8th Conference on Systems Engineering Research
March 17-19, 2010, Hoboken, NJ

Paper # 1569269957

Change Impact Analysis – A Case Study

Jorn Breivoll, Buskerud University College
Gerrit Muller, Buskerud University College, Mike Pennotti, Stevens Institute of Technology

1. Abstract

The objective of this study has been to investigate some effects of changes to products and processes at Kongsberg Maritime (KM), due to platform shift. The company's "New RIO Hardware Line" project (NRIO) was used as a case study. The purpose was to categorize changes, examine how they affected their surroundings, and to explore whether potential for improvements exist in KM's change handling.

One specific outcome of the study was a method which we named CEA (Change Effect Analysis); an FMEA-like matrix tool for documenting and predicting impact of changes.

Although changes in general do not necessarily propagate widely, major platform shifts may still impact the involved products considerably. The ultimate goals are to avert avoidable changes, capture change effects in early stages, and to render possible prediction of future effects from past time experiences.

2. Introduction

Kongsberg Maritime is a long-established Norwegian company within maritime technologies. It has a world-wide leading position and a rich history of product development in the areas of process control (PC) and dynamic positioning (DP) systems for ships and installations at sea.

We analyzed changes over a period of 3 years in one specific platform project to understand the impact of changes. Throughout time, a steady stream of changes has occurred in the company's products and product development. In 2005 one specific project was initiated, called "New RIO Hardware Line" (NRIO). The objective of the project was to develop a new cost-effective hardware portfolio for the next generation stand-alone DP systems. This was realized through a new synchronous I/O BUS, a new redundancy concept, and a number of new hardware components for processing I/O. The NRIO solution represented a technological changeover, later used as next generation product platform in several KM departments. It turned out to be of major influence to the company, initiating a substantial variety of changes to be handled.

Even if techniques aimed at facilitating impact analysis exist, they are not often practiced[10]. This may be because they are too complicated and time consuming[7]. The methods suggested in this paper are meant to be applicable in a way that they can be implemented and used with a minimum of effort. While knowing that such methods most certainly have weaknesses, there is a conviction that "simple, imperfect, but used", is better than "complicated, perfect, but unused".

Based on the retrospective analysis we proposed and explored a change impact analysis tool. The objective of the proposed tool is to avoid cost and schedule overruns in future projects. More research is needed to evaluate such tool in other domains.

3. Basics of CIA and GD

3.1. Change Impact Analysis. Change Impact Analysis is seen as one of several change management activities. Within Systems Engineering, Change Management itself has the purpose of supporting the processing of changes[14]. One sub-activity is planning for change, which again has the sub-activity of assessing the extent of changes. This is what is called Change Impact Analysis, usually defined as the activity performed to "*identify the potential consequences of a change, or estimate what needs to be modified to accomplish a change*"[5,10]. Change Impact Analysis and Change Propagation are often considered to be two aspects of the same thing. Hence, impact analysis is much about how changes propagate and affect their surroundings.

3.2. Generic Development. Generic development is about reuse of know-how and well-established processes for the purpose of arriving at benefits like reduced cost, improved quality, reduced time to market, etc[2, 18]. The reuse will typically materialize in hardware or software repositories utilized as platforms for several products. A general definition says that a product platforms is "*a common design, formula or versatile product, based on which a family (line) of products is built over time*"[17]. Various platform definitions exist and are closely adapted to particular business specific drivers. They often include words like "commonality" and "reusability"[16].

4. Related Work

4.1. Related Work on Change Impact. Fricke et al[1] suggest five strategies for coping with changes. Those are Prevention, Front-Loading, Effectiveness, Efficiency, and Learning. The essence is that it is possible to take advantage of major alterations in complex systems by systematically addressing and preparing for the changes in early stages of a development process. As suitable tools for such activities, they propose techniques like FMEA, QFD and TRIZ, to ensure proper documentation and information flow.

Rutka et al[3] describe an advanced Change Prediction Method (CMP) based on Design Structure Matrixes (DSM), where they calculate propagation risk and likelihood from certain algorithms.

The strategy of Heindl & Biffl[4] is to establish test cases for tracing of changed elements from changed requirements, by using a Tracing Activity Model (TAM). This is applicable for re-testing of software after implementing change requests. They compare three tracing strategies with respect to effort, called No Trace Reuse (NTR), Trace Based Re-Testing (TBRT) and Ad hoc Trace Reuse (ATR).

Hassine et al[5] introduce Use Case Maps (UCM) in their approach to assessing the impact of changed requirements in software. An UCM-CIA tool is provided with several algorithms for handling dependency information.

