirements and time-consuming integration of the systems. The approach taken in Ship System 2000 (SS2000) aimed at tackling these problems and increase

productivity and shorten time to market for their system products.

The product line approach has shown to fulfil established goals. Nearly 60 system products have been built embracing similar technology and sharing a common layered architecture and CelsiusTech Systems has considerably shortened time to market for their products. Cost and schedule can now be predicted, based on known requirements for product line configurations, and system products from this application domain with predefined performance behavior can be delivered.

2.2. Process model

The process model advocated in SS2000 was document-driven. The structure of the model was similar to the standard DoD-STD-2167A [3]. However, the product line engineering approach taken evolved the process model to a dual more reuse-driven process consisting of two sub-processes for domain engineering and application engineering respectively.

The domain engineering sub-process supported the application engineering one with adaptable components. In the latter sub-process, actual customer requirements were analyzed and building blocks integrated into system products according to existing customer requirements.

The process model used also became more iterative and release-oriented in time to the growth of the system family. This approach is particularly obvious in a subsequent project, the development of an Air Defence System, partly based on the architectural platform of SS2000.

Process models similar to DoD-STD-2167A require establishment of a very extensive set of documents. Even if CelsiusTech modified the structure of the process model, the set of documents in SS2000 covering project management, development and integration, system functions, system family, and system products became very large.

Documents also tend to be obsolete after some time. Even if CelsiusTech Systems strived for keeping the documents in SS2000 updated, the team members had a problem to get sufficient understanding of the system family through the project documents. Some of the documents were too thick and impenetrable, others were inconsistent. The documentation for different customer projects gave for example different views of the product structure in spite of the system products contained the same set of system functions. Furthermore, design knowledge is implicit and contextual.

One of the main challenges in projects as large as Ship System 2000 (SS2000), at peak time it involved up to 250 people, is to make the project members understand what they are going to develop. According to one of the chief architects in SS2000: *"In a very real sense, the most valuable information resource asset of a project is the knowledge accumulated by the designers and developers."*

Therefore, any strategy must ensure that the process of building that knowledge and experience is as efficient as possible. It must also ensure that this knowledge is stored in such a way that it is not irretrievably lost if a key person should disappear. The process of knowledge building related to system development has not been taken into consideration enough in our view. The documentation is of course part of that process as well as the growing system/system family itself.

2.3. The architecture as carrier of shared understanding

The decision to develop a family of systems instead of separate products, in spite of having only two customers at the start of the project, was a key strategy in SS2000.

CelsiusTech Systems was firmly resolved to achieve as much commonality as possible in the system products based on SS2000 and they used domain analysis to identify commonalities and variations and conceive domain-specific reuse.

The tactics was to reuse existing components in the first place. If that was not possible, the second option was modification of existing components followed by purchase of commercial-off-the-shelf components. Last of all, new development of non-existing components was chosen. However, the highest priority went to the fact that reusable components did not contain any information that could be used to identify a particular customer or customer system product.

From the domain analysis, a generic architecture was developed. We have seen many definitions of the concept software architecture [4]. The following one was established in a discussion group at the Software Engineering Institute in 1994 [5]: *The structure of the components of a program/system, their inter-relationships, and principles and guidelines governing their design and evolution over time.*

A software architecture is communicative. It shows the allocation of requirements into software components and their dependencies. During the efforts to establish the architecture, knowledge and understanding of the requirements increase and the customers as well as the system developers get a more thorough and comprehensive view of the future system.

The task to establish the architecture in SS2000 was given to a core design team. They defined a layered architecture and identified nearly 50% of the existing set of around 250 components rather early in the process.

Conceptually, the components were an inheritance from earlier software projects at CelsiusTech Systems and the experience from those projects coupled to extensive domain knowledge at the chief architects and very competent customers from the beginning, were the main reasons behind this rather rapidly established architectural vision of the system family.

There were rather few people working with the establishment of the architecture. It was therefore relatively easy to reach an architectural understanding among those architects. After having defined the first version of the architecture, the architecture team was enhanced with people from organizational units responsible for the four main functional areas C³, MMI, Weapons, and Fundamentals. The reason was a desire to hold the architecture together and avoid a situation where a separate architecture was developed for each functional area. Another effect was that knowledge about the architecture was transferred out into the organizational units.

The main task however was to carry the architectural vision to the large quantity of incoming project members. This was done using unofficial project documents and an extensive training program containing design exercises as well as real problems derived from the SS2000 project.

Through moving the focus from development of code to integration of large building blocks which are part of a stable architecture, CelsiusTech Systems managed to fulfill their goals. The architectural infrastructure has survived over time supporting instantiations of system products of various sizes across the product line with a degree of reuse between 65-85%.

