

The Role of Humans in Complex ICT Systems

Thesis for the degree Master of Science



Valentyna Kazymyrova

October 2016

Acknowledgments

I would like to thank my supervisor Kjell Jørgen Hole for providing support and constructive feedback, for fruitful discussions, reading suggestions, proof-reading, and for always finding time for meetings.

Many thanks to my colleagues. You inspired me to start this research, shared your knowledge and good examples to help me reveal the complexity of the analyzed system.

Special thanks to my husband, who was my first reader and sternest critic. Our thought processes work differently, but this is what makes us a perfect team. Thank you very much for your warm support and bright ideas.

Table of Contents

Abstract.....	5
1 Introduction	6
1.1 The role of ICT systems in the oil and gas industry.....	7
1.2 Structure of thesis.....	9
2 Definitions and Model Description	10
2.1 Information systems	10
2.2 Complex adaptive systems.....	10
2.3 Anti-fragility concept	13
2.4 Diverse software needs in a large company	14
2.5 System model and software tools description	15
2.5.1 Graphical representation	15
2.5.2 Software tools.....	16
2.6 Dependencies between model components.....	19
2.7 Summary	21
3 System Analysis	22
3.1 Are employees parts of the system?.....	22
3.2 “Where” and “when” are as important as “what”	23
3.3 Hidden and additional nodes in the system	26
3.4 Incidents having negative impact.....	32
3.4.1 Downtime.....	32
3.4.2 Unavailability of error-free data.....	33
3.4.3 Unauthorized access to sensitive data	35
3.5 Does a classical risk assessment cope with rare negative events?.....	37
3.6 Summary	40
4 Technical Improvements of the Company System.....	41
4.1 Redesign needed	41
4.2 Cloud use.....	43
4.3 Why control and validation.....	45
4.3.1 Access control and permissions	46
4.3.2 Input control and validation	49
4.4 Openness.....	52
4.5 Summary	53
5 Human Improvements of the Company System.....	55
5.1 Keeping everyone busy	55
5.2 Diversity	57
5.2.1 Projects to practice team diversity	58
5.2.2 Team size and balance matter	60
5.2.3 How to make diversity work?	65
5.3 End-user as a customer.....	67
5.4 Summary	69

6	Conclusions	71
7	Bibliography	73

Abstract

In the modern world, companies, regardless of their type of business, rely on information and communications technology (ICT) systems to carry out their everyday operations. The ICT systems have been developed over time to fit companies' changing needs. It is often hard to determine when and how these systems were modified because of constantly changing environments and stakeholders that come and go. A large industry such as the petroleum business is completely dependent on ICT systems and, as a result, it faces the typical problems related to such systems.

This thesis models the interactions between applications used by employees of a real engineering company. Analysis of the model unveiled several classical problems of complex systems, such as centrality, hidden relations, and closedness. The analysis shows that insufficient understanding of the interdependencies between the applications lead to unjustified actions that caused unpredictable consequences.

It is argued that the interactions between humans and information technology can never be excluded from the analysis of complex ICT systems without damaging the quality and usefulness of the results. Many real-life examples presented in this thesis show that humans can be the source of errors, but they can also be a critically important to rectify problems before the consequences become intolerable. The role of humans in ICT systems is analyzed from a bottom-up perspective with examples based on the author's experience. The conclusions are supported by case studies from everyday routines.

The thesis considers both theoretical and practical aspects of the design, maintenance, and analysis of complex adaptive ICT systems. Since it analyzes a real system, the thesis proposes several practical improvements like openness, advanced human error validation, and team diversity. While the thesis only studies the ICT system of a single company, the recommendations should be of interest to other companies as well.

1 Introduction

Being one of the most successful industries in Norway for more than 40 years, the oil and gas market has experienced hard times during the last couple of years. Norway's economy and welfare depend on this market to a large degree. To be more precise, the income from oil and gas comprises more than 15,8% of Norway's gross domestic product [1].

After many years of high activity, the focus has moved to cost-reducing measures and increased effectiveness. In order to survive the crisis most companies in the industry have had to adapt their existing workflows, as well as their methods, tools, and capacities to follow the market trends.

Considerable reduction of the oil price and offshore activity make it nearly impossible for Norwegian oil service companies to implement expensive changes or start large new initiatives to improve profitability. Instead, the companies are struggling to implement large voluntary redundancy programs and have started intense improvement programs to meet the challenges in the market. Employees must support each other through different improvement projects and focus on becoming profitable in a tough and demanding market. In order to do so, companies have to define clear goals and develop a plan of further changes and innovations.

But where should the companies begin? Introducing almost any change in a large company's operation makes it necessary to alter components of the company's ICT systems. When oilfield development and exploration were at the initial stage stakeholders had to find better tools and methods to do the job. It was also necessary to motivate the employees to carry out the work with acceptable quality and within accepted deadlines. Stepping into present times, the organizations have become so complex and tightly integrated that it is not always straightforward to determine the consequences of even a minor change to an employee's daily routines. The larger the company is, the harder it is to understand the interconnected structures that ensure operability of the whole organization and support the needed interaction between different departments.

In a challenging market that requires the same or higher quality work at a much lower cost, important modifications have to be made, including simplifying employees' daily routines, revise the organizational structure, and rethink the importance of interaction between the departments and their ICT systems.

It is not only a company's technical systems that are important. People are still the main driving force of any business. Being employed in the most

prosperous industry in Norway for a long period caused complacency among employees. Many of them did not care about winning new contracts or deliver results on time. To lose a tender did not mean that manpower would be dramatically cut. There were still more projects in the market waiting to be implemented.

A seemingly secure financial future has created a slow but steady working environment with an immense quantity of different internal rules and routines that now have to be reconsidered. Is the way we work effective enough? Do our deliveries satisfy customers' needs? Such questions emerge more and more often. However, a lot of us are creatures of habit and sometimes we can be deaf to even good reasoning and attempts to show that methods we use can be improved, especially if we have used them for a long time. It is possible to change processes and project life cycles in short time on paper, but in practice it takes years. Thus, the challenge is not only to find efficient methods to work in the current situation, but to make people use these new methods as well.

The intention of this thesis is to analyze a complex ICT infrastructure of a huge engineering company from the internal perspective of an engineer. The thesis considers interconnections, sometimes hidden and ambiguous, between different parts of the organization. It proposes a set of recommendations to improve the organizational structure and ways to increase effectiveness of the deliveries, taking into account features of the company's information systems. While the thesis only studies a single company, the recommendations should be of interest to other companies as well.

1.1 The role of ICT systems in the oil and gas industry

There was a focus on cost-efficient production during the initial development of ICT systems for the oil and gas industry. While proper information exchange and communication accessibility are important to any business, the most effort, and a considerable part of the companies' budgets, were used to improve process technology and realize industrial innovations. To this day, ICT systems are seen as tools needed to accomplish tasks. These systems continue to be modified to fulfill changing needs, but they were never the main objects of improvement.

Being a member of multiple engineering teams, the author has noticed that even in 2016 we can still suffer the consequences of insufficient attention to data gathering and data storage back in the last decade of 20th century. Drawings and other documents produced during that time still have meaning and must be

considered in current projects. In the best case, there exist bad quality copies of these documents, but often the documents have disappeared into the depths of the paper archives. Legacy databases with old documents still exist offshore and serve as archives, but current files are added to new data storage systems that are not compatible with the legacy systems. Since new employees are exclusively taught to operate the modern systems, only retirees can retrieve information from the old data archives.

The point is that ICT systems in oil and gas industry need closer attention in order to be optimized for current tasks and to be helpful tools for the employees rather than a source of frustration. But any improvement requires financial investments and, no less important, attitude change. It seems that such a change is already happening. A survey [2] conducted by Cisco Systems among petroleum industry professionals shows that 48% of the respondents consider “data” as the main area for improvement in their firms (see Fig. 1.1).

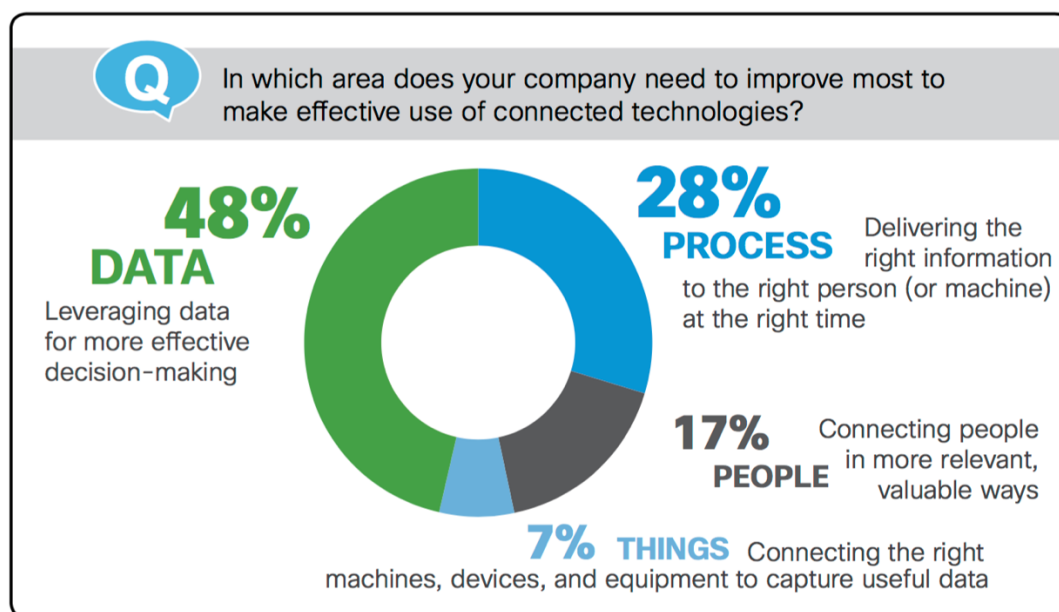


Figure 1.1 – Possible improvement survey results (picture from [2])

Today, ICT systems are inseparable from humans. Together they comprise a complex structure with diverse means of interactions, dependencies, and mutual influence. These complex systems are often hard to describe, and it is even harder to predict their behavior. However, companies need to better understand the behavior of their complex systems to conduct more realistic risk assessments and to become better prepared to handle possible negative outcomes.

1.2 Structure of thesis

Working in a large engineering company, the author of this thesis has analyzed its complex ICT infrastructure from an end-user perspective. Having to deliver high-quality results on time, it is important for employees to understand what else but themselves can influence the outcome of projects. Why is it sometimes practically impossible to complete the whole task on time, even if your own part was finished within the estimated timeframe? How to collaborate with other team members to deliver results above expectations? Does the engineering process utilize the information systems available in an optimal way? All these questions and more motivated this work.

The information provided in the next chapters is by no means a complete description of the ICT infrastructure. The description is based on the author's knowledge about the infrastructure's logical dependencies and does not disclose any essential business processes. While toy models are widely used for analysis, the behavior of the real system is considered in the thesis. The conclusions are supported by case studies and situations taken from the everyday life of the engineering company.

Chapter 2 models the ICT infrastructure from different perspectives and describes the different subsystems. It also introduces definitions that will be used throughout this work.

Chapter 3 discusses the reliability of the infrastructure, events that have strong negative impact on performance, and argues why standard methods for risk assessment are unable to provide realistic forecasts.

In Chapters 4 and 5 the author reviews possible improvements of the technical and human parts of the infrastructure. Some of them are global and require much financial support while others need a negligible amount of resources and can be tested with little effort.

2 Definitions and Model Description

2.1 Information systems

The backbone of a company is the information and knowledge contained in its ICT infrastructure [3]. To build an efficiently functioning organization, a company has to deploy information systems that satisfy its requirements for processing, exchange, and storage of data. There exist many definitions of information systems [4], but we are not going to spend any time trying to figure out which one is the best. Instead, we will explain information systems by means of a company's work processes. Based on the diverse needs of a company, work processes include procurement, engineering, financial services, manufacturing, acquisition, service promotion, and logistics. The functionality of these work systems depends directly on the information flow connecting the systems together in a complex pattern. Thus, an information system is a system in which machines and humans perform work activities using information, technology, and other resources to produce informational products and/or services for internal or external customers [4].

An information system must cover the needs for planning, organization, coordination, and control of information activities and processes, as well as communication inside an organization. Creating an adequate information infrastructure is important. The task needs people capable of understanding and creating a general overview of the processes and the roles that information plays in a company.

2.2 Complex adaptive systems

The term "complexity" has been applied to many very different things and systems in various fields during the last decades, not only in a scientific sense but in advertising slogans as well. Examples of such usage are "complex brand development" provided by PR-management companies, "complex programs-development and customers' support" from IT-companies, "complex approach to your problem solving" as part of psychological help advertising, and "complex examination" from medical services. It seems like the term "complex" adds value to a described object. But do all these entities use this term in the same way? Researchers do not agree on a common definition of complexity [5-8]. The

disagreement can, at least partially, be explained by the fact that complexity is a collective term with different meanings dependent on the context.

Quite often scholars distinguish between a complex system and complex adaptive systems (CAS). In terms of computational technologies, an “adaptive system” refers to a process where an interactive system adjusts itself to suit the purposes of a certain user or to fit a changing environment. This property requires that the system has the ability to gather information from the surroundings, analyze it, and conclude that some adjustments are needed. It implies the existence of feedback loops in the system [9].

A complex adaptive system in this work is a set of entities that interact with each other in a mixed (ordered and/or disordered) way featuring feedback loops (see Fig. 2.1).

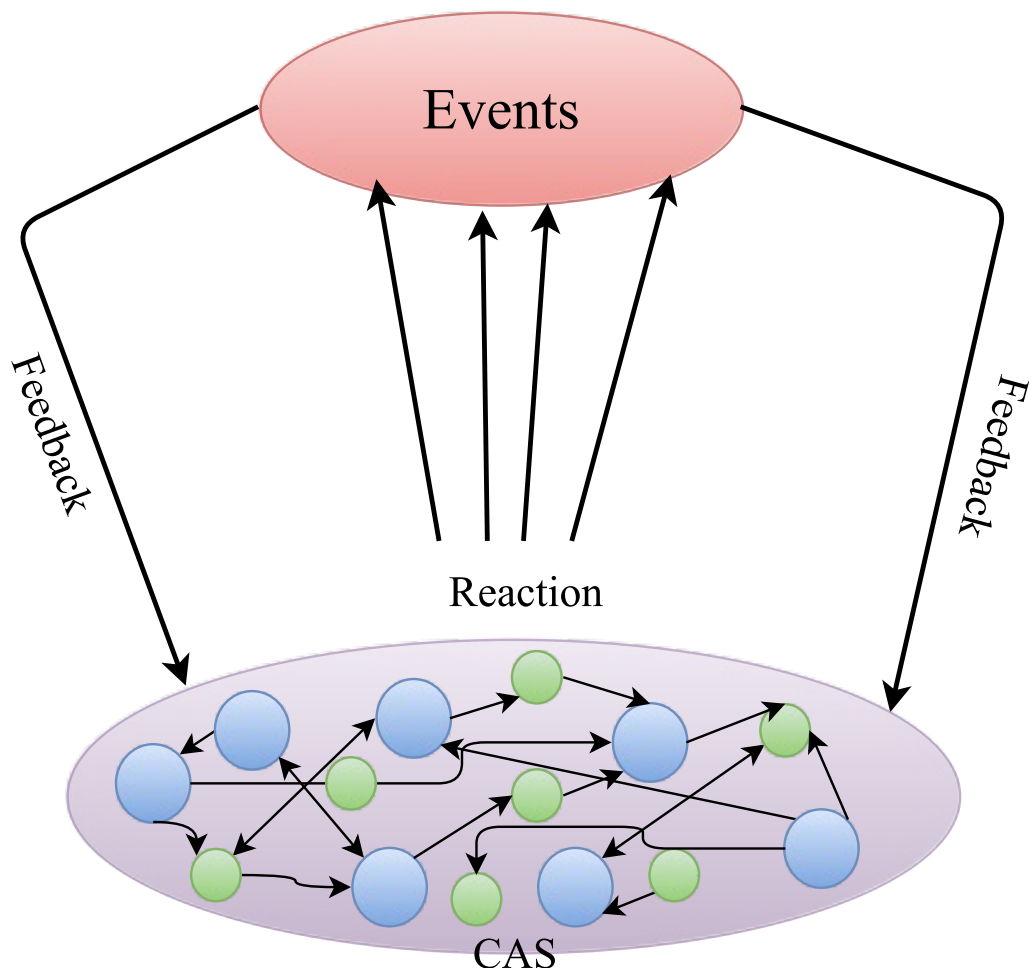


Figure 2.1 – Complex adaptive system

Many natural systems (e.g. ecological systems, societies, and brains) and man-made systems (e.g. the Internet, artificial intelligence systems, ICT systems) are complex adaptive systems. Some of them can even be of a mixed type, i.e. a system consisting of humans and artificial intelligence software. Such systems

cannot be easily represented by a graph or some other standard means of representation. One of the distinctive features of a CAS is a tangled and/or hidden set of connections between its elements. Even if it seems easy to determine all logical connections between the elements, there are always some more covert and unpredictable ones. That is why a CAS is hard to analyze. One has to study the system as a whole; working just with separate elements will not give the needed understanding. It is like working with chemical compounds: a substance obtained as a result of chemical reaction has another set of properties than its elements. The final substance will not inherit all the intrinsic properties of the ingredients. Some combinations of elements can give unpredictable results.