Comparison of two approaches to product change, the Change Prediction Method (CPM) and the Contact & Channel Model (C&CM) is the contribution of Keller et al[6]. The CPM method is earlier introduced by Clarkson et al[11] as a tool for identifying effects of propagating changes by mapping direct and indirect links between components. The C&CM method is developed at the University of Karlsruhe. It is a rather abstract approach to systematic product model representations and problem solving described by Working Surface Pairs (WSP) and Channel and Support Structures (CSS). The methods utilize matrixes and node links to describe change propagation paths and risks.

Giffin[7] presents a case study of a large set of industrial change data collected over nearly a decade. She applies and extends the methods used in many of the cases to analyse her results. The outcome is a rather extensive statistics on mapping of change paths and propagation patterns.

Jarrat et al[8] address the pitfalls of engineering changes through a case study of an engine manufacturer. They proclaim that most problems connected to changes come from minor mistakes and lack of system understanding and overview. Also communication breakdown is seen as a pitfall. Like Fricke[1], they recommend tools like FMEA, QFD etc. to be better prepared for change handling.

A case study on evaluation of impact analysis (IA) is performed by Lundvall et al[9]. They present an example of how to assess the quality of change prediction by systematically compare the expected changes with the occurred ones.

Several other studies may be found, suggesting different approaches to and techniques for change handling. Many of them are identified by Giffin[7]. Nevertheless, she proposes that none of the existing models are properly suited to effectively predict change propagation in large, complex systems[7]. What is needed is a more applicable documentation method requiring little enough resources that engineers are willing to use it regularely[7].

4.2. Related Work on Generic Development. Muller[2] states that effective implementation of platforms has proven to be difficult. He mentions certain processes which are always present inside development companies; e.g. the Customer Oriented Process (COP), the Product Creation Process (PCP), and the People and Technology Management Process (PTMP). He says that one reason for platform problems is that generic development tends to increase the distance between the PTMP and the COP, since it represents an additional process to be handled; the Generic "Something" Creation Process (GSCP). To succeed, he recommends awareness against a number of platform pitfalls and prescribes a certain interaction between a lead customer as driving force and a carrier product directly coupled to an ordinary product development process.

Halman & Pohl[13] emphasizes the difference between single and family product development in the area of requirements engineering. Two supporting tools are offered to facilitate mediation between requirements for a specific customer and the product family; Categorizing and Use Cases. Also seven principles for combining such requirements are sketched, all based on implementation efforts.

From their work with industrial case studies on physical and amorphous products, Yang et al[15] breaks the platform development process into five sub activities. The steps are to: Identify Platform Drivers, Specify Product Architecture, Identify Platform Requirements, Propose Platform Improvements, and Evaluate Platform. They also emphasize the iterative nature of this process due to the architectural changes that typically appears during platform development.

5. KM and the NRIO Project

5.1. Dynamic Positioning. A DP system from Kongsberg Maritime is an advanced configuration aimed at stabilizing ships and other floating constructions in exact spatial positions over time without applying anchors. DP is typically used by oil- and gas industry related vessels[21,22]. Advanced mathematical theories, cybernetics, and control engineering are applied to analyze all the environmental forces from wind, waves, and current on a floating hull. The DP system regulates the amount of engine power needed to operate propellers and thrusters around the hull, to prevent drift. Signals from compass, gyros, GPS and other navigation accessories are used as references[20]. The system is constructed to counter influence from surrounding forces *before* they actually have accelerated the vessel away from its determination. By this, the energy needed to keep position is reduced.

5.2. Project Incentive. Although KM has several departments addressing market segments of automation, process control, and navigation, they have gained most recognition for their Dynamic Positioning systems. Within certain parts of this area KM is in control of more than 90% market shares world wide. The systems are manufactured by the DP Department in the Offshore & Marine division.

Growing markets, from the turn of the millennium, in the form of increased demands on both number and size of deliveries, as well as on technology, made KM see the need for a more effective DP system solution. A transition to distributed I/O units on serial interfaces had already been introduced to save lots of work with cabling inside ships. This was a paradigm shift. Now the requirements were more in the areas of security and enhanced demands for redundancy backup. It was soon clear that KM's old DP solution was not able to keep up with the new requirements. To avoid extensive change orders on old equipment, they chose to go for a radical renovation of the I/O architecture.

At KM, the department performing technological development tasks is called Technology Base (TB). Product development at TB is commissioned by the product groups inside KM, mainly the Dynamic Positioning group (DP), the Process Control group (PC), and the Integrated Control System group (ICS). Those act as the customers of TB. What TB offers is a Base Technology (BT) and a collection of HW and SW components which all their internal customers apply in different market segments. The product groups distribute their commodities through the Sales & Marketing Department (S&M).