2.4. Summary from the CelsiusTech Systems arena

The software paradigm built on product line engineering and systematic reuse offers a new approach of handling complexity. The approach is very cost-effective and will shorten time to market for the developed products compared to traditional software development approaches.

The establishment of an architectural platform in the SS2000 project was a key mechanism for achieving a shared view of the system products to be developed. An architecture offers a support infrastructure from which common issues for the product family like system product capabilities (functional as well as quality), infrastructure investments, support processes etc., can be considered. It acts as a communication tool and contributes to an early and better understanding of the future systems.

The shared architectural vision among the team members of SS2000 corresponds to observations made by Dikel *et. al.* [6], who emphasize six principles to be critical for extensive success of an architectural platform as a base for product line engineering. Among these principles are the need of establishing and broadening relations with stakeholders, and the necessity of maintaining a shared architectural vision across the company.

Experiences from CelsiusTech show that the number of people involved in the initial stages of a large project should be limited. Once the components and the interfaces between these components have been specified and frozen, more people can be supplied to the project.

The architectural framework has turned out to be stable and survived external and internal disturbances. The company also managed to transfer the architectural vision created by the architects to the team members who got a very consistent view of the building blocks, their functionality and dependencies. This shared architectural vision among the team members was very essential for the technical success of SS2000.

3. Arena ABB: A company-adapted system development model

The second arena we studied was ABB Information System. We looked at the use of systems and tools in the development of information systems [7,8,9,10]. The foundation for the study was the statement that: *Someone develops something with the aid of some tool*, where *someone* refers to a system developer, who carries out a system development project, with the aim of developing an information system. The information system corresponds to "*something*" and a system development model is required, *i.e.* a "*tool*" in order to develop the information system.

This report describes empirical results from an administrative system development project within ABB, where the aim has been to describe what a company-adapted system development model ought to contain. The content is based on the requirements of the system development project, and this approach constitutes a new wider perspective of what a system development model should contain. It takes into consideration knowledge, training, model, method, communication, and decisions; as carriers of shared understanding.

3.1. The study

Twelve men and six women were interviewed during the fall 1998 within ABB. All the project leaders and the majority of consultants in five sub-projects were interviewed. There were several new employees, persons with a long period of employment within the company, and both Swedish and foreign consultants, among those interviewed, all with different project positions and with different work content from four locations in Sweden. The aim in choosing the interviewees was to obtain a representative selection of interviewees from different phases and positions in the system development project. The project leaders have selected the majority of the persons interviewed.

The interviewees were given the interview document one week before the interview and subsequently had it with them on the occasion of the interview. Two persons were interviewed by telephone and received the questionnaire via e-mail. Some other interviewees have also been sent the questionnaire via e-mail. The interviews were usually about 45 minutes long, sometimes longer. The

questions in the interview were determined in consultation with one of the project managers within ABB, and the majority of the questions were open qualitative questions of the “what” and “how” type.

3.2. Results

The results of the interview study were presented in a final report to ABB during autumn 1998, and they are summarized below.

3.2.1. Knowledge. About 1/3 of the project participants had not been involved in such major development work during the preceding five years. It was the interviewees from ABB who lacked such previous experience. On the other hand, several had worked with the development of this type of administrative information system in main-frame environments. Many in the group felt that some group members lacked competence in component-based system development.

The majority of the external consultants had previous experience of development work in three-layer solutions in Client/Server environments and experience of the development tools used in the system development project. None of the consultants had previously worked with Business Objects; see Sims [11]. The interviewees believed that it was the consultants who were responsible for the knowledge on how the system development project should be carried out.

3.2.2. Training. It was evident from the interviews that many of the consultants also acted as trainers. They often produced models for the other system developers and for themselves concentrated on carrying out the most complicated tasks. Several of the project participants were complete beginners in Client/Server environments at the start. Therefore they often underwent some form of introductory training in connection with the respective phase start of the system development project.

3.2.3. Model. The interviews revealed that the work operations were sometimes underestimated, particularly in the beginning and at the end. Some project leaders said therefore that it was important to describe these work operations in the system development model particularly.

3.2.4. Method. Several interviewees thought that documents in the various sub-projects did not contain the same information and that there were differences in how detailed the descriptions were. A common description was therefore required for what the minimum content of documents should be. The documents, which act as bridges between different phases, should be described in more detail.

Tools were required in order to produce the documents during all the phases of the development work. The interview results included improvement proposals for the

specification phase and the production phase. Many of the interviewees were also concerned about the maintenance phase.

3.2.5. Communication. Several interviewees believed that the common work could be affected negatively if the members of the project group were in different geographical locations. Many of those interviewed also felt that there was poor exchange of experiences between the various project groups, it sometimes felt like “them and us”.