While analyzing a CAS, one can quickly run into insurmountable obstacles. Regular methods of analyzing systems do not work, there is no good way to order interconnections, and it is even hard to identify all of them. The behavior of such systems is in some cases unforeseeable, and it is quite complicated to determine when and how this unpredictability will occur. This is one of the reasons why studies of CAS are fascinating to researchers with different backgrounds and interests.

Dependent on the primary objective, an analyst will pay special attention to certain aspects of a CAS. In our case, we are focused on finding out whether the users of the system described below can understand how the system is functioning, whether it is optimal and straightforward, whether it is exposed to harmful external or internal effects, and whether it is possible to make it more robust.

We are going to describe functions of elements, the connections between them, and explain how different user groups operate the system during different stages of their projects. Connections between the entities of a system define dependencies in this work. An entity *A* is dependent on entity *B* if *A*'s operation is influenced by entity *B*. Entities *A* and *B* are interdependent if they influence each other (see Fig. 2.2)

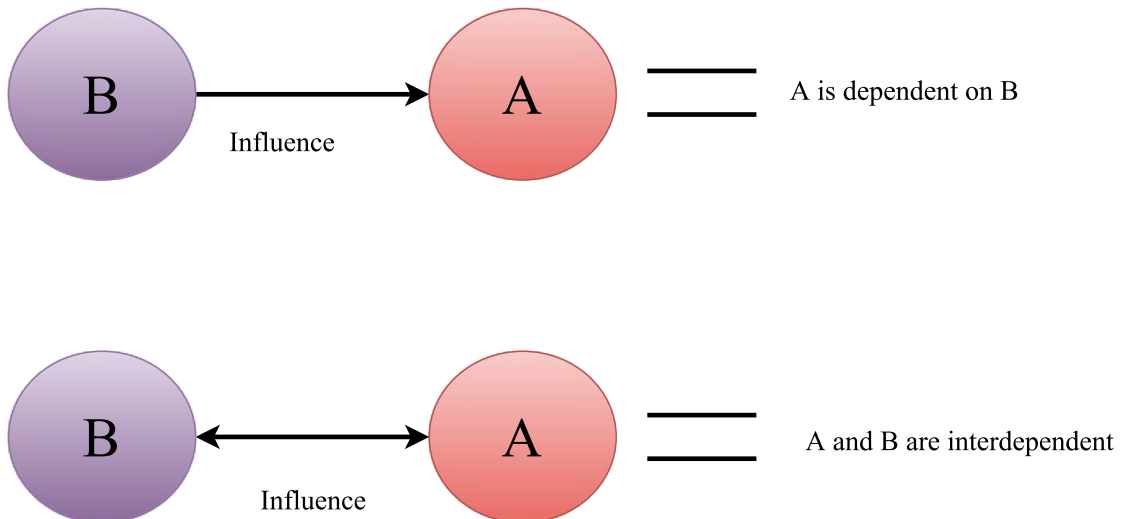


Figure 2.2 – Dependency and interdependency

2.3 Anti-fragility concept

The concept of anti-fragility was introduced and developed by Nassim N. Taleb [10]. A system is said to be anti-fragile when it is capable of not only withstanding certain types of impact like robust systems do, but also to learn from them (see Fig. 2.3).

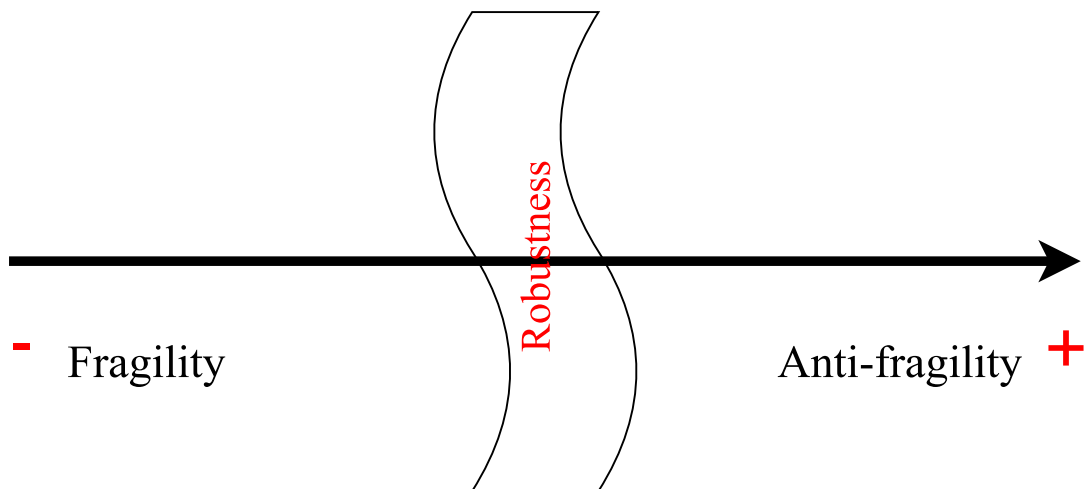


Figure 2.3 – Anti-fragility concept

Anti-fragile systems need both negative and positive events to learn to adapt their reactions and behaviors to maintain anti-fragility. No system can be anti-fragile to all possible types of impacts; this is why it is crucial to understand what types of impact are the most critical when designing new systems. A fragile system is unlikely to become anti-fragile. However, a robust system can become fragile. Even if a system is made robust to a certain type of impact, this system property needs to be maintained. Any CAS becomes fragile over time if it is not

maintained. There are no general methodologies to create anti-fragile systems. Kjell J. Hole has described core design and operational principles needed to move towards anti-fragility [11].

2.4 Diverse software needs in a large company

To create a well-functioning organization, it is important to have rules and principles that maintain a suitable working environment. To complete a task where two or more people are involved, everyone needs to understand their roles and responsibilities. The ideal situation is when the task structure is crystal clear, all team members have experience with the tasks from before, time and other resources are not in limited supply, and nothing can go wrong. Of course, this never happens in real life. That is why when you start working at new company your new manager outlines the routines for how the employees work. It can be hard to remember everything, there definitely will be some points that are difficult to understand as long as you have not been doing them yourself, but you will at least get the feeling that the company has things under control and seems to function properly.

We view a working environment as a combination of employees, software, and hardware connected to each other. The combination of components varies based on the type of task and methods that are used. For example, to perform the daily work of a secretary, there is a need for a standard PC and means of communication, as well as standard programs like an office package for information processing, a mail client, and a browser. But when the matter concerns some more specific and complex type of work, an advanced software package is needed. Quite often software products from known development companies are used for these purposes as well. In other cases, we get customized products made by a small IT company, or developed by an in-house team of developers to suit a particular purpose. In this case the risk of getting a multi-headed monster bred by several generations of developers must be considered. It is unlikely that IT departments whose main responsibilities lie within operational maintenance are able to develop high quality software products and support their lifecycles.

To maintain the operability of a large company with an elaborate set of services, many specific software applications are introduced to employees. A company project can be of a different size and duration, involve different departments, and include collaboration with some of the company's partners. In order to make a quality product and to provide a service within an agreed

timeframe and cost, efficient tools must be used to assist and simplify the everyday tasks for all departments within the organization. These tools must be used in accordance with the execution model adapted by the project.

We are going to examine a system in a company (hereinafter referred to as the Company), which provides engineering, procurement, commissioning, and project control services, including installation, prefabrication and logistic. Furthermore, we are going to take a look at the specific software used and especially their dependencies and interconnections.

2.5 System model and software tools description

2.5.1 Graphical representation

The main interactions between applications analyzed in this thesis are represented by the graph in Figure 2.4. The graph was created by gathering information from sources on the Company's intranet. Arrows connect pairs of application nodes. An arrow between two applications indicates that there is an information flow between the applications. The application pointed to by the arrow depends on input from the other application. The arrows show the information flows and thus the dependency directions; double arrows mean that information is transferred both ways and the applications are interdependent. An information flow can be routed through adjacent applications if there is no direct connection. This case is not depicted on Figure 2.4.

We consider the applications and their connections as a single system. The information flow between nodes is not instant and goes according to a predefined schedule. Informatica Power center is used for this purpose [12]. Power center makes it possible to send data between different databases regardless of the data format. It is used to ensure dataflow even between totally different (e.g. format, structure, etc.) information sources. For example, if one needs to transfer newly inserted information from TIME to MIPS (see description of the programs below) to accomplish a task, it is necessary to wait until next scheduled transfer before completion will be possible.

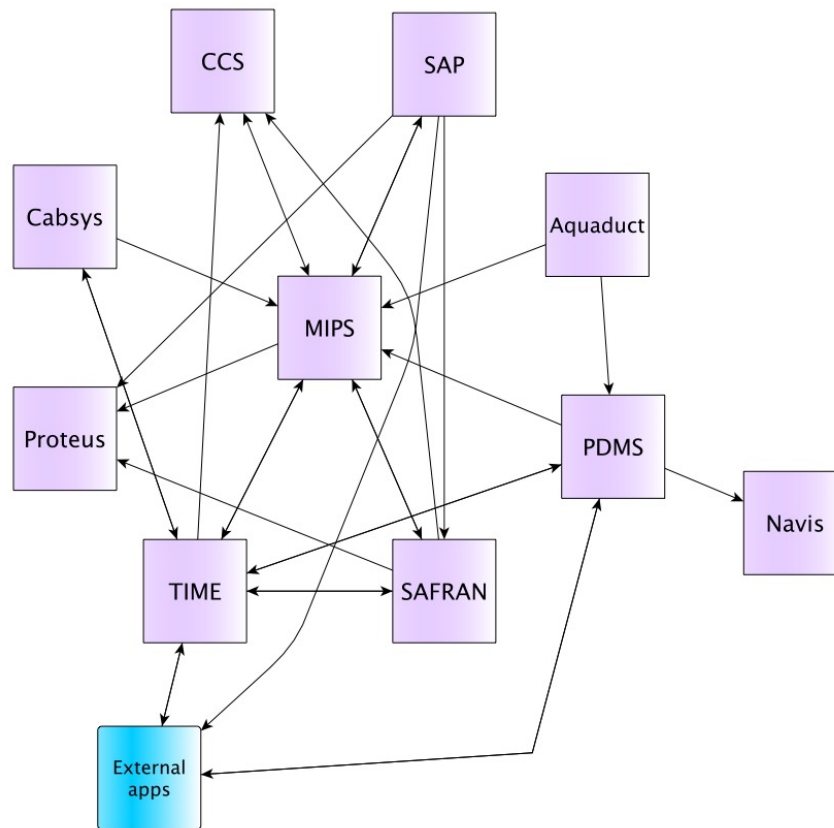


Figure 2.4 – General system model

2.5.2 Software tools

TIME is an engineering database that is used to store and update technical data about tagged equipment, tagged bulk, lines, cables and signals. Moreover, it is a control system for documents handling project documentation from equipment and materials vendors, as well as documents created and updated internally.

Engineers from all Company departments use *TIME*. The database contains equipment descriptions, technical characteristics, assigned functions, coordinates of equipment positions, cable types, and planned actions (i.e. installation, upgrade, and demolition of equipment). *TIME* makes documents available for review, including drawings and certificates from vendors. An engineer responsible for procurement can accept, reject, or comment on vendor documentation and send a response to a vendor. Thus, *TIME* provides vendors with important feedback. All acquired vendor documents and their version histories are kept in *TIME*.

TIME facilitates the maintenance of engineering documents, including internal discipline checks of the documents issued by a project. Moreover, *TIME* stores other documents without an assigned document number, for example,

correspondence, presentations, minutes of meetings, reports, and checklists. Finally, TIME contains information about system topologies and implemented designs.

TIME tracks the progress of engineering activities. An activity's progress is measured by the percentage of completed work. Management receives weekly progress reports. The reports allow management to create an overview of a project's progress and change plans or introduce other compensating measures when the progress is unsatisfactory.

TIME is one of the main tools for all engineers in the Company. Dependent on the stage of a project, TIME accumulates information with varying levels of details regarding work activities, documents, progress, and equipment to be used during installation. TIME communicates with other applications by pushing new information and fetching updates based on a predefined schedule (see Fig. 2.4). It is obvious that information stored in TIME is sensitive and very interesting to people involved in industrial espionage. Therefore, the database must be properly protected.

MIPS covers all phases of the project execution model from the start of the system engineering to the hand-over of a tested and installed product to a customer.

Engineers create job cards in MIPS. These job cards consist of information needed to perform certain tasks on an oilrig. They are divided in chapters and contain, for example, a work description, a detailed sequence of operations to be done on a piece of equipment, materials required, and documents and drawings that can be useful during the work. MIPS provides an estimate of the number of hours it takes to complete the work, dependent on its type and complexity. Moreover, MIPS is used to order materials necessary for the final product. Different departments (i.e. engineering, commissioning, procurement, material technologies, and planning) report on their deliveries for every job card in MIPS. It is possible to follow the entire lifecycle of a job card in MIPS.

MIPS is an important tool for financial control by accountants. The hours spent on different projects every week, registered by all employees through SAP, are sent to planning systems with the help of MIPS.

SAP is a well-known and powerful platform for resource planning. It is used for administrative purposes covering finance, salary, manpower planning, and man-hours register [13]. Any employee can fetch information about payroll, travel, registered work hours, rotation, and tax deductions for his or her account.

SAFRAN is a project control application that includes project planning, progress reporting, and project management [14]. According to the planning department, it is a comprehensive, flexible, powerful, and easy-to-use tool. SAFRAN has a built-in control system for variation orders (change requests that need to be accepted by the customer). It is also a tool for creating graphical or table-style reports in seconds. A flexible approach to reporting is considered to be a must among experienced planning personnel. Reports and graphics are communication mechanisms that drive the demand for better planning and scheduling [15].

Proteus is an additional application for reporting and visualization of planning data. The planning department uses Proteus to create a workspace with all data relevant to a project. The main features are advanced and easy to share visualization reports. Proteus is used as an accompanying tool to SAFRAN, providing more flexibility and configuration for visualization of planning data.

PDMS is a fully integrated 3D multi-discipline design environment, primarily used during structural and piping tasks. Necessary tagged equipment from other disciplines is also included in the 3D model. Information from a model might be used for early material purchase, especially for structural and piping tasks.

Naviswork is an application to view and walk through 3D models. It is used by “white disciplines” (i.e. electrical, instrumentation, telecommunication, and automation) for analysis and finding available placement for new equipment. The application has a supporting and informative function.

Aquaduct is software helping to choose pipes according to specifications. Results are transferred to PDMS allowing early material purchase.

Cabsys is an application for cable routing. It is used to estimate needed cable lengths and to maintain a network of cable guides.

CCS is a tool to register changes in a design due to incomplete information or other reasons. The person initiating the change describes the change and its possible consequences, chooses the disciplines that might be affected by the change, and sends the description of the change to a manager for approval. There is an opportunity to choose a change type: internal design change request or external technical query issued to client. The change is also send to all potentially affected disciplines for evaluation. In the case of an internal request, it is up to a change board to decide whether or not this change will be implemented. In the other case, a client has a right to approve, approve with comments, or decline the change. Additional materials, such as documents, photos, or graphics can be

added as attachments to any request to document the importance or to explain complicated technical issues in detail.