Commissioned by DP, TB started to work on a DP system with technological solutions suited for future purposes. However, TB wanted the new component portfolio also to be adapted to ICS and PC needs. All departments were invited to join the project and to deliver their own product specific requirements. For some time there were internal disagreements within each department whether to join the project or not. Both PC and ICS finally dropped out, since they did not se any benefits at the time. Inside TB as well, there were different opinions about the project.

Nevertheless, the project was accomplished with TB and DP as the only cooperating parts. All the time however, TB had in mind that the new products should be built for PC and ICS applications as well, even if those sections did not participate. TB knew that the only rational solution in a long time perspective would be to join the three departments on a common product platform.

6. NRIO Changes

The NRIO case study was performed by interviewing project participants and main stakeholders from affected departments. Also a database of documents concerning the project was available.

During the study, two main areas of alterations became visible. The areas represent the product/ technical changes and the project/process changes.

6.1. Technical Changes. The technical product changes were about developing new hardware modules as substitutes for the old ones, and developing a new redundancy concept (RedNet). The new hardware consisted of a remote control unit (RCU501), a remote multi-purpose I/O unit (RMP200-8), a hub (RHUB200-5), a remote serial interface (RSER200-4), supporting devices (RSupport/RDUM200), and a new generation I/O BUS (RBUS). Also the form factor of most of the components was new. A brief summarizing of the main changes is listed in Tbl.6.1.

All the hardware modules were integrated by adding software components in the AK (Albatross Kernel) and the PCK (Process Control Kernel), where AK and PCK are parts of TB's established software layer structure.

Tbl.6.1: The Main NRIO Changes

FormFactor	New RIO200 modules based on old RIO400 modules. Reduced size, change from rack based to cost effective, plug & play DIN rail based assembly. Arranged for mass production.
RBUS	Serial communication, Manchester coded RS485. Based on its predecessor SPBUS. Improved by tripling throughput rate, tripling I/O capacity. Change to message based protocol.
RMP200-8	Based on RMP400. Reduced number of I/O channels to fit new redundancy concept. RBUS interfaced. Challenging HW development (internal power), changing requirements (time/cost overrun)
RHUB200	Based on its predecessor SPBUS-HUB. USB interface and support for triple redundancy added. Flat cable wiring interface changed to RJ45 interface.
RSER200-4	Based on its predecessor TBSS. Flat cable changed to RS422 and RJ45. NMEA compliant.
RCU501	Based on its predecessors RCU500/510. Change to RBUS interface. Power backup, inrush current limitation and ethernet interface added to improve redundancy. Improved memory layout.
Rsupport/ RDUM200	Developed to fulfil environmental requirements for the new RIO200 form factor. Providing stiffness and solidity to DIN rail mounted devices.
RedNet	Improves redundancy and real time abilities. Challenging and time consuming development of master/slave concept. Change of Time Tick (clock frequency).

6.2. Categorizing Changes. During the NRIO project, there was a close cooperation between TB as the developer and DP as the customer. After a period of development and internal testing of a certain component at TB, a prototype went to DP for further testing in their environment. If any faults, concerns, or issues indicated that a change was needed, a so-called "track" was established in a database. The track described the issue, its severity, category, status etc. It also suggested how the change could be performed. Tracks were continuously taken care of by project participants from TB. In some cases, a triangle meeting between the TB project group, DP project group and a TB/DP management team decided how to deal with the most challenging or controversy tracks.

From the material in the track database[25] we were able to extract a summary of the different change requests. The requests are tied to product, component, and fault severity. Tbl.6.2 shows a short overview on how the tracks were distributed during and after the project period.

Tbl.6.2: Tracks prior to and after Project Completion

number of tracks	Severity					Product						Category														
	Comment	Minor	Enhance	Major	Critical	AK	PCK	FW	I/O driver	AIM	HW	Net	RCU501	RMP200-8	RMP201-8	RSER200-4	RHUB200-5	I/O System	RBus	WinPs	Redundancy	ASYNC	Alarm	NMEA	Sys.Surveil.	Other
before completion	4	118	18	49	2	15	62	43	65	1	5	2	7	24	1	17	1	30	68	2	15	4	3	2	4	11
after completion			2	6	1		1	5	3					1		4			3		1					

Apparently, most issues have minor severity. The category involved in the majority of change requests is the RBUS, while the most affected product is the I/O Driver. The changes were mainly quite small, like correction of SW code or adding of functionality. They were not found to be objects of propagation to any great extent. Sometimes changes in one SW layer spread to the next levels claiming for corrections also there. Only occasionally did they propagate further to other products or categories. This implies some agreement with the findings of Clarkson et al[11], saying that changes may, as a rule of thumb, propagate up to four steps from the origin. In her comprehensive study Giffin[7] confirms this, finding that a change flow usually terminates rapidly, and rarely brings about a chain reaction beyond four steps.