The interviews indicated that above all the foreign consultants were used to a greater element of common work and a stricter control of objectives and measurement of the work time. About half of the work time consisted of individual work for the majority of the interviewees. During the rest of the work time they worked with colleagues or had joint meetings. The allocation of time between “individual work” and “common work” was often related to work tasks and the role in the system development project. The majority of the interviewees were satisfied with their work assignments. Nearly all respondents cited their colleagues and the actual work assignments as the most important work-related reasons that they were satisfied with their job

3.2.6. Decisions. Approximately 2/3 of those interviewed felt that their work assignments were unstructured. The majority thought that the work assignments were not routine character, and that the tasks often required a high level of concentration. Some of the interviewees lacked guidelines and considered that the prerequisites for the work results were not known beforehand. Sometimes it was impossible to get hold of the person who needed to be contacted in the project group, which many of those interviewed found frustrating.

Several of those interviewed reported that they relied too much on communications via e-mail. This was not experienced as positive and indeed many felt that it was negative that the work was carried out at different locations.

3.3. Summary from the ABB arena

It is important that the system development work is carried out from an overall perspective, where knowledge, training, model, communication, and decisions, are all carriers of shared understanding. This overall perspective also facilitates communication within a development project, and it can be carried out by working in a new way with models and methods in this broader view of the development work, where knowledge, training, communication, and decisions can be integrated with models and methods. The empirical results in this system development project indicate that there is a need for this overall perspective for what a system development model should contain [9]:

- Links between different phases of the development work, where the links constitute a transfer of knowledge to different work categories.
- A common concept that permeates the entire development work, which means that communication in the analytical and investigative work, is facilitated.

The links facilitate the transfer of knowledge and establish the need for training among different work categories in a development project. The links will facilitate understanding for what must be carried out. It is important that the work operations are easy to understand for all work categories. The links must transfer of knowledge between different work categories. There is also a need for a common concept, in order to facilitate communication and decision making within a system development project. The company language can then facilitate communication between the various people in the analytical and planning project

4. Arena Ericsson Utvecklings AB: An incremental development strategy

Our third arena is Ericsson Utvecklings AB. The telecommunication market is changing very rapidly, mainly due to two forces: the deregulation with the entering of many new operators leading to more competition, and the proliferation of new technology, for example mobile communications, intelligent networks, Internet etc. This has put a demand on the suppliers to be more reactive and flexible to the market needs, which means shorter lead-times and more flexibility in handling late and changing requirements. It is a very challenging task to achieve this considering the size, complexity, and in service performance (up-time) requirements of the telecommunication systems.

The main part of the Ericsson development work concerns software for controlling and supervising the telecommunication traffic. The traditional way of developing the software, the so called waterfall development model, is not apt to meet the challenges today. Therefore, the way the software is developed is currently being changed to an incremental development model. The transition means that a shared understanding about incremental development must evolve from the previous, traditional knowledge. This is complicated by the fact that Ericsson has many designers (more than 10 000) working at local design centers all over the world.

4.1. A method for achieving shared understanding

In this section we are proposing a new method for achieving shared understanding. This method is based on an action-oriented view on knowledge; see for example Polyani [12] and Molander [13]. The basic idea is to provide a group of actors, usually designers and method

developers, with concrete instruments for reflection and action, which they use in a dialectical interplay to achieve shared understanding about a certain study field or context.

The instruments for reflection are *conceptual models*, which describe the static and the dynamic aspects of the context, in this case incremental development of software. The purpose of these models is to make explicit what items are managed during incremental development, how they are related to each other, and in what order they are treated. The static aspect is modeled by for example UML (Unified Modeling Language) or OMT (Object Modeling Technique) notation, while the dynamic view is modeled by *information flow diagrams*, which belong to a particular class of process models, called entity-based models [14]. The reason why we choose these models is that they focus on how things are interrelated. Thus, they are very suitable as vehicles for achieving shared understanding about a context. The instrument for action is an *object oriented PDM-system* where these models are implemented on the fly.

By action we mean that “real” managed items are being created and managed based on the reflection models and tried out in practice. The experiences gained from this will in turn affect the models for reflection. In order to maintain the dialectical interaction between reflection and action, it must be very simple and straightforward to implement the reflection models in the system. A system which has these properties is Matrix from Matrix-One Inc., USA. This is also the system used in this work.

A shared understanding is achieved when the actors agree that the models and the implementation in the PDM-system are sufficient as a starting point for any local adaptations necessary at particular design centers or projects. This means that there will always remain a certain amount of disagreement among the actors, and it is important stop the process in time. The outcome is individual knowledge as well as shared knowledge among the actors. Furthermore, the reflection models and action support represent explicit organizational knowledge, which can be maintained and transferred to other design teams.