External customer applications are used by the Company to acquire information about existing equipment and technical details that are needed during a project. Moreover, external ICT systems are used to coordinate actions and plans, and to perform commissioning. The updated documents and information about equipment are transferred back to a customer's system at the end of any project.

2.6 Dependencies between model components

Let us consider a typical project at the Company and describe how the dependencies depicted in Figure 2.4 are created. When a contract is signed, one of the main priorities is to establish routines and tools to control the flow of technical information. This initial setup shall be done in accordance with the client's requirements for life-cycle information (LCI) and data transfer. The information management (IM) department is responsible for this stage, and makes relevant tools and routines available to the project team.

It is difficult and time-consuming to configure all systems and dependencies. However, it is only necessary to establish the connections between the applications actually used by a project. The initial set up of a project should be done in cooperation with the project team to eliminate applications, which are not required at all or are not needed at the initial stage of the project. If, for any reason, it is impossible or time-consuming to determine the needs of a project during the startup phase, then standard connections priority list shall be used to determine a minimum set of connections and the sequence in which these connections are to be established. For every connection, the list shows which application is an information source and which is a recipient. It also determines a synchronization schedule. The list gives the IM department a general idea of what shall be done and when. No one has ever seen a connection with priority one, so all priority lists begin with priority two, leaving priority one for some unexpected non-typical need. An actual sequence of connection setups, especially after the first 10 priorities, depends primarily on the project specifics. Section 3.3 discusses a concrete example of such a list.

Company departments need different sets of tools and programs to complete a project. These sets are likely to intersect. Let us add a "human" layer to the system model. The employees interact with the different applications in the system all the time. They provide input to the applications, which definitely

influences the outcome, create feedback loops in the system, react to different events, and adapt their behavior. To accomplish their goals, the main departments involved in a project (engineering, planning, procurement, and management) use specific sets of applications. Figure 2.5 depicts how these sets intersect.

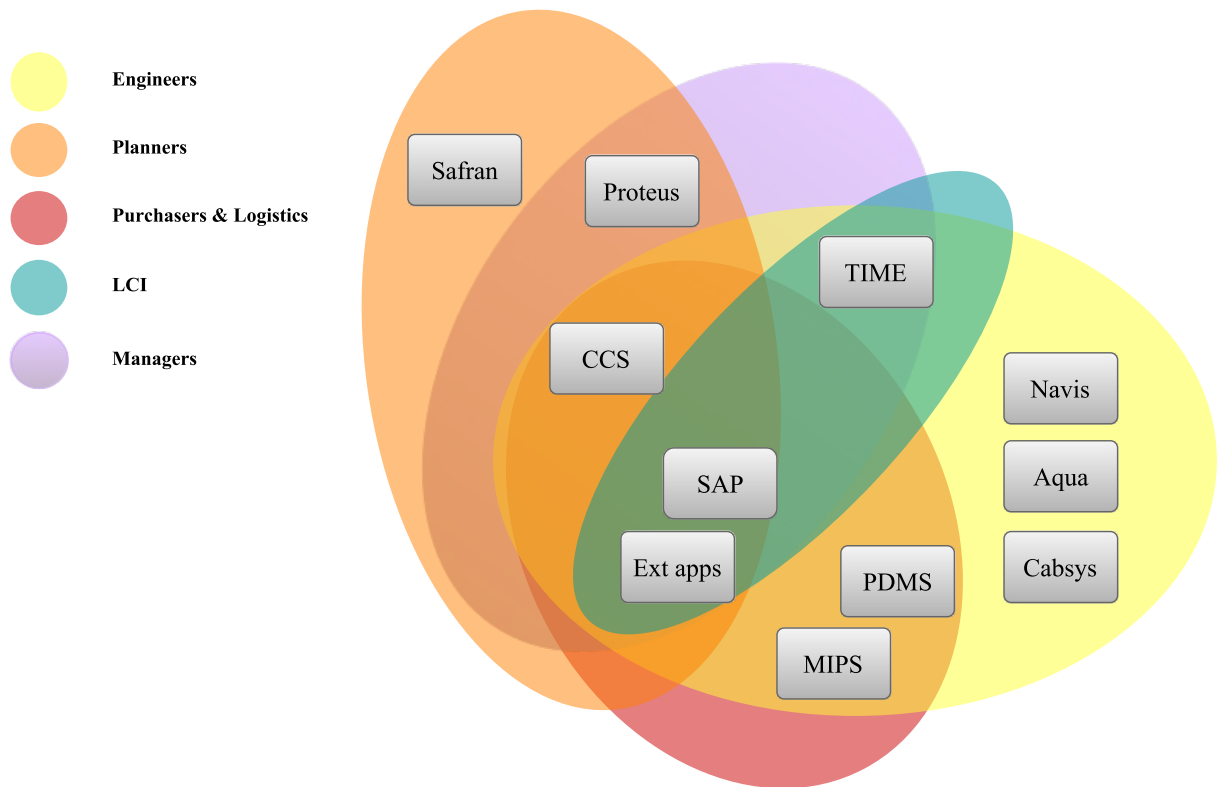


Figure 2.5 – Intersecting sets of the applications used in different departments

The figure above shows that all departments use SAP and external applications (owned by customers), and these programs seem to be the most important in terms of information flow.

Visualizing data flow from another prospective, taking into account Figure 2.4, it becomes clear that MIPS has a central role, not SAP or external applications (see Fig. 2.6). MIPS has the most ingoing and outgoing connections, serving as a hub for the information flow. Its role in a project is not clear for all user groups. As it can be seen from Fig. 2.5, MIPS is not included in the workflow of many departments, but it still has a “hidden role” in a project even if not used by end-users. Section 3.3 will study an example of this phenomenon.

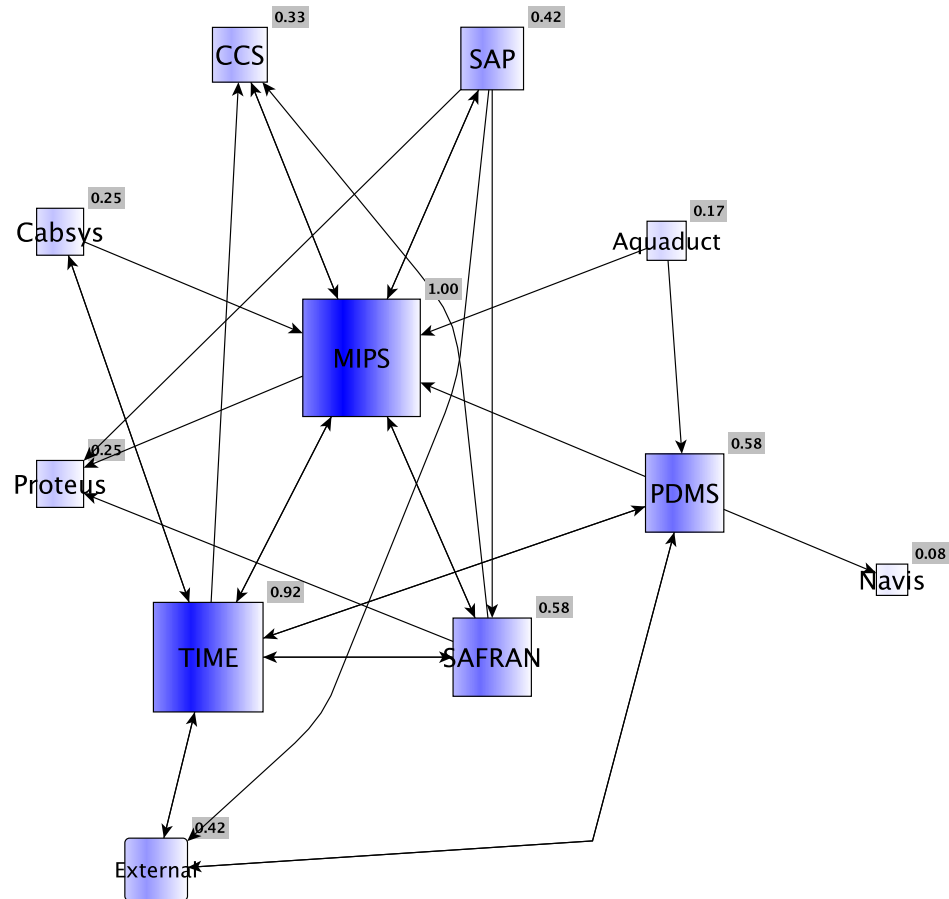


Figure 2.6 – Nodes’ “weight” based on ingoing and outgoing connections

2.7 Summary

The diverse needs of the Company lead to frequent system changes. Each department uses a certain set of software nodes to complete typical tasks. An end user has no overview of the interactions between nodes unless he or she actually uses them. An analysis of individual nodes does not reveal a full picture of the system since it does not take into account the interdependencies and mutual influences between nodes.

Describing a system from various perspectives can give different results. The result of any system analysis is highly dependent on the initial model. Therefore, finding a proper representation must be addressed during the initial CAS analysis. Furthermore, it is necessary to study the system from different perspectives to reveal surprising connections.

3 System Analysis

This chapter discusses the role of employees in the Company's ICT system, as well as the role of information, its flow and context. Furthermore, it illustrates possible consequences of hidden nodes and new nodes that are strongly connected to the rest of the system.

Any modern ICT system is constantly interacting with its environment, including other ICT systems. It is discussed what kind of events have strong negative impact on the performance and reliability of the Company's system. Finally, the chapter explains why standard methods for risk assessment are not always able to provide realistic forecasts and why they provide limited guidance on how to develop adequate prevention measures.

3.1 Are employees parts of the system?

A major improvement of an ICT system matters little unless the stakeholders understand the impact of the upgrade and are willing to change the way they interact with the system. Well-functioning business processes that describe human behavior must be tightly integrated with information technologies to ensure good outcomes [16]. The Company has formalized and integrated simple low-level steps in its business processes. For instance, the case of registering a new tag is a well-defined sequence implemented by an application. High-level judgments are not supported in the same way by the ICT system. Since there are no formalized processes to decide atypical problems, improvement of the ICT system has little or no influence on rare high-level processes that may not be repeatable.

The situation is different when it comes to mid-level processes. The system provides documents and official procedures, but leave the details of how to carry out a task up to the employees. The employees may even decide to use different tools and procedures than the ones prescribed by the system. Is it a problem when an employee decides to do things differently? It could be that the existing procedures are not understandable to the employee or they do not describe a good way to complete a task. The employee may also lack the computer skills needed to carry out the prescribed procedures. The result is the same in all cases: the implemented procedures are not used to their full potential and the working environment does not provide the employee with necessary support.

Stakeholders must be an integral part of an ICT system to be effective in their everyday routines. Only well integrated computer-human systems lead to success. This observation does not imply that only a person with intimate low-level understanding of a particular part of a system should be allowed to use it. Such limitations make a system more vulnerable and strongly dependent on one person. A mid-to-high level of cooperation between the ICT system and its users, needed to efficiently carry out regular and more seldom (but repeatable) tasks, is what we are looking for (see Fig. 3.1).

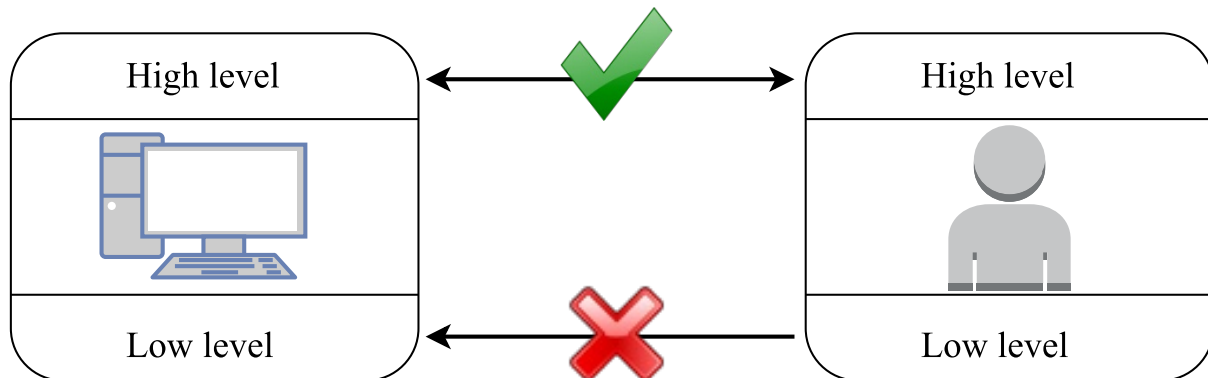


Figure 3.1 – A proper computer-human integration

An organization that wishes to benefit from an information system consisting of both people and machines must teach the employees how to use the latest, more productive technologies to complete their work in a simpler and faster way. The technical system must also be adapted to the needs of the employees. It is necessary to organize regular meetings between stakeholders to discuss whether the system provides adequate project support. Adequate support is particularly important during the life cycle of a large and difficult project. Representatives from different departments working with various parts of the system should attend these meetings. A team with diverse knowledge can base its evaluation of current challenges on previous experiences from many areas. The importance of a diverse team is discussed in Section 5.2.

3.2 “Where” and “when” are as important as “what”

Two large cooperating corporations can afford to engage four different departments in the acquisition of a 100-meter network cable. In a situation like this, a responsible staff member may not be informed about the acquisition in time to avoid problems. For example, before the responsible engineer has even had a chance to look at the cable, somebody else decided to buy it.

A purchase department may want to know whether 50 m of cable is enough, since one of the preferred partners has a network cable with a very low price, but only 50 m is available. The material technology department awaits a datasheet for the cable in order to add it to the existing material database, although the same cable type was purchased a month ago. The department cannot find the datasheet since their new internal routines requires that the color of the cable is written as “BU” (according to international standard) in the title field, while the old value was “Blue.” At the same time the logistic department would like to know when the cable should be sent offshore, though installation has not been planned yet and nothing was purchased.

When the responsible engineer tries to answer all these questions in one mail to all involved, the whole supply chain collapses. Most of the recipients are not able to pick out the information they requested to perform their specific task. In the best case, the engineer will get a couple of emails asking to specify something already stated in the original mail. The situation rapidly becomes worse when a person who did not understand the original email correctly, decides to fill in missing technical details himself or with the help of people lacking technical knowledge. The risk here is that the company may send the wrong type or amount of cable offshore, while the employees involved in the purchasing process are very proud that the work was done fast and cheap.

Just because someone thought that the last letters of the cable type are not important, a 500 m drum (the price was good, why not to purchase more?) was thrown away since the cable was outdated and could not be used.

The situation described above is a real life example of how things can go wrong when multiple departments are involved in a process. Imagine the situation when the task is not simply buying a rather cheap cable, but involves buying some very expensive hardware. Lack of communication and understanding about details could become enormously expensive for the company.

The described situation indicates that meaningful information at the right time is as important as the quality of the information itself. In certain situations, it is crucial to not overload employees with details they do not need. Informational overload can lead to wrong conclusions with unpredictable impact because the employees did not manage to extract and understand the information relevant to them. Accessibility of information is an important matter as well. People prefer to access and process data in different ways. Some people like to discuss things orally and simply continue their duties based on the received information, while others will do nothing until they receive a written summary of decisions made.

In one of my projects, colleagues only accepted the unreliability of the current record keeping method after the third replacement of a person in a key position. The two first arrivals of a new person caused the development of the basic design to be delayed for weeks or even months. We were forced to discuss already defined and frozen solutions just because the two former key employees had not seen any value in keeping records for a large project that lasted two years.

Everyone in a project team has to figure out what kind of information they need to be able to perform their assigned tasks, what kind of data he or she possesses that can be useful to their colleagues, and where the latest version of the information can be found. As was already discussed in the previous chapter, the Company has specially designed software applications to complete engineering projects. The applications and the communication between the applications are configured at the initial stage of a project. We all have individual ways to document our working processes, but unfortunately not all of us clearly understand how others depend on the decisions we make. As a result, following the agreed procedures regarding data flow is seen an unnecessary rule that can be neglected.