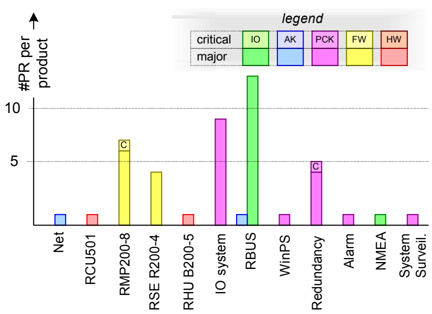

Figure 6.1 Bar Diagram of the Major and Critical Tracks prior to Project Completion.

Fig.6.1 emphasizes the tracks with major and critical severity in a bar diagram. As expected, most tracks with major severity are associated with the RBUS and the I/O System. Those are not necessarily more extensive than others tracks; they are judged to be of major importance since RBUS and I/O issues, quite often, affect the most fundamental system capabilities with impact on overall system performance.

The statistic is considerably smaller when it comes to change requests after project completion. After completion there were 2 enhancement proposals, 6 major tracks and 1 critical track.

6.3. Process Changes

- **Project Reviews.** Along with the NRIO project, TB made some changes tied to working routines and how a project is conducted. For several years TB has practiced Readiness Reviews (RR) on the product line according to common KM approaches. A new dimension was introduced by the NRIO manager, where also the project itself conducted internal Reviews. Those were named Project Test Readiness Reviews (PTRR) and Project Release Readiness Reviews (PRRR), and made it possible to have very short turnaround times (TT) for handling faults and changes. The project TT could be as low as one day during hectic periods. In contrast, a product line TT would usually be one month or more, due to the extended administrative apparatus typically involved. The left rectangle in Fig.6.3 describes the added functionality.

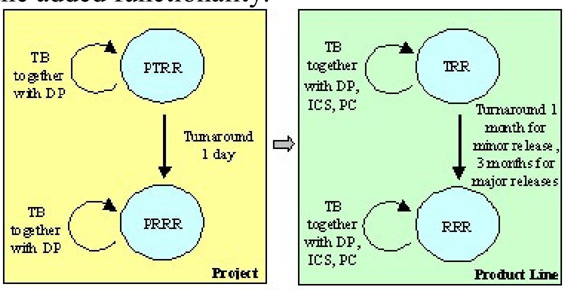

Fig.6.3: Readiness Reviews for Project and Product Line

- **Firmware Download.** A new firmware (FW) download and administration concept, Firmware Manager (FM), was developed during the NRIO project. Since firmware handling, over time, had grown rather complex on the old SPBUS, it was necessary to make the upgrade process a lot more efficient. The new RBUS technology made this possible. Both the technical routines and the logistics were improved to render possible an FW upgrade several times faster than before. In the present release, FM supervises that every system receives the right FW version. The new routines represent considerable progress both from a cost and safety perspective, since the number of upgrade error situations is strongly reduced. The transition to FM implied several changes in AK, PCK and the BUS driver.

7. Proposed Measures

The NRIO project was rather complex. Shenhar & Dvir[20] describes certain aspects of successful project performance dependent on complexity, novelty, pace, and technological uncertainty of the work (NCTP Framework[20]). In the light of this, it seems that the NRIO team did many things right. At the same time, there are room for improvements which may be important for proper handling of changes.

7.1. Communication and Documentation. Fricke et al[1] claims that lack of communication may both cause unwanted changes to happen and increase the impact of changes.

TB's attitude of heavily involving its customer, DP, was very advantageous to the project. Formal meetings held on regular basis ensured close relationship. Also a more informal communication existed, which as well was beneficial. However, the informality was sometimes exaggerated, especially when it came to documentation. DP systematically kept accounts on abbreviations and faults in the track database. However, reporting tracks to TB was usually done by telephone or email, and the tracks contained references to different emails. The DP project leader received copies of all emails. This manual way of operating was flexible, but also risky. All emails were not available to everyone. People who should have been informed were not always remembered. And new people entering the project had difficulties recover the history. According to DP personnel, the practice was quite vulnerable to mistakes and omissions.

Also inside TB a culture existed which sometimes hampered proper communication. According to TB itself, the department was good at doing things and finding good solutions. But they were not always good at documenting what was done, and especially *why* things were done.

To encourage communication, it is recommended that TB incorporates a more formal way of documenting their activities. A common database for all departments would facilitate both the flow and availability of knowledge. From what we know, the whole KG and some parts of KM have introduced Teamcenter™ as their preferred tool for documenting and managing processes. In other parts, TB included, there is a prevailing scepticism against Teamcenter™, due to its complexity. Also lack of common guidelines on how to use the system, like rules for storing, search criteria etc. is stated as excuse for not adopting the system. Despite the refusals, it is difficult not to recommend KM to arrange for a more widespread use of a common database. Proper documentation of changes is extremely valuable, and continuous learning from previous change processes is contributory to make the next process more efficient[1].