4.2. Transition to an incremental development model

In the traditional waterfall model, the development is done in sequential phases where a certain result is to be accomplished after each phase. For example, the requirements are analyzed and frozen very early in the development. In the incremental development model, the whole development assignment is divided up in steps (increments) which can be developed and tested as independent units, see Figure 1. This means that all the requirements may not have to be frozen at the same time. Late incoming requirements can be directed towards late increments,

and removed requirements will only affect single increments. This means greater flexibility with respect to

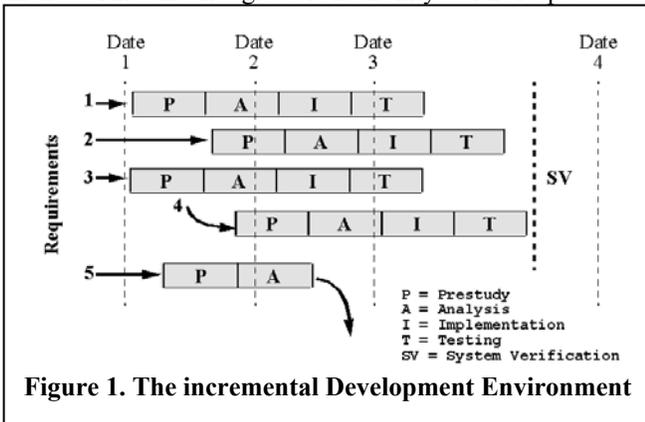


Figure 1. The incremental Development Environment

requirement changes than the waterfall model can offer.

A number of attempts to work in this way had been done at various local design centers at Ericsson. In 1996, a method team was assigned the task of consolidating these efforts. Besides the method team, a project support team also participated in this work, altogether 6-10 people. The reflection models were discussed at regular meetings, and implemented in the support system by the method team. The emerging models and support were tried out in a small number of pilot projects. The resulting static model is shown in Figure 2.

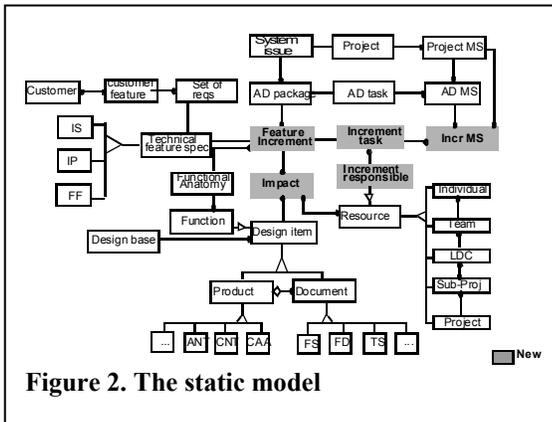


Figure 2. The static model

The implemented support system corresponds to the action of trying the reflection models out in practice. An example of this system can be seen in Figure 3. It can be noted that the same managed items are visible in both the models and the support system.

Some observations from this work were:

- The one-to-one mapping of the models to the support system gives an immediate feedback of the practical relevance of the models.
- Before the work was started there was virtually no agreement at all about the meaning of the concept of “increment”. Still, the number of model iterations was more than thirty. The work took nearly a year and the

total work effort several thousands man-hours. Thus, achieving a shared understanding is a very hard process. However, once established, it is easier to convey it to a larger group.

- The final model contains very few categories that are specific to incremental development (marked dark in Figure 3). A comment often heard about was “that’s obvious, we are already working like that”. This indicates that the traditional way of working was not made explicit, and that a large part of the process was devoted to achieve this.
- In the waterfall model, documents were the carriers of project information. In the new model, these documents have been replaced by pure information elements. This is a consequence of the support system, which makes it possible to extract information from different perspectives, and, if necessary, collect this in the traditional way as documents. This is a significant change within Ericsson.

4.3. Summary from arena Ericsson Utvecklings AB

The objectives of the proposed method is to achieve both individual and shared understanding, as well as carriers of that understanding in terms of models and support system. The first experiences from applying this method are very good. In a rather short time (a couple of years), and with a relatively small staff (less than 10 persons), a shared understanding of the incremental development and a corresponding support has been objectified within large parts of Ericsson. The same method has been used in the transition from document based to a computer based engineering change order process, and the results are equally positive in terms of an inter-subjective, shared understanding among configuration managers, project leaders and support staff.