As an example, we had to design a new network for an automated safety system. At a project meeting it was decided to use switches with 24 ports. One of the customer's requirements stated that every network unit had to be designed with 50% spare capacity, which was not fulfilled with 24-port switches. It was concluded that the document containing this requirement was not governing for the whole project and could be ignored. The responsible person added the information about the selected network switches (i.e. tag numbers, model, etc.) to the engineering register, which is a "master copy" for this kind of data. All other employees who were involved in the project used this information.

After about one month, the person responsible for selecting the switches had a meeting with the customer, who said that 50% spare requirement was a "must." So the responsible engineer changed the number of ports from 24 to 48 ports in her order and updated her personal Excel-list with a new model and dimensions. She did not consider this information to have value for other colleagues so no one except the purchaser was notified. She only updated the shared engineering register after three months because of a planned milestone check. It came as a complete surprise to the other members of the project team when they found out that:

- Not enough bulk materials (i.e. patch panels, cables, connectors, etc.) were ordered, but that could be fixed quite easily.

- There was not enough space in the ordered hardware enclosures since standard 24-port switches needed 1 unit height in a rack, while 48-port switches required 2 units. The extra spare capacity requirement for the new hardware enclosure was 25%. The first most logical option was to order an extra enclosure and to rearrange the equipment. But there was not enough space for this solution in the equipment room. Hence, it was decided to cancel the order for all the enclosures and replace them with another special designed model, which was 20 cm higher and thus gave us 4 extra units in each enclosure. Of course, any custom designed solution is usually much more expensive and has a much longer lead-time.

In this case, all the additional time and money wasted could have been easily avoided by just having the right information in the right place and at the right time.

A proper operation of a system does not depend only on a well-functioning information system. The “human” part of the system can be a significant source of delays and data corruption as well. Clearly defined information flows, openness, and good communication between the involved stakeholders can noticeably shorten the list of events leading to failures.

3.3 Hidden and additional nodes in the system

As was already mentioned, it is important for the end-user to understand an information flow within the system and logical dependencies between different applications. It helps to find the source of an error when the required information does not appear at its final destination, as well as to make better estimates of the time needed to complete tasks. If the end-user is not aware of some stops the information takes on its way through the system, the user can easily underestimate the time needed.

Chapter 2 introduced the concept of a priority list. Table 3.1 contains a shortened example of a list we will use to illustrate how the information flows through a system in practice. For example, if personnel have already begun to work on a project, they have to register how many hours they worked on the project every week. A job code (or an activity) is used to register the hours in SAP. The registrations allow management to track the progress of the project. To be able to monitor the project, planners need to create project activities in SAFRAN that are transferred to SAP. After the hours are registered, activity progress from SAP is transferred back to SAFRAN. How obvious is it for a

planner, primarily working in SAFRAN and registering working hours in SAP the same way other users do, that this type of work is dependent on MIPS?

Table 3.1 – An example of a priority list

Priority	From	To	Description	Schedule
2	SAFRAN	MIPS	Transfers information about planned activity	Every 2 hours
3	MIPS	SAP	Transfers job timesheet	Every 2 hours
4	SAP	MIPS	Transfers expended man-hours	Weekly
7	MIPS	SAFRAN	Transfers progress per activity	Weekly
38	Cabsys	MIPS	Transfers data about cable routing	Daily
44	MIPS	TIME	Updates purchase order number	Daily

Figure 3.2 depicts the system-level dependencies between the applications.

Activities from SAFRAN are first transferred to MIPS (connection with priority 2, first row in Table 3.1). From MIPS, legal codes are transferred to SAP (priority 3) where project engineering and administrative personnel register their timesheets with hours spent during the each working week. Spent hours per job code are then transferred from SAP back to MIPS (priority 4). MIPS, in turn, transfers data about progress (priority 7) on different activities back to SAFRAN, so that a customer can get weekly or monthly reports and pay for the completed work. Most likely, the dependencies between these three applications are not obvious for even advanced users.

Just like the hidden MIPS node in Figure 3.2, the introduction of an extra node can have a negative impact on the whole system. The following case study illustrates this statement.

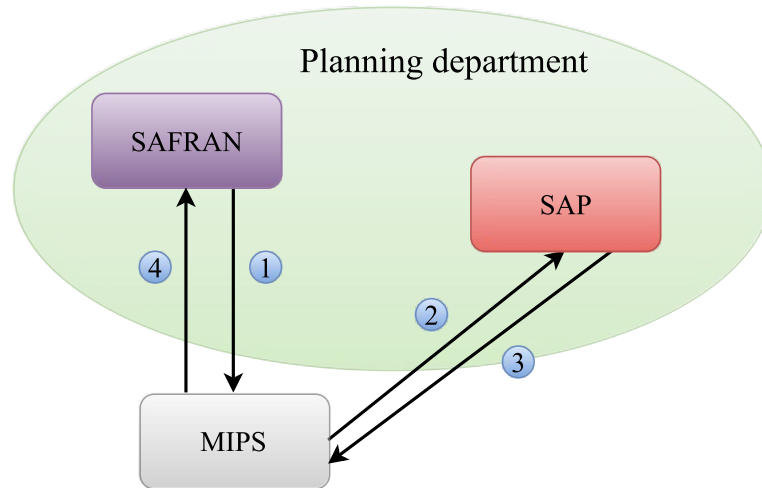


Figure 3.2 – Hidden node in communication between SAFRAN and SAP

Case study: Introducing Frames as a new node

To upgrade the drilling systems on a customer's oilrig, it was decided to use two of the Company's subdivisions located in different cities and belonging to different legal entities. The main contract was signed with a subdivision, which still used an outdated application to maintain an engineering database called Frames. Due to peculiarities of the contract, information had to be transferred from the customer's database to Frames, adding an extra node between the customer's database and our internal database TIME. Since Frames had limited performance, lacked a user-friendly interface, and did not contain essential data fields to store equipment features, which were not obligatory earlier, the employees had to use TIME (see Fig. 3.3).

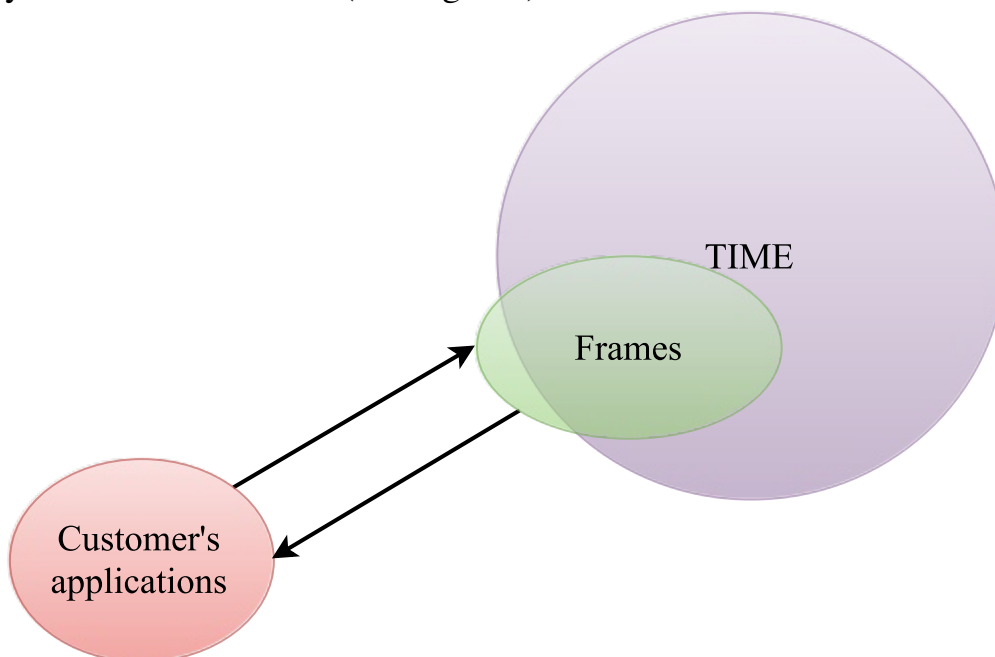


Figure 3.3 – Frames in the middle of data exchange process

Nevertheless, to work according to the contract it was necessary to exchange data between the customer and the Company via Frames. This solution led to additional risk factors and errors since it was necessary not only to perform cross checks in two independent databases, but also synchronize two internal ones.

The definition of “current data owner” became an important issue. By default, Frames owned all the data in the project meaning that Frames was the “master copy.” What did it mean in practice? If an end-user (a responsible engineer) working with TIME filled in required fields, or updated the existing values, then during the next synchronization the new data were overwritten by old data stored in Frames. Thus, important registrations were lost.

Engineering objects (e.g. equipment, cables, etc.) have status from 1 to 7 in the engineering databases depending on the degree of information completeness and the project phase. It was decided to use this functionality to determine the “owner” of the data. The point was: depending on the status, either Frames or TIME was the owner of the data and the master copy was stored in both applications when the two programs were synchronized. It was quite convenient since groups of people who used those two databases worked with the information at different stages: those using Frames processed the initial information needed to identify all engineering objects (status from 1 to 4), and those using TIME had the responsibility for completing the design and installing equipment (status from 5 to 7).

However, even this approach did not remove all inconsistencies. At the final stage of the project TIME was updated to a newer version. As it was stated by the IT department, the transition from the old version to a new one should happen smoothly and would not be noticed by personnel at all since the transition was thoroughly tested. But in our case, the database settings were imported incorrectly and the rules defining the information owner were ignored.

Thus, for several weeks all data inserted into TIME were overwritten by the old data from Frames. These discrepancies were found during the final inspection of documents packages before sending them offshore. All the engineers responsible for the packages had to reenter the changes again. Once an error was reported to the IT department, it took another week to resolve it. A significant amount of hours was spent doing the same work twice, especially taking into account the fact that a part of the overwritten data was stored only in TIME.

The costs associated with reentering data into TIME were not included in the estimated project costs; even the estimated contingency costs did not cover all

the expenses. Unnecessary and time-demanding work can easily lead to customer dissatisfaction because idle hours for already mobilized offshore personnel are very expensive.

Introducing new strongly connected nodes in a system may very well result in increased risk of system breakdown, forcing employees to re-do lost work. According to the project's risk analysis, it was very unlikely that the simultaneous use of two applications would lead to any problem, but it was also recognized that the consequences could be major in the unlikely event that something bad actually happened. The Company should avoid ad hoc solutions and carefully evaluate data flow in their systems to avoid embarrassing and costly mistakes. The risk of dataflow disruption while applications execute their functions was most probably neglected. In the particular case described above it was crucial to either solve the contractual issue with data transfer or to add missing functions to Frames in order to use it without TIME.

When TIME was updated during the final stage of the project, it was too late to implement the possible solutions mentioned above. Realizing that the existing work methods were not good enough and did not provide sufficient flexibility, predictability, and stability, a group of project members initiated an effort to improve the way common tasks were executed. Moreover, to compensate the lost hours due to synchronization problems, an effort was initiated to remove deliverables that had little or no value to the customer.

The effort began with an email to all employees involved in the project. The email invited everyone to take part in a short survey with the aim to determine ways to improve the execution of the project. Based on the survey's results the decision was made to create teams consisting of people already working inside the project. Each team got a task within an area needing improvement according to the survey results.

Each team consisted of a diverse set of people with knowledge and experience gained through work in different departments and in various roles. Employees were divided into teams based on their expertise to be able to analyze problems from different viewpoints and find the most effective and universal solution. The teams analyzed work packages, bulk purchasing, installation planning, as-built routines and more. The main goal was to save time and money by simplifying existing routines and eliminating unnecessary stages in the workflow.

The team studying work packages was able to optimize packages' lifecycle, reducing the hours it took to complete a package by 60% on average. The main

reduction was achieved by removing the time needed to print documents that then had to go through several rounds of signatures. Performing the same procedures electronically instead, the team managed to save time without lowering the quality of the final product. It also led to faster creation of electronic job cards.

It was difficult to convince some people involved in the work packages to use the new and simplified methods instead of the old ones. It is definitely hard to change people's habits when they have followed the same procedures for decades. The work of the teams in the Company to improve workflows led to the following recommendations:

- When setting up a new project, try to eliminate unnecessary nodes in the application graph because if something can go wrong it will.
- Key software should never be upgraded during the final stage of a large-scale project unless it is absolutely necessary.
- Having an IT-support group that fixes potential problems within hours is a good long-term investment.

The case study above unveils a fundamental issue: introducing additional strong dependencies increases the likelihood of unexpected failures. It is clearly important to eliminate unnecessary strong connections. As an example, earlier the Company used an application named Project Wise to maintain all the documents in the project. One had to create document numbers in TIME, but source files were stored in Project Wise. This application was also involved in transfer of data (see Fig. 3.4).

It was decided to give TIME the functionality of Project Wise and therefore remove the dependency between them. The decision made a positive impact on the system's fault-tolerance due to reduced likelihood of failed communication between the applications.

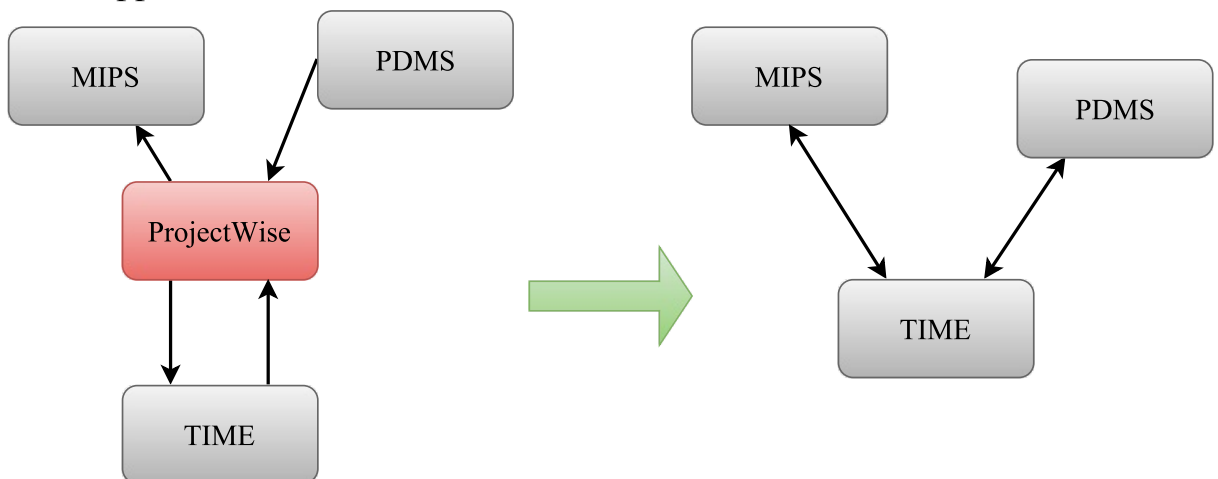


Figure 3.4 – Simplifying of information flow

3.4 Incidents having negative impact

For the system described in Chapter 2, high availability is a key requirement enabling the Company to complete tasks in a timely manner. Since the described system is not fully isolated from the surroundings, it is not only influenced by the internal communication between the components, but also by external factors. Natural disasters, accidents, or technical issues as well as human factors can trigger weaknesses in any complex system [17]. In the case of the Company's system, the seriousness of the impact will depend on the affected application(s). The following sections describe events that have been costly to the Company.

3.4.1 Downtime

An unplanned outage has a significant impact locally and can result in decreased revenue globally. Various studies and surveys show that a company loses, on average, between 84 000\$ to 108 000\$ per hour its ICT system is down [18].

There are several circumstances that determine the cost of an outage, beginning with the direct cost of countermeasures needed to eliminate the problem, employees' idle time, paid overtime to employees in the IT department and other involved departments, and a temporary need for additional hardware and software. Indirect costs, not always taken into account, include reduced customer satisfaction and additional expenses to regain customer trust.

The duration of the downtime is of crucial importance when it comes to determine the total cost. For an engineering company, an application that goes down for a couple of minutes is acceptable even if some work is lost. If the downtime increases to a few hours, then the total cost may be much higher. Important variables determining the cost are the project stage, affected departments, task urgency, and amount of personnel who work only with the affected application. A whole day or more of downtime can trigger nonfulfillment penalties from the customer and delay the payment of the ordered services or products.