7.2. Requirements. According to several use cases studied by Rutka et al[3], understanding the dependencies between requirements is seen as an important aspect of information when it comes to change handling. Further, according to Fricke et al[1], one way of dealing with changes is to actually prevent them from happen as long as they are avoidable. More in-dept analysis of requirements may then provide for a reduction in the total number of changes[1]. It will also help validate the project in early phases to avoid changes in late phases, which often are the most expensive ones (ref. the "Rule of Ten"[1,6]).

Our judgement is that the requirements handling during the NRIO project was not sufficiently thorough to take care of all aspects concerning the product development. This can be seen from the considerable time and cost overrun, which, according to TB personnel, was much because changes appeared in requirements along the development phase. Those changes were to a great extent initiated by TB itself, as they repeatedly discovered new aspects and functionalities which they wanted to incorporate in the products. At the time, no requirements tracing existed, and the testing activities were not connected to requirements.

According to Heindl & Biffl[4], requirements tracing represents a systematic way of mapping dependence between requirements and products, also useful for change impact analysis. TB is in possession of DOORS™; a database tool used for different purposes. This tool has excellent prerequisites for requirements handling. It is nonetheless a fact that rigid requirements tracing may take huge effort. It should always be tradeoffs between the effort of using a technique and the value it represents. But even if TB does not regard it appropriate to trace every requirement in detail, we still think there is room for a more systematic approach to requirements.

7.3. Platforms. As we have seen, during the NRIO project TB faced one specific challenge: They had to mediate requirements from three main stakeholders without interacting with two of them. One effect of this was that the compact DIN rail concept made for DP cabinets showed not to fit PC/ICS. Later, TB also had to develop the new RIO420 series to utilize RBUS facilities in PC and ICS installations. By this, they experienced what Halmans & Pool[13] call "integrating new components into the product family". This is challenging, especially in a situation where the different stakeholders have different key drivers[15].

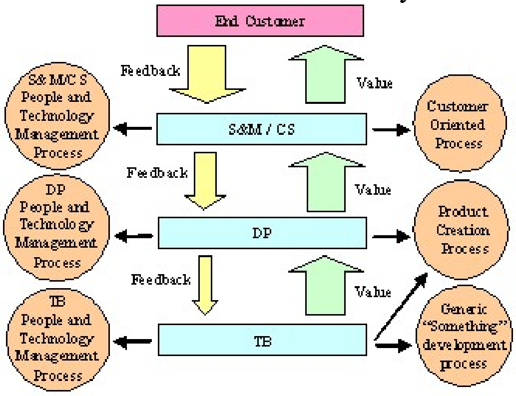

Fig.7.1: Value & Feedback Chains NRIO

Muller[2] claims that increase in the length of a company's value chain is a problem associated with generic development. This will inevitably throttle feedback, which follows a corresponding chain with opposite direction. KM has in general quite long value chains. Fig.7.1 shows the value and feedback chains during NRIO. As seen, KM's external customers had very little interaction with the product development. The NRIO was more like an internal project initiated for platform refinement. In itself, such a situation is often judged to be a pitfall. The effects, mainly experienced by Customer Support (CS), were some inefficiency regarding downwards communication (towards TB), vague product ownership in the chain (between TB and DP), and fragmented responsibility among internal groups at TB. However, since the NRIO was primarily aimed at DP, the system acted as a carrier product, vouching for success in normal product development. In view of KM's organization, arranging it the way they did probably was the best. Their external customers are in general not very concerned about DP system details. As long as the system works reliably and fulfils regulation requirements from the authorities, they are satisfied. In this way, it seems that TB did not suffer too much from lack of end customer feedback.

Some of the advantages associated with platform development were achieved, like reduced cost per function (over time), parallel developments of multiple products, improved quality etc.[3] Typical effects of platform pitfalls however, were not all avoided, e.g. both development time and cost were considerably higher than expected.

Recommendations for change handling in future generic development will be that KM steps up their attention to the known platform pitfalls[3]. They should consider clever organizational solutions for a somewhat closer interaction between TB and the end customers. This will shorten the feedback channels and even be beneficial for requirements handling. Combining the lead customer model and the carrier product model is usually the ultimate approach[3]. But since balancing of platform development and ordinary product development is difficult, they should formalize the trade-off between product family and customer specific capabilities. Also being better at selling their ideas will be of benefit, to include all affected internal stakeholders in their projects from the outset.