Object	State	Detail	Owner	Type
AD Package AD 1 -	CR FREEZE	Program production for Configuration and Network Synchronization Functions	usdbtb	AD Package
Integration Increment Network Spec -	Identified	Network synchronization functions	usdbtb	Integration Increment
Integration Increment Configuration -	New	Network synchronization functions	usdbtb	Integration Increment
Function ENH -	New		usdbtb	Function
CNT 213 1157 R1	PREL		creator	CNT
CAA 107 6828 R1A	PREL		creator	CAA
Structure Specification 131 61 131 61-CNT 213 1157 A	PREL	STRUCTURE SPEC.	creator	Structure Specification 131
Description 1551 1551-CNT 213 1157 A	PREL	DESCRIPTION	creator	Description 1551
Product Revision Information 109 21 109 21-CNT 213 1157 A	PREL	PRODUCT REV. INFO	creator	Product Revision Informat
Document Survey 1095 1095-CNT 213 1157 R1	PREL	SSBOARD	creator	Document Survey 1095
Description 1551 11551-CNT 213 1157 A	PREL	FLOW CHART BLOCK SSBOARD	creator	Description 1551
CNT 213 1094 R2	PREL		creator	CNT
CNT 213 1161 R1	PREL		creator	CNT
Function NT config -	New		usdbtb	Function
Detailed Req 1230 02 PA1	AGREED	Desor...	creator	Detailed Req
Detailed Req 1230 PA1	AGREED	Desor...	creator	Detailed Req
Input Req MR 160 PC1	UNCLEAR		creator	Input Req
Req Issuer RMOA	New	BR, American market	usdbtb	Req Issuer
Req Issuer PNI	New	BR, Public Network	usdbtb	Req Issuer
Baseline Requirement PA1	APPR	Business all MRS requirements	usdbtb	Baseline
Req Issuer RMOJ	New	BR, Japanese market	usdbtb	Req Issuer
Req Issuer RMOO	New	BR, OEM market	usdbtb	Req Issuer

Figure 3. Example from the support system showing traceability between objects

5. The Whirlpool arena: Understanding without documents

The fourth arena we studied was the Whirlpool Microwave Oven Business Unit. It was recognized that several product development projects were plagued with difficulties, causing delays and frustration in the Project teams. It was decided to install a Corrective Team to study what could be improved in the product creation process. The team came back with specific suggestion for improvement. Novel elements in project management were defined and implemented in a specific product project. All initiatives were based on the principle “Understanding without documents”.

5.1. The C2C process

C2C is the acronym for the Stage Gate Process Whirlpool used in all New Product Development Projects. The process has four main phases: Ideation, Conceptualization, Conversion, and Execution. Ideation is the phase, as the name indicates, where ideas for the new product are created. The Conceptualization phase is characterized by the creation of several competing concepts, which are evaluated against each other. At the Concept selection milestone, one concept is created and described in all possible details. The Concept Evaluation Tollgate is the critical evaluation of the selected concept. During Conversion, all uncertainties associated with the concept are straightened out, which means that this phase is a “uncertainty reduction phase”. When the project enters the Execution phase, the likely hood to realize the concept is relatively high and all detailed designs are made and manufacturing tooling are prepared.

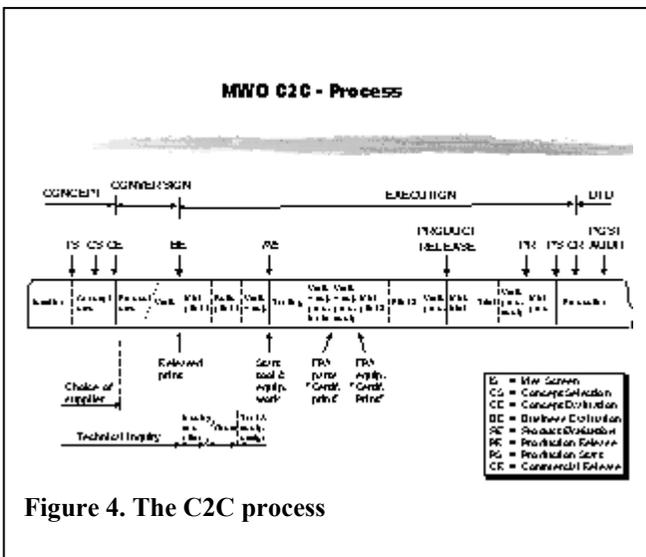


Figure 4. The C2C process

5.2. The C2C in action team

The small “Corrective Action” team which was installed to assess the present situation was called “the C2C in Action team” and its charter was to define the basic underlying problems, *i.e.* making a Root Cause analysis of the Process and the Product Creation System

The C2C in action Team followed a very simple and straightforward process:

1. Identify symptoms
2. Clustering the symptoms
3. Formulating Problems, which could be associated with these symptoms
4. Cluster the problems
5. Define the key problems in these areas

The Symptoms were collected in a 2-hour session where the Corrective Team and one of the Business Teams got together. The symptoms were analyzed and edited to remove duplications. This list of Symptoms was then give to the members of the Business Team and they were asked, individually, to formulate five Problem Statements related to these symptoms. Each statement should start with “The Problem is.....”.