Case study: Outage due to power loss in January 2015

Several natural disasters took place in Norway in the middle of January 2015. For three days, all the Company's employees in Bergen lost access to a customer's Citrix-based applications. Almost all the customer projects were affected by this loss to varying degrees. Nearly completed projects were most affected because the projects members were unable to hand over the project

deliverables (e.g. installed equipment and new software) to the Customer's systems on time. The Company's IT service published an alert stating that the situation was under investigation, but no explanation was available at the moment. No workaround instructions were given. After a couple of days without any access, the most affected employees started to call the customer's IT service. After some time, they found out that a power line to a datacenter hosting the customer's servers was broken. A power company was working hard to fix the problem, but there were several breaches on the way from the power plant and they were struggling to solve some greater issues as well.

It turned out that a workaround existed. After several calls to the IT-service department and help for IT experts, it became clear that the applications we needed were available, but on a different address and requiring a two-step authentication. Luckily, a majority of the employees could complete the authentication process because they had OTP generators. Some of the employees spread this new information to other projects and to the Company's IT service. Unfortunately, it took three days before the information was published on the Company's internal web pages.

Downtime caused by natural disasters and the consequent delays are not taken into consideration when projects are planned. In this particular example, the downtime could have been shortened significantly if different stakeholders had communicated better and there had been more emphasize on implementing easy-to-use alternatives.

3.4.2 Unavailability of error-free data

This chapter started with an effort to show the importance of having both the correct amount of information and the correct information flow. Realizing what kind of input is needed to complete a certain task is one of the main success factors. But often, due to external influence or internal interdependencies, the data quality or quantity are unsatisfactory.

A study of the system described in Chapter 2 reveals several reasons why problems may occur. Application databases inside the system are synchronized at different times. From time to time, corresponding fields in different application databases are not updated when they are supposed to because of problems with the synchronization process. As an example, if the synchronization happens once every twelve hours, a failed synchronization could mean a delay of one day. The failed synchronization can cause a database to return outdated values of important project parameters.

Incorrect or low-quality input provided by users can have significant negative impact. Wrong data may result in false confidence and increase the likelihood of future mistakes. There is no thorough control of the data entered into the databases. The same applies to customers' databases, which have only a few standardized fields with input value verification or a predefined drop-down list. The lack of input validation is an industry-wide problem. Although "note," "additional explanation," and "description" fields are filled with explanatory text, there are still many fields that can be, but are not, standardized, including cable types, vendor and model names for widely used products

Case study: "C-K" problem for cables

All cables used in a project must be registered in the engineering database TIME with a reference tag. The tag is stored in the field called "type." The applications in the system allow one to choose a type from a list (which can be really long) or search for it by name. People select the type they believe to be correct. But due to rebranding or some change in the design, the type name could be changed. A widely used family of signal cables had the letter "R" in the cable names to indicate that the insulation was made of rubber, now it has a "B" for fire-resistant halogen-free insulation, which still can be the same rubber.

By default, the type list consists of a few predefined tags. If a needed cable type is not found in the list, a new one is created by an engineer and is automatically shared with all database users. The "search" function allows the use of wildcards if one is not sure about the full name. It is possible to use "%" to represent an arbitrary number of unknown symbols or "_" for exactly one unknown symbol. In practice, engineers, especially the older ones, do not use wildcards; instead they search for a full name.

Another example is an ordinary twisted pair, where Category 6a is the most widespread type offshore. This cable type could easily be tagged as "Cat. 6a," "CAT6 a," "Kat6.a" (especially relevant because it is an abbreviation of the Norwegian word "kategori") and many others. Taking into account that the search field is case sensitive and different disciplines within the project can independently use the same cable type for different purposes, the space of possible tags becomes even larger.

In one project, four different names were created for a single type of cable. Internally, different names for the same object create inconsistencies and made an already long list of items to choose from even longer and more inconvenient to use. It got even worse when, at a later stage of the project, the Company's database

was merged with the customer's database with a fifth name for the cable type, resulting in no less than five different name variations for the same cable type.

When users input much data to the system, it is unlikely that all typos and other mistakes will be detected immediately and that failure propagation will be avoided. Erroneously entered data often have no harmful effect on the nodes inside the system and do not lead to any system crash. Personnel involved in a project only detect such errors after they have propagated. While it is hard to prevent all erroneous data from propagating to other parts of the system, we can limit such propagation. Some recommendations to limit this problem are given in Chapter 4.

3.4.3 Unauthorized access to sensitive data

While working on a project, an engineer gets access to different information on existing systems, including information about future updates and modifications. Often, this information has value only to a limited number of people involved in the project. But in some cases, the information can be misused and its disclosure can have undesirable consequences.

As described in Chapter 2, TIME serves as a database and a control version system for maintaining documentation and equipment data. The project information stored there can be of large interest to competitors. The following case study shows how the lack of attention to details when designing an applications' notification system can facilitate unauthorized access to user credentials.

Case study: TIME security

In earlier versions of the TIME application, a user had to provide a username and password to log in. The password could be very weak because there were no requirements on the length or characters of the passwords, but the password-based login still provided some security against an attacker wanting to copy sensitive information.

The author used TIME to work together with another company on a complex project. The organization of the project process was quite demanding in terms of roles. We (Company B) were responsible for engineering while representatives from another company (Company A) managed the whole engineering process and was responsible for delivering the final product to the customer (see Fig. 3.5). At the same time, another subdivision of Company A was sub-suppliers to us. Being a "man-in-the-middle" required permanent awareness with regard to information flow.

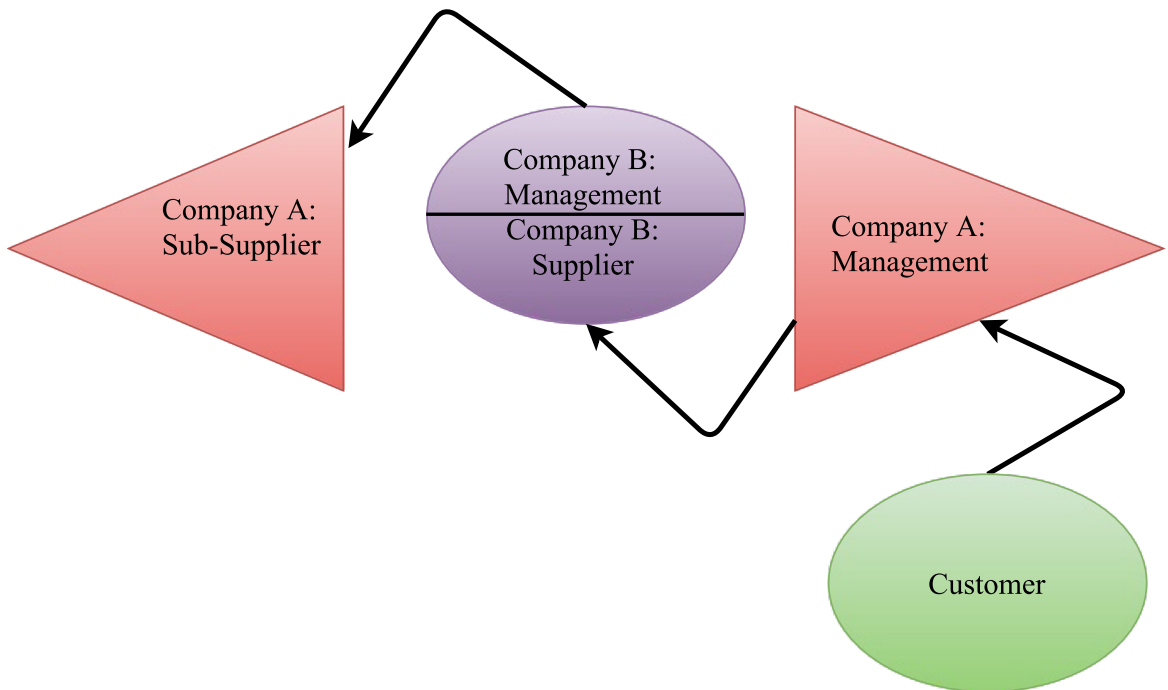


Figure 3.5 – Complex relationship between companies' roles

So back to the point, data security is important. During a certain period, TIME was quite unstable and error messages were quite a common. The messages contained both username and password in plain text (see Fig. 3.6).

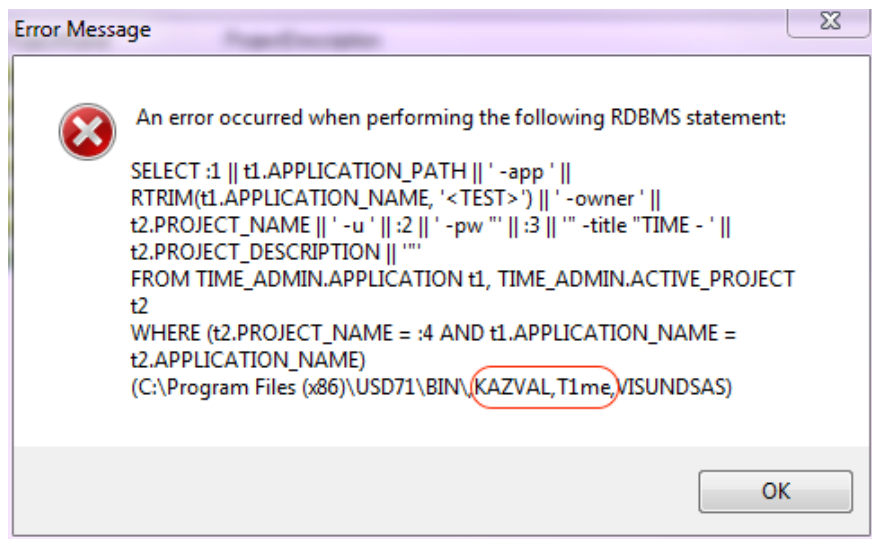


Figure 3.6 – A screenshot of an error message containing user name and password

So yes, one could start TIME and generate a known error in order to obtain credentials. The official answer on a ticket describing this security issue contained comments from the IT service stating that this was a minor issue they were not

going to fix. The reason was that according to the company's security policy, any PC must be locked every time a user is not present by its side.

In the next version of TIME, the login functionality was completely removed. Now, only Active Directory manages access to TIME.

3.5 Does a classical risk assessment cope with rare negative events?

It is compulsory to assess the risks of any new project. The level of detail and the number of team members involved in the risk assessment are determined by the size and complexity of the project. For a small project, a task manager usually performs the risk assessment, medium size deliveries imply key-discipline engineers' involvement, while a large project has its own risk manager and, in special cases, even a risk assessment team.

A common practice is to estimate the likelihood and impact of a negative event. This type of risk assessment is based on experience from similar projects and is often quite subjective. The axiom saying that personnel executing the project makes things right from the start is often accepted.

Moreover, despite the knowledge that any project implementation is highly dependent on the existing ICT system and its normal operation, IT engineers are almost never involved in the risk assessment. When no personnel with special competence are involved in a risk assessment, the items on the risk list usually have relatively high likelihood, are well understood, and have small predictable impact. This makes the results of the risk analysis look very nice on paper, but it does not help to cope with rare events with large negative impact that have not occurred before. These events tend to come as a total surprise to stakeholders.

Taleb [19] has introduced two special classes of events that are most often ignored during traditional risk assessment, namely black and gray swans. A black swan is a rare unpredictable event with a huge impact on a system and its stakeholders, while a grey swan is somewhat predictable but still hard to foresee and with huge negative impact. In retrospect, after an event has occurred, risk analysts often conclude that the event was not so hard to predict. In the case described in chapter 3, where two almost alike applications were used in parallel to maintain the engineering database, the failure was a gray swan because it was hard but not impossible to foresee the problem. However, the opinions of a few stakeholders with "extreme" views are most often ignored during a risk assessment, making gray swans black to most stakeholders in a company.

Case study: Tag reservation for new equipment.

Until 2008, the internal TIME application held all work on equipment tags (ID numbers for equipment in a customer's systems). After a project was completed the internal database was simply merged with the customer's database to reflect the final status of the delivered product. Nobody considered the possibility that the customer or other suppliers could be working with the database simultaneously and thus become affected by this merging operation. For example, the new tags used in a project could already be reserved for some other purposes. Inadvertent tag changes in completed projects due to database mergers led to considerable additional costs, including the cost of extra manpower, because all technical documents and drawings containing the affected tags had to be fixed. The customer of course did not pay for such mistakes.

After a couple of quite significant financial losses it became mandatory to reserve equipment tags in a customer's database at the beginning of a project. Therefore, a tag could not be used by another project. Furthermore, information regarding the equipment was transferred to a customer's system only after the design, equipment models, and configuration were frozen.

Although it is not that hard to detect the kinds of methodological problems described, the changes to the Company's procedures were implemented only after couple of incidents creating large extra costs. Only major financial losses seem to induce changes to existing routines in order to prevent even larger losses in future. Companies prefer to spend money on fighting the consequences instead of trying to develop preventive measures.

A risk assessment team with diverse expertise can result in better threat identification and more adequate judgments of possible consequences. Creating a team consisting of specialists with experience from different fields can make some gray swans easier to predict, especially for existing systems.

A standard 5-level risk matrix that is used to evaluate each risk separately creates an incomplete picture of the overall risk. Smart risk assessment tools that consider how one event can influence another [20], [21] may become a proper alternative in future, though they are far from perfect at the moment. But human evaluation is still central to the risk assessment process. A group of people gathers information from the system, as well as from external sources and decides what should be used as an input to a risk assessment tool (see Fig. 3.7).

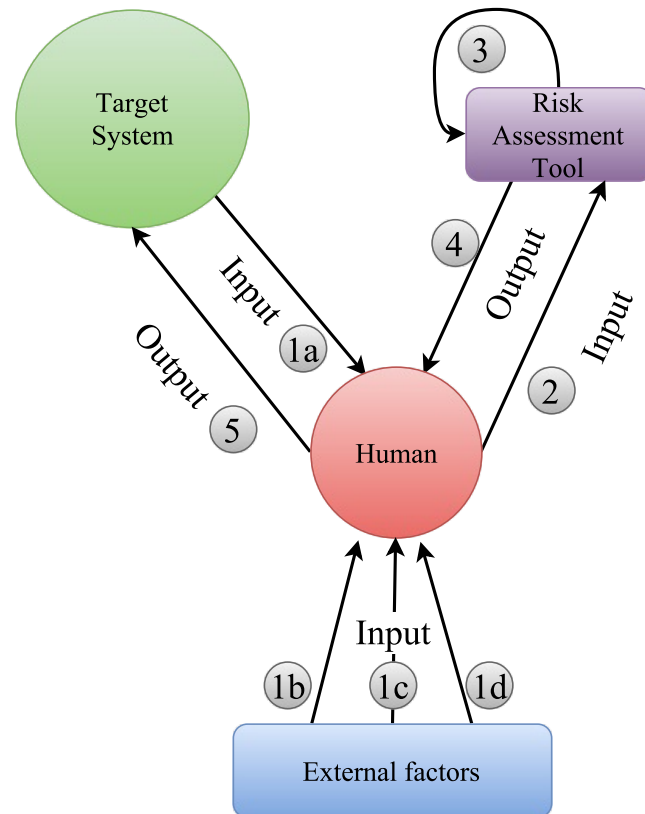


Figure 3.7 – Human-centric risk assessment approach

To develop a new ICT system, or any other man-made CAS, developers must keep in mind that it is more important to be able to cope with consequences than to accurately predict their causes. Assessing risk, a human should have an observer role and be capable of maintaining and providing tuning to the assessment system, but she should not be a decision maker (see Fig. 3.8).