7.4. Change Experience Database and Change Effect Analysis. In the purpose of helping KM to deal with the challenges illuminated in this study, especially the information and documentation parts, we have developed a specific method for handling changes. The method was named Change Effect Analysis (CEA), and is a kind of change oriented FMEA.

● **Change Experience Database.** To perform a CEA, KM will need to be in possession of some background experience about changes from earlier, related or comparable projects. Such information can be captured in a change Experience Database (ED). It will take relatively little effort to build up an ED as long as it appears as an outcome of a continuous and embedded documentation process in all projects. Certainly, skilled designers could deploy CEA based on their own experience. However, the EA will provide valuable historical facts especially suited for guidance of attention, e.g. towards *unexpected* changes or impacts.

Tbl.7.1: Proposed Experience Database for the NRIO Project (Extract)

Predecessor/Basis	Change/ Changed to	Change Prop. 1	Change Prop. 2	Cost Effects	Value Effects
RMP400	RMP200-8	Cabinet layout DP		Increased stock Increased services Increased FW mngm. Increased price pr. I/O DP dedicated	Reduced longtime prod. cost Enable mass prod. Reduced TTM FW Manager
Old redundancy	RedNet	Time Tick	PCK/AK	Changes in time tick related code	Better sys.clock resolution Additional system clock Better redundancy Better real time

A log should follow every project. The ED would typically be part of this log as a final evaluation or "what did we learn" minutes. In cases where such information is not filed, one would have to reveal change impacts by interviewing project participants and searching in the more general base of documents. It would take a lot more effort to recover the information this way, especially if the project was finished some time ago. To facilitate usage of the material, the content should be organized and transferred to a common repository of documentation. This storage does not necessarily need to be equipped by advanced tracking abilities or searching tools; the most important is that it is available and somewhat organized. Special focus should be upon the nature of the particular changes, like how they propagated and what the effects were. It would be very important to identify and capture both the values (benefits) and the costs (disadvantages) connected to each change. The ED framework should be somewhat standardized to facilitate the work of supplementing it. At the same time, no rigid set of rules should hamper its flexibility. And to make it as little complicated as possible, the content should be rather amorphous, with describing text in an expedient and understandable matter. A layout example of such matrix is shown in Tbl.7.1, containing two top level changes from the NRIO project.

Change Effect Analysis. The content of a change experience database should further be applied as basis for CEA in a coming project. The same ED table format could be used as a starting point for CEA. In addition, columns for likelihood and importance/severity of changes could be added. Importance will refer to beneficial effects of a change; the value part, while severity refers to the cost part (disadvantages). To unit impor-tance and severity in one word, we chose “Impact” as notion. Then we speak of positive and negative impacts, which may be indicated by figures of opposite signs. A Change Number (CN) can be calculated as a product of the likelihood (L) and the impact (I). When a change causes several effects, their impacts can be summarized to assess whether the total effect of a change is positive or negative. A meaningful summarizing presupposes however that all impacts in a certain measurement refer to the same unit for every contributing effect, e.g. economy. If not, each CN should rather stand alone without being summarized. Changes with a large total Change Number (CN_{tot}), meaning a large positive figure, should be encouraged, while a low CN_{tot} should warn the development team about probable negative consequences of a change.

After project completion, the CEA form should be supplemented by and stored together with the ED form. The two can be compared to give even more information about the ability to predict changes and their effects. E.g. the comparison would be of help if a more systematic evaluation of the conformity between expected and occurred changes was to be made, like Lundval[9] suggests. An example of using CEA on one top level change to a certain NRIO component is shown in Tbl.7.2.

Tbl.7.2: CEA Exemplified on RMP400 in the NRIO Project (Extract)

Component/ Item	Change	Change Prop.	Effect	Likelihood (L)	Impact (I) importance / severity	CN (L*I)
RMP400	RMP200-8	cabinet layout DP	Increased stock	6	-3	-18
severe	slightly positive		Increased FW management	9	-6	-54
			Reduced long term prod. cost	9	5	45
slightly negative	important		Enable Mass Production	9	9	81

In this example both likelihood and importance/severity is stated on a scale from 1 to 10, where 10 is most significant. The CN is then the product of those two.

It is important to be aware that this method, like FMEA, is not an exact science. Assessments and assumptions play a big role in the appli-cation; there are no absolute answers. When using such a method, it is also necessary to “time box” the effort. Everyone familiar with FMEA may have experienced that building up forms can be a rather time consuming activity. This will be true for CEAs as well, if not confined. It always needs to be trade-offs when it comes to thoroughness and covering of every potential change mode. The amount of forms may easily explode and the time/cost increase awkwardly if common sense is left out. To avoid this, the recommended approach is to concentrate on the most severe/important and most likely changes to happen. And again the benefits of having a good experience database as reference should be emphasized.