The BT members came up with a list of 37 problem statements. These statements were analyzed and clustered

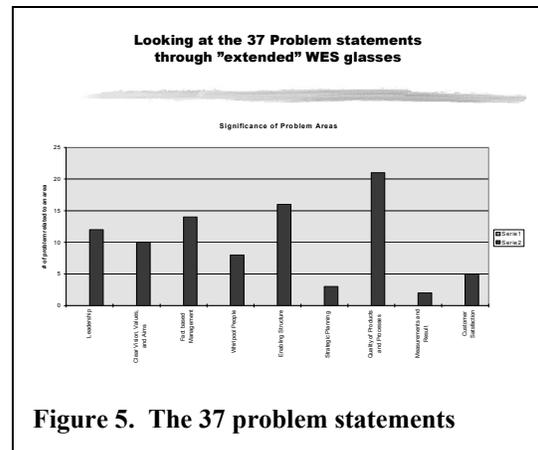


Figure 5. The 37 problem statements

by the Corrective Team. All in all 10 clusters were defined, and each of them was defined as an element of the New System for Product Creation. They are shown in Figure 5.

The corrective actions could start only after a prioritization of the problem statements. Confronting an ongoing project, a new microwave oven named Talent Compact did the prioritization. This confrontation led to the formulation of five problems, which were believed, could be addressed and tested immediately in the Talent Compact Project making this project "best in class".

Five problems were:

1. The Vision for the new product only partly common

and shared

2. The charter for the team incompletely stated and detailed targets not harmonized and agreed upon, making the requirement specification a moving target for the team
3. Very often the product cost does not meet the target when the project comes closer to production start, and in most cases the supplier parts come in at too high cost.
4. Resource allocation is normally difficult and frustrating, due to scattered priorities, and incomplete understanding of the status of a specific projects at a given point amongst the individuals who manage the resources
5. In the Execution phase, where all the detailed design work takes place, the designer often finds it difficult to always consider that a modification he does on his design, can affect other designers.

5.3. A series of initiatives were implemented to cure the problems

Five mechanisms were developed to address these problems:

- Strategic Intent – to sharpen the Vision and make sure the Vision is shared and the Charter is clear.
- Socratic Management by Understanding – to make sure basic assumptions about the new product and the market are challenged and that the Product Specification is without “holes” and ambiguities, agreed upon, and understood by all parties.
- Target Costing – to safeguard creative dialogue between the team and the supply base
- Planet organization – to make sure resource issues are addressed without delay and maximum experience is injected into the project
- Weekly build – to always have a latest prototype for anyone to look at

5.3.1. Strategic intent – Value driven specification. The microwave oven business is highly competitive and dominated by Asian companies. Price erosion with 2-3 % every year makes the business primarily price driven. This situation is not unique for microwave ovens but characterizes the entire White-goods business.

Efforts to move the products out of the commodity area, is spent by all players and it was recognized that the new oven must “stand out” on the shop floor, create value to the consumer, and at the same time have a competitive going price. Three strategic goals were formulated by senior management and given to the team early in the Conceptualization phase:

- to reduce final assembly time by 50%

- a product that stands out on the shelves in the outlets and creates unique value to the user
- a product that meets unmatched cost targets

The team used the DFA concept, and a Company Internal consultant, who trained and coached the use of the tool, enforced the use. The end result was a 35% reduction in assembly time. The second goal was addressed in a one-day creative session, with participation from about 25 individuals, managers and team members. The result was the invention of a unique, for the consumer highly visible but at the same time highly useful product feature.

By being crisp in formulating the strategic intent of this new product, Talent Compact, the entire organization was focused on this strategic direction:

To create a product which “stick out” on the sales floor and at the same time can be assembled in 50 % of the time needed for its predecessor.

All involved people could clearly understand what was important in this project.

5.3.2. Socratic management by understanding. Popper is very precise and correct when he says that no one can know anything about the future for sure. We are all dealing with Assumptions and Hypothesis about the future. This is also true in product development. The best way to increase knowledge is to formulate theories and make assumptions and the critique them in a constructive way. Again according to Popper. This is the philosophical background for the two initiatives Whirlpool introduced a few years back. The Red Team is a small group of senior managers, no more than three, with long and recognized experience, normally in marketing, in the technical and financial field. The Red Team conducts in-depth reviews with the project team, a review which is based on questions which challenge the assumptions and hypotheses the team has made for the project, and the Principles which will be followed. These review meetings are in most cases very enlightening and causes the Project Team to sharpen and verbalize its assumptions.

The Technical Design Reviews has the same background and structure, and were conducted by senior technical managers. All these reviews were conducted in the Conceptualization phase, and forced the team to go back to the drawing board and on many points reshape its assumptions.

Socrates claimed he could teach anyone mathematics just by asking questions.