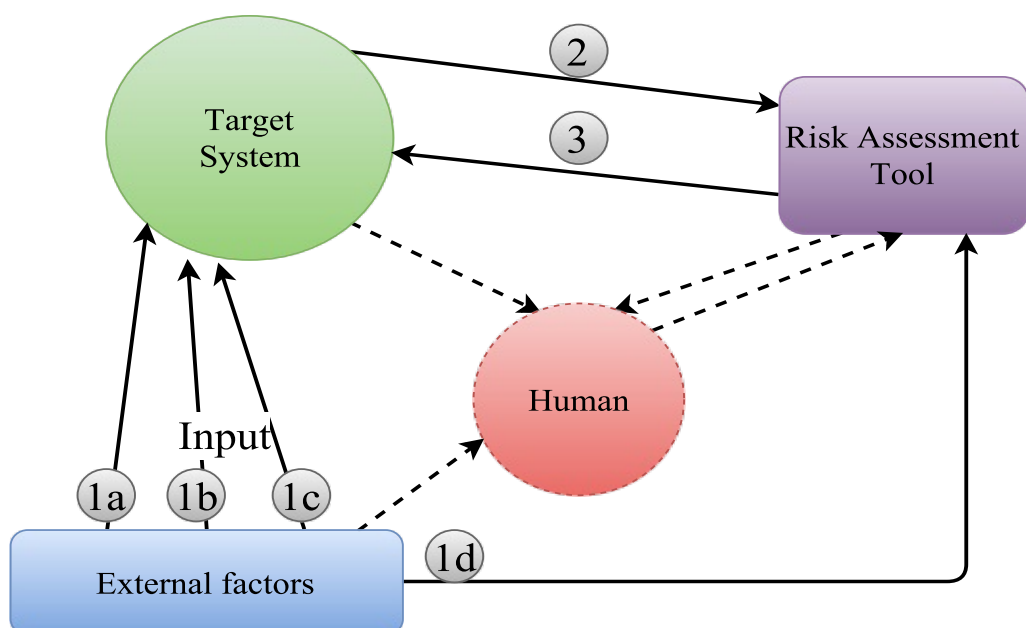


Figure 3.8 – System-centric risk assessment approach

The risk assessment tool in Fig. 3.8 periodically updates the risk analysis based on information from the system's event log. At the time of writing, many companies utilize machine-learning techniques to first detect anomalous system behavior and then trigger automated procedures to deal with the unexpected behavior. The assessment tool should be considered as a part of the whole system as it provides the system with a degree of self-learning to handle anomalous behavior.

It is crucial to understand what kinds of negative impacts are unacceptable to stakeholders of a system and then try to decrease the system's fragility to these impacts. The system should fail fast to prevent error propagation and learn from these small failures how to adapt to prevent larger failures [11].

3.6 Summary

It is necessary to include the behavior of stakeholders when modeling modern ICT solutions. It is particularly important to include stakeholders that interact with the technical system or that are responsible for its maintenance. One cannot assume that a system will always operate smoothly just because it has gone through rigorous testing and all known issues have been removed. Results obtained in a test environment do not guarantee that there are no hidden issues in the production system. In particular, the possibility of high-impact events must be taken into account when performing major upgrades of key software during a critical stage of a project.

Examples and case studies described in this chapter emphasize the importance of two fundamental problems for any system design: the inability to foresee the consequences of adding extra nodes to a system and the difficulty of detecting hidden dependencies between existing nodes. Additional nodes must operate independently of already existing nodes to avoid failure propagation. Hidden dependencies reduce the understanding of how the system operates and could lower the operational effectiveness.

4 Technical Improvements of the Company System

4.1 Redesign needed

The real-life ICT system described in Chapter 2 is the backbone of the Company's everyday operations. It is impossible to separate the system from the rest of the Company without halting most of its operations. According to a member of a pioneering development team, the ICT system's original design can be traced back to 1998. Since then the system has grown rapidly, fulfilling new and steadily stricter requirements from customers and governments. Moreover, the need to scale the operations of the ICT system increased dramatically as the Company established new offices in Norway and abroad.

The system has never been redesigned from scratch. Instead, the original functionality has been modified and new functionality has been added to the existing system. As a result, the system has become clumsy and hard to maintain. It is likely vulnerable to gray and black swans because of strong dependencies between its parts and because nobody has a complete overview of how the system operates any longer.

As the system was modified and updated during its nearly 20 year of life, the number of non-obvious dependencies between its parts has increased. Some of them were introduced unintentionally, because of insufficient system knowledge; others were the result of quick fixes or "temporary" solutions. Today, the system contains modules and interconnections that are no longer in use, adding unnecessary complexity and making it harder to determine the system's many dependencies.

The system contains legacy technologies that should be removed. In particular, the system depends on an old version of Informatica PowerCenter that periodically synchronize the workflows. If synchronization fails, the system may report erroneous values of important project parameters. It may take up to two days before an old value is updated when two workflows are involved in the operation and one of them has a failure. Today, Informatica also offers PowerCenter Real Time that ensures updates in real time, thus avoiding the long delays of the current system.

Time is of the essence for a variety of business-critical processes. Failed synchronizations and slow updates in general contribute greatly to unrealistic time estimates given to customers by the end-users. For example, in case of a simple, but highly critical task, a Company engineer knows exactly how long it takes to

perform the standard operations to execute the job. When a couple of hours become a couple of days due to slow and problematic synchronization neither the customer nor the project management are satisfied.

To summarize, a redesign is needed to reduce the time it takes to complete tasks and to increase the cost effectiveness of the system. Whether the redesign is done by the Company itself or in cooperation with other companies in the oil and gas sector, it requires significant initial investments. However, the cost of unforeseen incidents caused by an outdated system could also be substantial. It is often said that, if something can go wrong it will, and it is just a question of time in our case.

A redesign should be based on the following principles:

- **Openness** is needed to share knowledge about the system between stakeholders. When only a small isolated group of experts has all the information, it is easy to overlook a potential vulnerability. Continuous presence of diverse views is vital to correct system development. Moreover, openness helps to keep the stakeholders updated and involved in ongoing progress and decision-making. For end users, openness means availability of information needed to understand the working principles and connections between the modules of a system.
- **Trust.** To build trust, it is necessary to demonstrate that a chosen solution facilitates everyday tasks, is highly available, and provides reliable answers. Trust building measures stimulate occasional customers to become permanent ones, and encourage users to follow standard procedures instead creating their own workflows.
- **Input validation.** Input control and validation is a challenging task. More than a half of the errors in the Company system is due to erroneous input and inattention to details. When an incorrect letter in a hardware model or cable type can cause a tangible financial loss, measures should be taken to avoid these errors.
- **Access control.** Proper access control is needed to prevent unauthorized changes of critical data within a project and even avoid system downtime in some cases.
- **Diversity.** Teams of people with diverse skill sets are important to engineering projects. Especially risk assessments need diverse points of view to produce useful results. People are still the most valuable resource of any engineering company. And their peculiarities, interests, and abilities

should not be ignored but used actively every day to make the work faster, simpler, and more enjoyable.

- **Treat all users as valuable customers.** Often it is quite complicated for software developers to evaluate whether a new solution will be warmly welcomed among the users. Even if a new feature was made upon request, it could be implemented in many different ways. Making software development user-oriented can be advantageous for both users and developers. Creating pilot groups before deployment and taking feedbacks seriously are good ways to keep end-users satisfied. A happy user is a productive user. Of course, this approach imposes certain obligations on users as well. They have to be willing to collaborate and provide constructive feedback.
- **Keeping everyone busy** is another important idea. This statement does not imply a continuous heavy workload for all employees. It represents an alternative way to use assets. Let the employees work on at least two projects at different stages of readiness whenever possible. This approach leads to better utilization of resources over time. In particular, when an application goes down, the employees can work on another project, avoiding idle time.

The generality of the stated principles makes them valid for the redesign of other legacy systems as well. In the following, we discuss possible technical changes that could be made during the implementation of a redesign.

4.2 Cloud use

A cloud-based approach to ICT systems radically changes the paradigm of building infrastructures, allowing a company to create a flexible, easily customizable system that facilitates interaction between people. But like everything new, a transition to cloud-based technologies may face opposition and mistrust from traditionalists and top management of large companies who are accustomed to trust only time-verified solutions. While a customer may agree to give a contractor access to sensitive data to optimize workflow, it does not mean that the consent is automatically extended to third parties. This is more or less the case when a contractor decides to use public cloud services. While sensitive data may be encrypted during storage, it is hard to prevent a malicious cloud owner from accessing the data when it is decrypted during processing in the cloud.

Despite undeniable advantages such as robustness and even possible anti-fragility to downtime, increased stability and availability compared to traditional solutions, and effortless scalability, the fact that the software of tens and even hundreds of companies are hosted on the same hardware represents a significant security risk. Taking into account the security requirements of engineering and petroleum companies, a private cloud could be an option, but its establishment requires a significant investment of time, money, and human resources.

Using an infrastructure as a service and paying only a subscription fee for its use are two of the main advantages of a cloud. The cloud can provide additional resources within short time to satisfy a particular need or high activity level, while the same need would lead to time-consuming procurement, installation, and configuration of new hardware and software in a traditional infrastructure.

In the long term, the creation of a cloud that unifies the ICT infrastructures of the largest oil and gas companies in Norway would provide a number of advantages, such as highly customized services, availability of data at any time via the Internet, and dramatically reduced costs compared to the current costs of large in-house IT departments. In fact, experienced IT specialists from the oil and gas companies could collaborate and create brand new customized solutions running in a community cloud.

NIST defines a community cloud as an infrastructure provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations) [22]. The community members themselves or a third party could manage the cloud, which could be situated on or off the members' premises.

A community cloud requires greater financial and human investments in comparison to a public cloud, but it could be tailored to the specific needs of the participating companies. It combines benefits from both public (e.g., minimal shared infrastructure costs and pay-as-you-go pricing) and private (e.g., added privacy level, better policy compliance, and improved better resource control) cloud services. Community members can share startup and maintenance expenses, as well as engaging professionals from their existing IT departments.

Collaboration and sharing become easier when two or more contractors working on the same project distribute important information within the community cloud to all stakeholders. Moreover, since companies within the oil and gas sector have to comply with the same regulations, it is possible to implement the regulations directly in the community cloud environment to

enforce compliance for all members. Thus, companies get the opportunity to work on a custom-built platform that is formed to meet their current and future needs.

Even though a community cloud sounds promising and appropriate for a particular business sector, the choice to create a cloud must be carefully thought through. It requires serious efforts from all participants and the initial investment will only be returned after a considerable time. All participants should spend time defining clearly what services they require, how the services should be implemented and maintained, and how the companies would share the costs. Although oil and gas companies have many common needs, there are still some individual requirements, which have to be thoroughly assessed before the cloud is built.

To build a cloud-based infrastructure, one cannot just migrate existing services from old platforms. A complete re-thinking and redesign are needed to take full advantage of the properties of a community cloud. In order to make it worth the effort, a community cloud should be built on openness and simplicity principles. These principles contribute to minimizing the negative effect from unintended events, facilitate troubleshooting, and help keep all stakeholders updated and involved in a continuous process to improve the cloud services.

To conclude, a community cloud seems to be a good option to unite efforts within Norwegian oil and gas industry. The existing ICT systems are old and many man-hours and much expertise are required to maintain the systems. Furthermore, when multiple companies in a supplier-chain need to access the same data, it usually takes a lot of time to implement such access. A community cloud could be tailored to fulfill the industry's needs within a common ICT environment and increase the efficiency of each company as well as significantly improve collaboration.

4.3 Why control and validation

For many companies better control means additional management positions to ask more questions, fill out more checklists, and approve more steps of work processes. Many equate control with further bureaucratization, creating even more internal policies and regulations [23]. The resulting policy documents can become so cumbersome that even the authors are not able to read them from cover to cover, and the authors find it difficult to answer which version is the last one.

But this is not the type of control that is discussed in this chapter. The goal is to assist employees in their routine tasks and not to make them read thick folios.

In the following, we discuss steps a company should take to improve how employees communicate with the company's ICT system.

4.3.1 Access control and permissions

The Company's employees are granted access to different applications and databases, both internally and externally, to carry out their assigned tasks. A line manager or an application owner approves the access to a resource. Surprisingly, most of the granted accesses do not expire; employees have access to project data many years after a project has ended. In some cases, the employees not only have access to archived data, they actually have access to living data, reflecting current projects, major modifications of fundamental systems on oilrigs, new installations, and wide-scale demolitions.

The fact that internal employees have permission to read current data about other projects can be beneficial since it reduces time spent on searching for useful information. But such access also increases the possibility of industrial espionage. An employee that has access to a database via an application has read access to all other databases used by that application. The employee cannot make any changes to the secondary databases, but he or she could easily download sensitive information (e.g. engineering documents, hardware characteristics, and network diagrams). This policy was finally changed earlier this year. But one can imagine the amount of data possibly leaked this way.

Why be so careless about granting access to sensitive information? In a technical discussion with a customer representative, the Company's telecom team tried to convince the customer that adding real IP-addresses, both internal and external, to a highly available drawing of a network topology was a bad idea. This type of information should be restricted and available only to those working directly with network management. It gives no value to other employees, only to an adversary. However, the customer still wanted the IP addresses to be included on the drawing for maintenance purposes. A compromise was reached by excluding the drawing itself from the common database.

Case study: CIA triad in practice

As was already mentioned in Chapter 2, TIME is an application with a control version function. All official documents and drawings that are modified during a project have to be registered there. Every document has a responsible owner and at first sight it is not possible for other employees to edit or replace the document (see Fig 4.1):

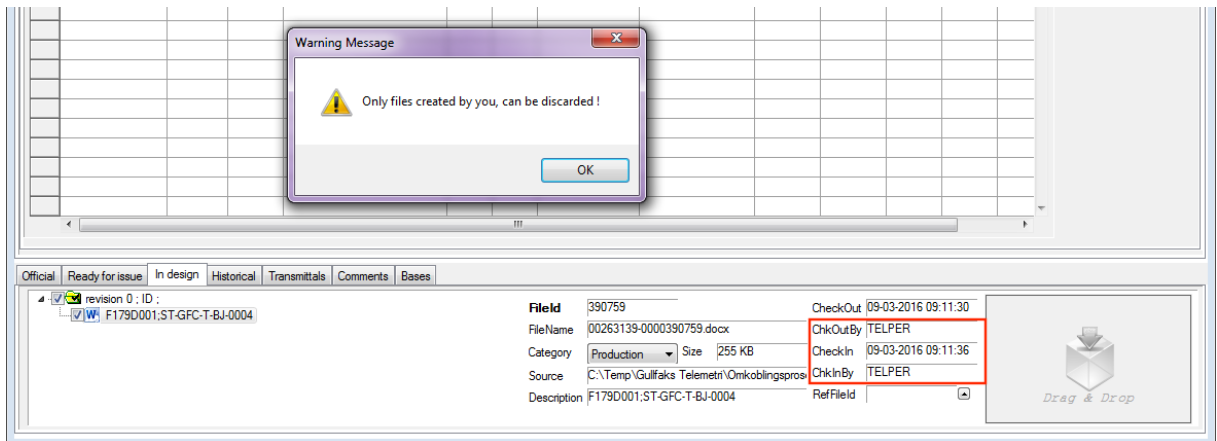


Figure 4.1 – The author's user is KAZVAL, document is owned by another user

However, one can still check it out (see Fig. 4.2):

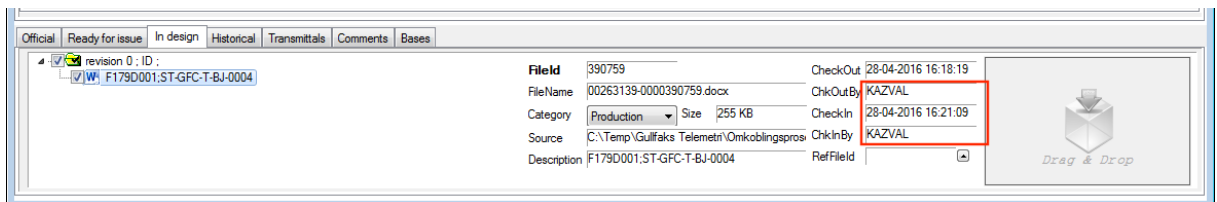


Figure 4.2 – The same document is checked out by user KAZVAL

Now the content can be edited up to full replacement, and checked in back to TIME. The fact that you simply performed the open-close operation makes you the owner of the document with full read-write permissions (see Fig.4.3):

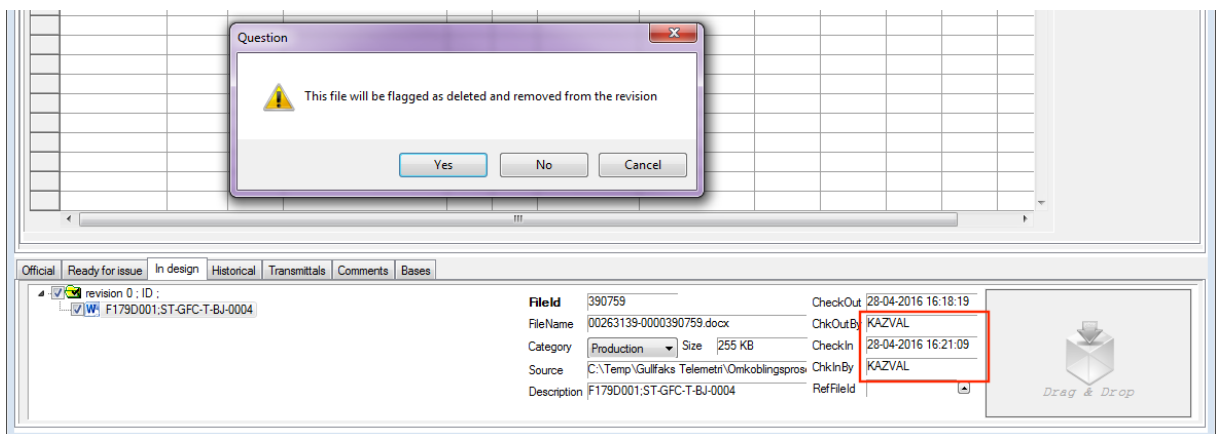


Figure 4.3 – Now the file can be deleted

The above actions were carried out to replace a document created by a colleague on vacation. But a person wanting to ruin an employee's work could easily follow the same procedures. It is unclear why the system allowed people to change documents they did not own.