● **Forward Change Effect Analysis.** Since the NRIO is a closed project, our initial CEA approach was retrospective. This is in accord-ance with the objective of the present study. However, CEA is mainly meant for forward use in coming projects, as already depicted. In the purpose of verifying the method then, we asked KM to apply CEA on a change in a planned or ongoing work. Since some experience is prerequisite for a good CEA, the company found it interesting to try the technique on the continuation of introducing FM. As we recall, FM was developed and launched along with the NRIO to improve firmware upgrade routines. In its present version, FM is able to verify that every module receives the right FW release during download. A wrong release will cause an alarm. From a security point of view this represents great progress. So far however, FM is purely a supervising “read only” tool. The upgrade itself will have to be performed manually. The final goal of FM is to change the whole FW download process into a fully automated activity. FM will then be able to both identify the current releases in all components and distribute appropriate new releases with considerably less need for manual operations than today. Such mechanisms will be introduced in a coming version.

Tbl.7.3 lists some examples of predicted top level changes and effects connected to introducing automatic FM. They are results of a time boxed brainstorm of potential consequences together with KM. The forecast approach was held on a conceptual level. The table represents an extract; it does not give a complete evaluation of every FM aspect. E.g. there will be different assessments connected to FW in new equipment developed for specific projects and in general spare parts for older installations. It will also be necessary to take some of the issues through lower level CEAs to reveal the more specific and technical effects of certain changes, e.g. in AK, PCK, CPLD, and infrastructure. Those effects will represent prospective further propagation of the top level changes.

Since all entries in the table can be associated with economy/cost, it is appropriate to summarize the effects in a CN_{tot} number. As we see, the number is positive, which implies that the change to automatic FM, from this limited assessment at least, should be encouraged. Also a table of issues not so directly cost related may be worked out to catch the more intangible and less measurable effects of changes.

It appeared that KM found the discussion of effects useful in itself. When guided in terms of importance/severity/likelihood and associated calculations, the brainstorming was contributory to making people more aware of change impacts, and the discussion correspondingly explicit. Hopefully, the CEA procedure may give grounds for enhanced consciousness around trade-offs and decisions in cases of change in general.

Tbl.7.3: Top Level CEA on Changing to Automated FM

Item	Change to	Change Prop.	Effect	L	I	CN
Manual FM	Automatic FM		Increased safety	8	9	72
			Reduced number of fault situations	7	9	63
			Reduced upgrade time	8	8	64
			Reduced spare part exchange time	8	9	72
		Staff	Reduced manual service	5	9	45
			Reduced manual supervision	4	5	20
			Reduced travelling	6	8	48
			Personnel training needed	8	-5	-40
			Reduced user threshold	8	6	48
			3. part operation possible	6	6	36
			Customer training	6	5	30
		Stock	Less need for stock operations	8	8	64
		Staff	Reduce number of stock workers	8	8	64
			More vulnerable to systemic errors	6	-9	-54
			More rigid release patterns	8	-6	-48
			More planning	6	-5	-30
			Less flexible	5	5	-25
		Production	More rigid quality control	8	6	-48
			Simplified working processes	10	9	90
		AK, PCK	Adaptation of SW code	10	-5	-50
		AK, PCK	Adaptation of FM concept	10	-5	-50
		RCU501	Adaptation of internal CPLD programming	10	-5	-50
		RCU501	Adaptation to platform structure	10	8	80
		RCU501	New version RCU501	8	-6	-48
		Infrastructure	Arrange/organize	8	-4	-32
		TB resources	Occupying time and effort (man-hours)	10	-5	-50
			Increased customer satisfaction	10	10	100
				$CN_{tot} = 371$		

Important / Slightly Positive / Slightly Negative / Severe

8. Conclusion

Change is an inevitable part of all product development, as well as product life cycle processes. In many situations the ability to change is a "be or not to be" for a particular product or even its manufacturer. This realization accentuates the importance of dealing properly with changes. It also encourages companies to handle the topic with thoroughness. At the same time, handling, and especially predicting, changes in large technical systems can be extremely difficult. The time and effort needed to comprehensively address the area has often shown to be beyond a realizable level in practice. Hence, it has been neglected.

This case study of the New RIO Hardware Line project at Kongsberg Maritime was performed to throw light on the changes experienced during the project. The purpose was to learn more about how changes behaved and how they were handled by the company. This again led to some recommendations for improved change handling in the future.

Our study disclosed that the majority of situations leading to change orders were of minor severity. Most technical changes were connected to software bug fixes and did not propagate beyond one or two steps from the source. Some of the major changes, especially on hardware, arose from particular technological challenges. Even if the propagation of those were still limited, the changes occupied lots of resources, as in the case of the RMP200-8 power supply and the RedNet master/slave concept (Tbl.6.1).