5.3.3. Target costing. The third initiative was introduced in the Conversion phase – Target costing. The idea is to give to the supplier of a specific part a max allowed cost for that specific part. In most cases the given price is so aggressive that the supplier has to come back with suggestions to the designer on how the part could be modified in its design to allow the supplier to meet the target cost. This methodology works when there is a lot of ex-

perience within the company about product design, as in the case of microwave ovens, which have been developed and produced by Whirlpool since 1962.

In the Talent Compact project it was decided to enforce this initiative strongly, and for many suppliers the methodology was unusual but created the desired result. All suppliers in the supply base agreed to follow this methodology.

The target costing method forces the supplier to understand, by asking questions, what detail he is supposed to produce, before any drawings are available. It is in his interest to understand early since he has committed to the price.

5.3.4. The planet project organization. The fourth initiative was to create an organization, which effectively linked the project work with the rest of the development organization, which did work on other development projects. A PLANET organization was formed and it was introduced from the beginning of the Execution phase.

The details of the picture are:

- The Project Leader is the “SUN”
- The “PLANETS” closest to the sun are The Functional group leaders who at the same time act as Module Leaders
- The second level “Planets” are the designers working on the specific Product Module.

The expected advantages with this organization has been confirmed and are:

- Close connectivity between the team and the rest of the organization
- Resource issues are quickly resolved through the double role of the Functional Leaders
- The most experienced individuals in the organization are involved in the project and they are accessible to the team members, and transfer of experience to new team members comes naturally.

5.3.5. Weekly build. The fifth, and last, initiative was also introduced from start of the Execution phase.

The Talent Compact is a “local project” which means that all technical people engaged in the project are co-located. The number of individuals engaged was in the order of 25 at a given time and they were working in a building with three stories. The Organizational Functions were spread between the three stories of the building.

The C2C in action team found that many projects share the same problems of “consequence blindness”, which means that individual designer has difficulties always to keep all other designers updated of his design modifications.

And the solution to this problem was to introduce “the weekly build concept”, which is a modified version of Microsoft’s “daily build”. One technician was given the responsibility to keep one prototype

always updated with the latest version of the designs. Once every week he had his prototype available for everyone. It was located in a specific room, where any body could go and look for himself, and check if his colleague designer had made some changes which affected his design.

Not a single document was needed to describe what has changed. It was obvious to every designer looking at the prototype.

5.4. Summary from the Whirlpool arena

The Talent Compact has so far been the best product development project in the Whirlpools microwave oven business unit. The product is not launched into the market place yet, but is planned for introduction in December:

- all consumer tests indicate a winning concept
- the estimated cost price is still, five months before market launch, within the target

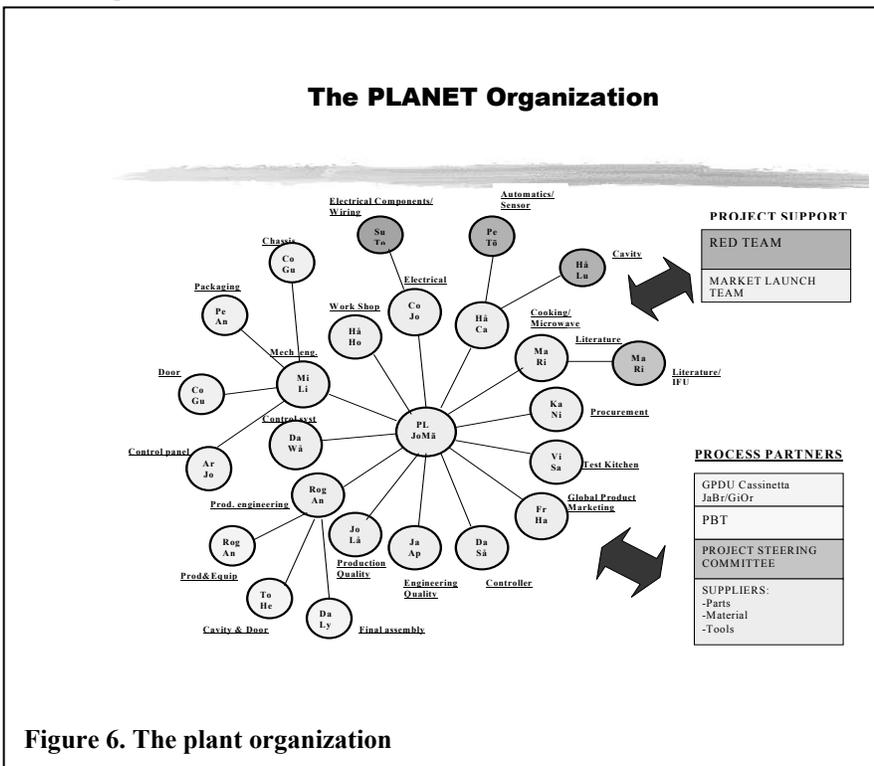


Figure 6. The plant organization

The Strong linkage between the functional organization and the system structure of the product is the key characteristic of this organization.