The author also experienced another issue while completing a huge and complex project. All the important information about the project, including notes, pictures, and drafts that were not registered in any particular system, were saved on the project's shared area. One day all files were gone. More than 600 employees were working on the project at that time, accessing the information on the shared area on a daily basis. Plenty of calls were made to the IT service, but they only said it was not their fault, no maintenance was planned, and they were investigating the issue. After a couple of hours an update was posted on the case page. One of the employees had tried to delete her folder and had accidentally deleted the whole shared folder. It took several hours for the IT service to restore the data. So why not to control the access to a root folder preventing unnecessary idle time for much of the workforce?

Sometimes a security problem is caused by negligence or lack of understanding. There exist restricted shared areas for those working on tendered projects. The data is classified as secret and any leakage could lead to loss of the project. Some employees working on multiple projects copy restricted documents to the shared area of another project because it is more convenient for them. The Company needs to institute security awareness training to explain the potential harmful effects of such behavior.

The questions addressed in the above case study, as well as many other questions, could be answered by seeing employees as valuable resources that make mistakes from time to time. As a first step, the Company should be attentive to the specifics of the employees' everyday activities, develop tools that facilitate their work, and monitor human-machine interdependencies to identify where incidents are often happening. Moreover, working in a high-competitive industry, people must receive enough training to realize what information security really means

A security initiative "Security Starts with You" took place in the Company in 2016. All employees were obliged to take a web-based course about information security, and to pass a short test in order to complete the training. It was not necessary to take the course if you were able to answer more than 75% of questions correctly. A group of colleagues from my department decided to start with the test. We were surprised how easy the questions were. Starting from "Is it a good idea to write down your account password on a sticker and place it on your monitor?" and further to "How often do you have to change your password?" having "When your previous password expires" among the answers. The questions did not require any security education, you really only needed to pick

out the one answer that fitted the question. Our department passed the test without any training. Was the security training really valuable? Instead of creating an interesting course that could make people think more about the security decisions they make every day, a trivial course full of self-evident information was presented to the employees, just wasting their time.

4.3.2 Input control and validation

As was already discussed in Chapter 3, if we consider applications and users as modules in a CAS, then systemic failures may be due to one or more erroneous inputs to a module, or a combination of individually correct outputs from a set of modules that are input to another module not designed to handle the particular input combination. In other words, even if each module functions perfectly for all anticipated inputs, the system may still fail due to unanticipated combinations of module inputs. Furthermore, network delays and other sporadic problems can also cause systemic failures. It can be difficult to determine the underlying causes of systemic failures in a complex system, especially when the modules are tightly integrated.

The majority of the applications need input from end users. The input may take the form of freeform text, predefined menu values, or checkboxes. In some cases, whether due to incomplete information, lack of attention, or just non-intuitive interface design, inputs will be wrong. Well-designed software should be able to identify unusual or erroneous inputs, and warn the user to avoid wrong results or propagation of erroneous data to other applications [24]. Even better designed software may anticipate inaccurate inputs, recognize wrong data, recover fast, and learn from the consequences of processing erroneous data. This is why input validation is of great value.

The applications used by the Company engineers provide some input validation. Here are some examples. Character checks prevent the engineers from using a coma instead of a dot when registering numeric hardware characteristics. Range check warns about an error when trying to register a negative resistance. But there are still so many checks that can be implemented to reduce everyday errors. Today, typos and logical discrepancies may not crash the system. Most likely the system will run without any error notification. It is only when the quality of the final product does not meet expectations that some input errors are detected.

Consistency check

Consistency checks help determine internal conflicts between input values. To register new hardware in an engineering database, one assigns it a tag number

of certain type that represents the function of the hardware. A computer network cable cannot be used for high-voltage electricity transmission, and a patch panel is not a switch, even though both have 24 ports.

This type of checks could prevent guessing in case of inconsistencies in the database. Multiple attempts to figure out whether a tag type or tag description is correct when they are incompatible could become a history.

Cross-system check

Cross-system checks verify the consistency of inputs in multiple systems. Consider a task where a cable is to be installed on an oilrig. The responsible engineer determines the type and length of the cable, and registers the data in the TIME database. When all the obligatory fields are filled in and a certain completion status is given to the cable information, the data will be transferred to MIPS, where the installation time will be estimated. MIPS does not collect all the data from TIME. For example, the cable weight has to be registered manually in TIME, while MIPS fetches it from its internal table of cable types. When the cable is installed, an engineer updates tag information (e.g. information regarding the weight since an estimated and installed length often differs). The updated information is transferred to a database used by the customer. After the installation is completed, MIPS creates a weight report, to control installed or demolished weight. This report is also sent to the customer to be stored in another customer database.

The MIPS table is a source of possible trouble. The stored data may be too general or not up to date. Two cables, having the same wire number and cross-section do not always have the same weight. The weight discrepancy is of little concern when a cable is relatively short and lightweight, but imagine when hundreds of meters and kilos are involved. The discrepancy between two weight values calculated from different applications can be huge, so the customer gets inaccurate information. In future projects, it will be hard for customers to conclude what value is more precise since the customers do not know the Company's internal work routines.

Moreover, applications use different units for weight input. For cables in TIME, one has to register weight in "kg per 100 m" while MIPS wants input in "kg per km." Units are not presented in the user interface. This difference caused 250 kg to become 25 kg when the final data was transferred to a customer. Internal cross-system checks could help to avoid such errors.

Table look-up check

Look-up tables provide irreplaceable assistance to engineers. Much time is saved when engineers can choose data from a list instead of searching for necessary information externally and then enter data manually. Tables exist in the Company applications for some input parameters, but have their own challenges such as maintenance. The values in the look-up tables must be up to date, especially when the lists consist only of predefined values without the option to add new ones.

Maintaining those tables is a challenging process for the IT department, but end-users can help resolve the issue. Users should be given the opportunity to add or update a record in a table. Of course, a modified or new record must be validated to avoid possible errors. This validation need not include the IT department. Almost all applications have a group of super users that consists of experienced engineers with much subject knowledge and good IT skills. They can evaluate records and update tables. To improve the process even further, tables can be updated according to governing requirements within the petroleum industry such as NEK TS 606 or Norsok [25] [26]. When a user registers a new incorrect type, he or she should receive a warning stating that the type is not compliant with the standard.

These “according to requirements” checks are especially useful for young professionals starting their career. There is much practical information that is not part of higher education. Therefore, seeing a warning with reference to a specific standard may induce some browsing to learn more about the subject. The checks might also be handy for long-term employees. The employees might realize that standards have changed and that there is a need to update oneself as to what is the current requirements.

Input validation is a great way to minimize the negative effect of erroneous input data. Users do not always pay attention to details or they may input incompatible values to applications. Close cooperation between end-users (e.g., engineers, planners, and purchasers) and software developers are needed to ensure high-quality data input. It is frustrating for a user to get “garbage out” from an application at the end of a project, although it could be the same user who provided “garbage in” to the system [27]. When particular inputs cause repeated failures, it is important to understand the reasons. Sometimes it is not the user that causes trouble; he or she could be a recipient of problems due to invalid inputs from other users [28].

4.4 Openness

Openness is a crucial principle whether we design a new system or maintain the operation of an existing one. An ICT system is open when the information about its structure (e.g., software, hardware, interfaces, data workflow, and user interaction) is easily accessible [29]. From the anti-principle of closedness, provided by Hole in his recent book [11], openness requires stakeholders to share technical and legal information about a system. A system cannot be analyzed by a broader range of diverse specialists when only a small isolated group of experts has all the system information. Insufficient openness can lead to lack of improvements needed to maintain functionality and competitiveness. For the end-users, the openness principle means access to information needed to interact with the system. To maintain interfaces between several systems, openness is vital to ensure a correct information flow.

Case study: No description issue

A large customer awarded the Company a major EPCI (Engineering, Procurement, Construction and Installation) project. This comprehensive project included partial demolition of existing offshore equipment and installation of new equipment to comply with updated standards. According to the Company's execution model, the engineers had to register existing equipment in an internal database. It was necessary to complete the field "Description" during a particular phase of the engineering process. During the verification of this information it was noticed that descriptions were missing for most of the existing equipment. When a piece of previously existing equipment is registered in an internal system, usually a description is fetched automatically from a Customer database. In this case, many descriptions were not completed because the Customer database did not contain the necessary information. The information was missing because much of the equipment was installed during the initial platform construction when there were less strict requirements to register equipment.

It was too time-consuming to add the missing descriptions because the work had to be done manually. Furthermore, it was not even feasible to find all the missing information. After a couple of brainstorming sessions conducted by the engineering and information management departments, it was decided to automatically update all empty description fields with the standard phrase "No description in source database." This phrase had no practical meaning but helped to comply with the "non-empty" requirement. Everyone seemed happy with this solution until the next major milestone.

When the milestone “Detailed designed completed” was achieved, the information in the Company's internal database became the master copy. As a consequence, data in a customer database were overwritten by data from the Company database. In particular, all empty description fields in the customer database were updated with the text “No description in source database.” The customer did not appreciate this change and the Company’s information management team had to remove the text from the description fields.

At the next synchronization of the databases no changes occurred in the customer database. Those responsible for the interface management inside the Company asked the customer for details about their database to understand what went wrong. The customer rejected the request, referring to the nondisclosure agreement. Furthermore, the customer did not have time to help determining the cause of the problem. The customer’s unwillingness or inability to share needed information led to a straightforward but very time-consuming solution to the problem. After the final update of the customer database, employees from the Company’s information management department had to update the description fields for thousands of equipment units manually in the customer database.

The effect of system openness is not fully studied and can be harmful in circumstances where stakeholders of different systems have a conflict of interests, but openness can also be an extremely useful principle. Openness is needed to design effective interfaces between interacting systems. Lack of information about systems’ external connections is often the reason for system failures. Moreover, when a “black box” system is changed without warning, interacting systems may suffer negative consequences. It is hard for stakeholders of a system to determine the reasons for unplanned downtime and other problems because they do not have information about other interacting systems.

4.5 Summary

Before a fragile ICT system is upgraded, stakeholders should have an overview of the serious problems that can occur and be able to identify the sources of these problems. In the case of a complex ICT system, this challenge is likely to be very hard. Hence, it is desirable to create systems that are robust to incidents that we cannot predict.

The Company's system was extended and refined over two decades. Since it was never rebuilt from scratch, the complexity increased and many hidden dependencies were created. Furthermore, the use of technologies that were hard to change made it increasingly difficult to maintain and update the system. It was found that a cloud-based infrastructure could reduce the problem with legacy

technologies and provide the robustness, diversity, and ability to isolate problematic subsystems needed to create changeable applications with very high uptime. Due to stringent security requirements and trust issues, it was suggested that the Norwegian oil and gas industry should use a community cloud rather than a public cloud. While a new cloud-based system can improve information exchange, working processes, and cooperation between companies, it requires considerable investments that are difficult to obtain as long as the oil price remains low.

Less expensive options to improve the Company's system were also proposed in this chapter. Introduction of proper input validation techniques can facilitate routine tasks, provide engineers with up to date information, reduce erroneous input, and, as a consequence, increase the overall labor effectiveness. The principle of openness should be promoted inside the Company. In particular, incidents with large negative impact can be avoided by sharing information between stakeholders.

5 Human Improvements of the Company System

As discussed in Chapter 4, a redesign of the Company system is needed to keep the system up to date and cost effective. People are the driving force of any business process in the Company. This chapter provides practical advice on how to improve the employees' use of the Company's ICT system.

5.1 Keeping everyone busy

In a large engineering company, all departments contain multiple employees. All engineers in a department have the knowledge needed to solve standard tasks within their discipline. For instance, a telecom engineer can make calculations for a radio channel, provide recommendations regarding antenna placement, or create a job description of how to install field equipment like loudspeakers, control panels, and phone and data outlets.

Unlike project managers, who usually have to deal with multiple ongoing tasks at the same time, an engineer is usually assigned one task at a time. Single assignments are reasonable when they take only few hours or days to finish, or when more time consuming tasks are easy to complete. However, let us consider a task of medium difficulty and duration. A single employee cannot complete this type of task on her own; the success depends on a precise statement of the problem, realistic planning, and a consistent project team. If a team member with unique skills or responsibilities suddenly becomes sick or takes a vacation, the whole project could grind to a halt because, e.g., requests are not followed up or mandatory safety and risk analyzes are not completed. Adding a new person to the team may not be a good solution since it takes much time to explain all the features and pitfalls of a project to a new engineer.

So what if we try to involve the same engineer in a couple projects, preferably at different levels of completeness? Let one of the discipline engineers be the main responsible, while another one can keep a track on project dynamics, consuming a small part of the project's time budget, and help when needed. In other words, we create a back-up plan for human resources. This will not only make it easier to cope with the problems described above, but could also add to the engineers' experience by having them work on more projects than before. We take for granted that critical hardware and software must have backups, why not have a plan B for such a valuable resource as human expertise?

In addition, multiple simultaneous assignments could help to minimize idle time in case of application downtime. Based on a project's development level, different applications play a central part in the everyday engineering process. For instance, at the kick-off phase no special tools and applications are required, while at the global design stage a project database application is utilized during the whole working day since all equipment registrations and design-based documents are created and stored there. When the detailed design is complete, another application is used to create the installation instructions and to report on the project progress. An example based on the applications TIME and MIPS described in Chapter 2 is depicted in Figure 5.1.