Another finding was that TB did many things the right way. Especially the technical outcome was successful. In the end the project team came up with a standalone DP system very well organized for mass production, with an improved hardware portfolio, communication BUS, as well as excellent redundancy and real time abilities. This was just what they had in mind. The new system represents a platform shift ensuring high technological performance and market competitiveness in the future, both for DP and PC/ICS products. Due to its qualities, the system is later approved by those refusing to participate in the NRIO project from start.

Nevertheless, the time and cost overruns show that not everything was running smoothly. From our findings we suggest that a few of the changes, resulting in unforeseen challenges, were contributory to this, and we believe that KM might have reduced the problems by paying a bit more attention to certain formal routines associated with Systems Engineering.

To facilitate change handling, some concrete advices were given to KM, like being more systematic on information flow, documentation, and requirements handling, and being more aware of typical platform pitfalls. As a helping tool, one specific easy-to-use method, which we named CEA, was introduced. The method is based on FMEA, but addresses changes in particular. Continuous building of an experience database about changes was seen to be an important prerequisite for applying CEA.

In general, our study supports Fricke et al[1] about the importance of communication and documentation to prevent, frontload and learn from changes. When it comes to change propagation, our findings agree fairly well with the discoveries of Giffin, while the work of Muller[2] is in accordance with our observation of success criteria and pitfall effects during generic development.

Even if the CEA method was proposed as an aid to KM for use in NRIO-like projects, the technique may have a more common utilisation. KM, or anyone finding this of interest, will have to further investigate to what extent the CEA method is applicable. If decomposed, the method will show to be much about putting change analysis into simple regulations of documentation and systemizing, to capture the effects of changes; in themselves good SE routines to be generally recommended.

References

Fricke, E; Gebhard, B; Negele, H; Igenbergs, Eduard: Coping with Changes: Causes, Findings, Strateg. (2000)

2 Muller, G: Product Families and Generic Aspects (2007)

3 Rutka, Guenov, Lemmens, Scmidt-Schäffer, Coleman, Rivière: Methods for Eng. Change Propag. Analysis (2006)

4 Heindl, M; Biffl, S: Requirements Tracing Strategies for CIA and Re-Testing (2007)

5 Hassine, J; Rilling, J; Hewitt, J; Dssouli, R: Change Impact Analysis for Req. Evolution using Use Case Maps (2005)

6 Keller, R; Alink, T; Pfeifer, C; Eckert, Claudia; Clarkson, P. John; Albers, Albert: Product Models in Design: A Combined Use of Two Models to Assess Change Risks (2007)

7 Giffin, Monica L.: Change Propagation in Large Technical Systems (2007)

8 Jarrat, Timothy; Eckert, Claudia M.; Clarkson, P. John: Pitfalls of Engineering Change (2006)

9 Lindvall, Mikael: Evaluating Impact Analysis – A Case Study (1997)

10 Kilpinen, M.S.; Clarkson, P.J.; Eckert, C.M.: Change Impact Analysis at the Interface of Syst.Eng. and Softw. Design (2007)

11 Clarkson, P. John; Simons, C; Eckert, C: Predicting Change Propagation in Complex Design (2001)

12 Matthiesen, S. (TU of Carlsruhe, Germany): A Contribution to the Basis of the Element Model “Working Surface Pairs and Channel Support Structures” on the Correlation between Layout and Function of Technical Systems (2002)

13 Halmans, G; Pohl, K: Considering Product Family Assets when Defining Customer Requirem. (2001)

14 Wikipedia: Change Mngmnt (Retrieved November 2008)

15 Yang, Tae G.; Beiter, Kurt A.; Ishii, Kosuke: Product Platform Development: An Approach for Products in the Conceptual Stages of Design (2004)

16 Salvendy, Gavriel: Handbook of Industrial Engineering: Technology and Operations Mngm. (2001)

17 Business Dictionary.com (downloaded February 2009)

18 Van der Linden, F.; Wijnstra, J.G.: Platform Engineering for the Medical Domain (2002)

19 Simpson, T.W.: Product Platform and Product Family Design: Methods and Applic. (2006)

20 Shenhar, A; Dvir, D: How Projects Differ, and What to Do About It (2003)

21 http://www.km.kongsberg.com/ (Retr.: Feb. 23, 2009)

22 Wikipedia: Dynamisk Posisjonering (Retr.: Feb. 23, 2009)

23 Kongsberg Maritime: Phase II Report New RIO HW Line

24 Kongsberg Maritime: Phase III Report RIO200

25 Kongsberg Maritime: Track Database

Acknowledgement

Special thanks are directed at our interviewees.