- the specification is, and has been since the Concept Tollgate, “dead stable”
 - the project follows the plan precisely
 - the project has used less resources than planned
- This indicates that these initiatives are powerful and helps understanding.

6. Conclusions

The project management model at CelsiusTech Systems was mainly aiming at enabling reuse of understanding and of system components from one system generation to the other. *The system architecture* was the carrier of the shared understanding. It was observed that the requirement specification and system specification documents did not provide the answers to the questions arising in later phases. However, the process of developing the documents had sharpened the shared understanding in the initial team.

The changing requirement and rapidly moving technology within Ericsson illustrates why it is not possible to separate specification from project execution in the IT area. The development team must have the capability to react on the fly on customer expectations, competitor moves, and advances in technology [15]. The system development model can no longer be based on the two components: documents describing what to do and what has been done, and the system being developed. A third cornerstone, equally important, is the shared understanding of what to do, why and how. The team is developing its understanding by building the system. The gradually evolving system and the accumulated understanding are interacting. One can't occur without the other. In this process, the development model and its associated support system become the carriers of shared understanding. As the understanding gradually matures, part of it can be packaged in documents and made available outside the team. However, once written down it is made static and thus vulnerable to changes. Our approach is to focus on the team's capability to answer questions and to keep the team and its capability available during the project.

Within Microsoft, the daily build strategy has been used to coordinate and synchronize the work done in autonomous teams [16]. This is natural when developing software. At Whirlpool the same strategy has been practiced when developing also Microwave ovens. Not only the software in it but also for all mechanical and electrical parts! The contributions were put together weekly rather than daily, however, and the prototype is one of the carriers of shared understanding.

The observations on the other arenas are supporting the general conclusions. In the future we will elaborate the system development and the project management models based on experiences mainly from the Ericsson and Whirlpool arenas.

References:

- [1] Bass L., Clements P., and Kazman R., *Software Architecture in Practice*, Addison-Wesley, 1998
- [2] Cederling U., "Industrial Software Development - A Case Study", Linköping Studies in Science and Technology, Thesis No. 348, 1992.
- [3] DoD-STD-2167A, Defence System Software Development, 1988.
- [4] Jesper Andersson: Towards Reactive Software Architectures. Linköping Studies in Science and Technology. Lic. thesis No 769. May 1999. ISBN 91-7219-501-0.
- [5] Garlan D., "Research Directions in Software Architecture", ACM Computing Surveys, Vol. 27, No. 2 June 1995.
- [6] Dikel D., Kane D., Ornburn S., Loftus W., and Wilson J., "Applying Software Product-Line Architecture", IEEE Computer, August 1997.
- [7] xxxx
- [8] Nilsson A.G, Utveckling av metoder för systemarbete - ett historiskt perspektiv, presenterat vid VITS Höstseminarium den 23-24/11 1994, LiTH-IDA-R-95-13, IDA, Linköpings universitet [The development of methods for system work - a historical perspective, presented at the VITS Autumn seminar on 23-14/11 1994, LiTH-IDA-R-95-13, IDA, University of Linköping] ,
- [9] Wedlund T., Att skapa en företagsanpassad systemutvecklingsmodell - genom rekonstruktion, värdering och vidareutveckling i T50-bolag inom ABB, Licentiatavhandling FiF a 6, IDA, Linköpings universitet (Creating a company-adapted system development model - by means of reconstruction, evaluation and further development in T50 companies within ABB). Doctoral thesis FiF a 6, IDA, University of Linköping 1997.
- [10] Österle H., Business in the Information Age - Heading for New Processes, Springer Verlag, Berlin 1995
- [11] Sims O., Business objects, delivering co-operative objects for Client/Server, McGraw-Hill, Cambridge 1994.
- [12] Polyani, M. (1966): *The Tacit Dimension*, Reprint by Peter Smith, Gloucester, Mass. 1983
- [13] Molander, Bengt (1996): *Kunskap i Handling*, (in Swedish), Daidalos, Göteborg.
- [14] Humphrey, Watts S. and Kellner, Marc I. (1989): *Software Process Modeling: Principles of Entity Process Models*. SEI-89-TR-002.
- [15] Kristina Davidson, Bengt Lennartsson: Team Learning Capability - A Strategy to Master Complexity and to Achieve Flexibility. Proceedings of the 7th World Conference on Continuing Engineering Education, Torino, Italy, May 1998.
- [16] Micael A. Cusumano, Richard W. Selby: Microsoft Secrets - How the World's most powerful software company creates technology, shapes markets, and manages people. The Free Press, ISBN 0-02-874048-3.