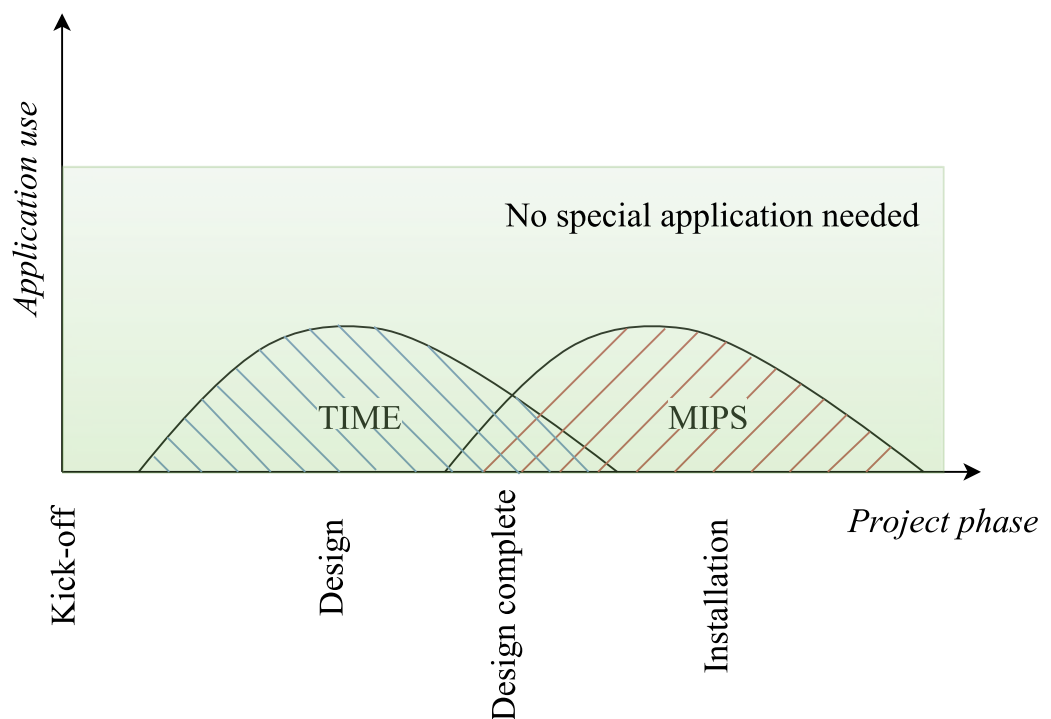


Figure 5.1 – Use of TIME and MIPS applications during different project phases

An approach like this can keep more employees busy in case of partial system downtime or unavailability for some other reasons.

Case study: Outsourcing

As a part of a major modification project, a task was outsourced to the Company's office in India. Task execution was dependent on both the information received from the main office in Norway and proper access to one of the Customer applications. The start date for the task was listed in the project plan, approved by the customer, and all needed permissions for seven Indian engineers were requested two weeks before project start. The request still had status “in progress”

a week after submission. The customer IT service informed the project team that the person responsible for granting access was on holiday, but was coming back that week, and would be notified about the importance of this request, so everything would be done in time. Friday the same week access was still not in place. Another call directly to the person responsible for granting access revealed that she had not been informed about the urgency of the request and had no available time to fix it. But she promised to start working on our case early on Monday and inform the Company when it was done. Omitting unimportant details, the access was only granted two weeks after the planned start date. No less than 560 hours were registered as idle time. The unfortunate situation was not only the result of a few coincidences, but also common practices within both organizations. The situation could have been avoided, or at least the negative impact could have been reduced, if human resources were allocated to several projects and the person responsible for providing access had a substitute.

Keeping everyone involved in various projects is a non-trivial task, demanding smart management and major changes in existing best practices. It makes management's job more complicated since additional factors and restraints have to be taken into account. However, allocation of an employee to several projects in different development phases is a way to make a company robust to idle time and financial losses caused by human "downtime." Moreover, the employee can expand her portfolio, switch between tasks to make the workday more interesting, and improve her self-management skills. The "keeping everyone busy" approach is a fine method to make a human part of the system more redundant and robust (in some cases even anti-fragile) to partial downtime. Furthermore, the additional cost is limited to the period needed to establish the practice

5.2 Diversity

The term "diversity" is familiar to all companies with an international presence. The term is widely used to refer to numerous types of differences between employees from different parts of the world. Commercial organizations, working with many types of customers and hiring all over the world, try to exploit the potential of diverse teams to increase the productivity. International expansion naturally leads to teams of employees from different cultures, with different religious backgrounds, and varying knowledge. It is vital for a company to determine how this diversity can work for the company rather than against it. This section considers how team diversity in a global engineering company can

increase productivity and provide high-quality solutions. Several examples illustrate important insights.

Considerations of team diversity cannot be started without reviewing the scientific research conducted during the last decades. According to Mannix and Neal [30], the research community is divided into two camps. The “optimistic” one is strongly convinced that team diversity has positive effects on the outcome of a project no matter the communication challenges between the team members. The reasoning is based on the belief that diversity initiates constructive discussions and leads to more creative problem solving. Most of the studies where diversity was recognized as an indisputable advantage are based on functional diversity in skills, education, and training.

Researchers from the “pessimistic” camp were primarily investigating groups with diversity in race, culture, religion, age, and social status. They believe that any beneficial effect of team diversity more or less disappears due to lack of mutual comprehension, collaboration, and integration.

Obviously, a significant share of the research in team diversity has addressed various kinds of differences jumbled together under the common name of diversity. Usually team members differ socially, demographically, functionally, and culturally; in particular, they have different personality traits. It is important to take into account many differences to create really effective diverse teams. All people are different by nature. These differences make us respond to challenges in different ways. A unique combination of characteristics makes us consider a problem from different perspectives, evaluate the problem in various ways, and see challenges and possible improvements differently. The right team of individual assets could create very good solutions to difficult problems. Nevertheless, a diverse team is not always the answer. The value of a diverse team should always be evaluated on a case-by-case basis [31].

5.2.1 Projects to practice team diversity

An employer must consider the time needed to establish a team. Whether it is a homogeneous team or a heterogeneous one, there is an initial period when people get acquainted and try to formalize in-team relations. The more diverse a team is the more time is required to set it up. This is why it is better to make a diverse team for a large long-lasting project, or at least a medium length project with a substantial initial phase, than for a small project. When enough time is available, team members can recognize people’s distinctive skills and behaviors and work out good ways to collaborate.

A diverse team is of most value when a lot of creativity is needed to solve a difficult problem. There is no need to create a diverse team to perform a simple task where a standard solution is the most effective one. On the contrary, it is an inefficient and unreasonably expensive way to use human resources. Here is an example.

Case study: Projects for outsourcing

The Company had a framework agreement with one of its customers to provide modifications of all kinds to oilrigs in the Norwegian Sea. The personnel managing the agreement were located in Bergen. To reduce costs, the Company decided to use international resources from low-cost countries for at least 40 percent of all projects. The projects covered by the framework agreement had to be managed by a project manager or an engineer in Bergen. Almost all project managers tried to outsource as much work as they possible could to get a top score for cost-effectiveness from the customer. The complexity and duration of a project were rarely taken into account.

The company received a small order to install four emergency lamps on one of the oilrigs. The project manager decided that an electrical engineer in Bergen should perform all needed calculations and determine the needed equipment and cable types. She had to follow up an engineer from India who would implement the changes to the customer systems and update the corresponding documents and drawings. After the project estimation was completed, the project manager was told that it was inefficient to divide such a small project between multiple employees. It would be possible to send the whole task directly to India. Another option was to perform the entire project using resources in Bergen. But the project manager was adamant about her decision to involve resources both from Bergen and India.

When all preparations were done and all the choices made, the data was sent from Norway to India. Weekly follow-up meetings via Skype and frequent dialogs on the communicator were needed to confirm that the work progressed and would be done on time. The bell rang when an offshore resource asked to place an order for a cable with a length of no less than 800 meters. To the question why so much cable was needed, the answer was that the length was calculated by the application for cable routing. The result was suspicious because distribution boards and junction boxes for different electrical systems are placed quite close to each other on an oilrig to minimize cable lengths. It could be an issue to find termination blocks with spare capacity, but it was unlikely that the nearest one

was almost 200 m from the future placement of the lamps. During the following investigation it was found out that the junction box used for the lamps was placed seven floors down. It was the first position on a dropdown list in the application where all new consumers had to be registered. Moreover, that junction box belonged to another electrical system, which was switched off in case of emergency. Clearly, the Indian engineer did not have the needed understanding of the oilrig's electrical systems to make sound decisions. To pull cables through seven floors meant thousands of installation hours, resulting in a price tag of a couple of millions NOK to install four lamps.

There could be several reasons for this unfortunate situation. One is team members' different perception and comprehension of a task. In Norway, things are often agreed orally during meetings or just left unsaid because they are considered obvious to the involved personnel, while in other countries everything has to be written down. Another reason is cultural differences. Indian personnel tend to ask fewer questions than Norwegians in order to not seem incompetent. This is especially noticeable when a question has to be addressed to a woman [32]. Finally, there is, most likely, a language barrier that hinders effective information exchange when employees come from different parts of the world.

All in all, the engineer in Bergen had to redo the work by herself since the allotted hours for this task were used up and the engineer from India was allocated to a new project. Moreover, extra operations had to be executed to cancel the reservation of spare capacity in the wrongly chosen junction box. Total spent hours were twice as high as the estimate, providing no credit to the project manager for her team's high productivity. Clearly, not all projects benefit from team diversity; careful thinking is needed to select projects that benefit from team members with diverse backgrounds and skill sets.

5.2.2 Team size and balance matter

While the project size could be an indicator of whether one should use a diverse team, the size of the team itself can dramatically influence the performance. To deploy a two-member team to perform a non-trivial task is rarely effective. More than two members are needed to really exploit the effects of different perceptions and creative suggestions. Furthermore, to reach consensus when there are only two people involved could be hard or even impossible.

Both the pools of potential team members and the selected teams themselves must be reasonably large to realize the potential of diversity [31]. A small team with diverse members (e.g., demographically, culturally, or functionally) is unlikely to outperform a homogeneous team. When a project

manager has no other choice but to create a very small diverse team, the manager must intervene in disputes between members with, perhaps irreconcilable, differences to ensure that the team finishes its work.

According to Page [31], under certain conditions diversity trumps ability. It is quite naive to expect that a group of art historians, diverse in all other aspects, can create a space ship. Team members have to possess a certain amount of knowledge needed to solve an assigned problem. So the balance between diversity and ability is still of great importance.

Below we study two examples of small diverse teams. The examples highlight the personality characteristics that are strongly believed to have an impact on the teams' performance.

Case study: Telemetry project

It became necessary to update the telemetry communication between three oilrigs belonging to the same customer. The task affected safety critical hardware and signals. The work had to be performed on a tight schedule because the three oilrigs had to cooperate to get the work done. Furthermore, the work had to be done while the oilrigs were starting drilling operations. Telecommunication was the primary discipline with only minor input from other disciplines. As a result, the team was made up of the following members:

- A telecommunication engineer, broad research background, automation expert, likes to work slowly, thoroughly studying every possible option, records every small detail in an electronic log on his PC.
- A telecommunication engineer, broad experience with telemetry systems, hardware expert, likes to take paper notes, productive, but pays little attention to details.
- A telecommunication engineer, young with no special experience with telemetry systems, expert in information flow and experience as a super user of a number of engineering applications, very attentive to details.

The team worked hard to figure out how to perform the project in the most efficient way. They divided the tasks such that everyone could use their strengths. While one member designed a model for the telemetry communication and gathered useful information, a second member contacted vendors and manufacturers to get quotes on needed hardware, and the third member kept a log of the work done in the Company system. The team members had short brainstorming sessions without any fixed schedule, where they tackled challenging issues. It was a productive cooperation; having different experience

and viewing an issue from different angles helped them to develop innovative solutions. Having no experience with telemetry systems in the past, the youngest team member asked many questions, sometime simple ones, to learn more about a topic. The questions helped the more experienced members revise some of their opinions that were initially considered to be established truths. Two of the team members managed to keep the third member on schedule although he did not like to work fast to deliver on time. Personal logs, as well as proper records in the Company's system, helped when one member lost his paper notes. Everyone was willing to learn from their colleagues and gained new knowledge during the project implementation. The project was completed on time and on budget.

Case study: CCR project

A major contract with one of the Company's customers was terminated under the detailed design completion phase due to unresolvable noncompliance of hardware delivered by one of the subcontractors. Six months later, the customer decided to award the Company a minor contract to modify installations in the central control room on an oilrig. This project was part of the much larger original project.

It was correctly pointed out that engineers from the original project should carry out the new and smaller project because of the knowledge they gained during the initial phase of the cancelled project and their familiarity with the originally chosen solutions. These engineers could start with the detailed design plans for the subproject right away. Only two out of the three engineers with intimate knowledge of the subproject were available at the time. Thus, a third engineer was selected from the engineers involved in the original project, but this engineer had not worked on the particular subproject. At the general meeting, the two other engineers with earlier experience from the subproject suggested another candidate with more knowledge about the subproject, but management decided that the team needed more diverse experience and practical skills since the two engineers were quite young.

The final team consisted of:

- An electrical engineer, discipline responsible from the original project, young Norwegian male, lacked experience with tools needed at the pre-installation stage.
- A telecommunication engineer, discipline responsible from the original project, the youngest team member, Ukrainian female that moved to

Norway, the least experienced, good knowledge of applications and information life cycle at the final stage of the project.

- An instrument engineer, responsible for other parts of the original project, male in his fifties, originally from Sri Lanka, twenty plus years of experience in different industries in Norway, provided training to offshore employees on how to use tools needed during the final stage of a project, former engineering lead for the instrument discipline.

To begin with the instrument engineer was awarded extra time to familiarize himself with the project while the two other engineers were finishing their current projects. During that time, the instrument engineer contacted the customer, and not his colleagues because they were “too young to know something by default.” Hundreds of mails were generated regarding issues that were already resolved and documented in the initial project’s archive shared with everyone.

When all three engineers joined the project, the engineering meetings were unproductive. The most experienced engineer had a firm opinion on every issue, even when the issue lay outside his area of competence. When someone disagreed with him, he seemed to be willing of listening to his or her view, and he even pretended to be persuaded. However, after the meeting he phoned or mailed the customer, expressing his own opinion and stating that his colleagues agreed with him.

After the instrument engineer had ignored the views and insights of the other team members many times, his colleagues talked to him about this issue, trying to find out what was the instrument engineer’s problem. The answer was indirect but quite clear: due to his diverse working experience, he knew better; young people in the beginning of their carrier, females especially, were prone to overestimate their knowledge and skills. Thus, he had to serve as a buffer between his colleagues and the customer to prevent the younger members from demonstrating their lack of experience. Since the discussion with the instrument engineer had no positive effect, the project manager had to be involved. Unfortunately, the manger did not understand the situation. Because the instrument engineer was very friendly and positive to all the employees he did not work with directly, the task manager thought that the large amount of correspondence showed the engineer’s deep involvement and hard work. The contents of the emails were not so important.

To try to solve the issue, the customer was invited to weekly engineering meetings. The customer was happy to be included and took part in constructive discussions with everyone involved. While the dialog improved, the most experienced engineer did not completely stop his practice of sending emails to the customer presenting his own views as the views of the whole team.

Another pitfall showed up at the final stage of the project, when work packages with installation descriptions had to be made and special attention had to be paid to the information life cycle, assuring that all databases were properly updated and the installation could be commissioned to the customer without any problem. The electrical engineer warned his colleagues that he had not worked on this phase of a project before and it could take him more time than expected. The telecommunication engineer offered to help both her colleagues, because she had done this type of work for the last half year and there were not so many discipline specifics to cope with. However, the instrument engineer said that he would do everything himself since he was an expert in this area and were coaching the Company's offshore resources on how to use the relevant tools.

His work proved to be a disaster. Many mistakes were made in the database for all three disciplines. When other team members tried to find out what went wrong, the instrument engineer claimed that it was definitely not his fault. However, declaring himself an expert, he should have known that the most important engineering applications logged all recent changes.

The situation became even worse when the instrument engineer left for Sri Lanka right before the installation was to start. He left a lot of unfinished work, and a significant part of his completed work had to be corrected due to many errors.

The project was psychologically demanding. The team managed to deliver on time under conditions of constant stress and some extra-long working days. Despite the adverse circumstances, the team members managed to improve existing working methods, reduce the time needed to complete certain types of small tasks, although the improvements originally aimed to compensate the unwillingness of one team member to cooperate.

To conclude, it is necessary to consider potential team members very carefully to make a successful diverse team. When creating a small team, it is crucial to avoid members that, for some reason, will not work with the other members as equals. Such team diversity usually reduces productivity, although it can lead to some innovation created in stressful circumstances. Any team must have a constructive dialog in order to achieve common goals. Moreover, a project

manager should monitor the behavior, especially the communication, of a group and listen to the feedback provided by its members.

5.2.3 How to make diversity work?

Instead of trying to determine all the reasons why a team did not work well it is better to concentrate on answering the question “How should we create a team that works well?”

Future team members must believe and experience the benefits of working in a group [33]. If a person is strongly convinced that it is better to work alone, rather than be part of a team, it is unlikely that she would become a valuable team member. People that strongly prefer to work alone could create many problems if they are put in groups where they feel uncomfortable [30]. Willingness to cooperate, being able to communicate effectively, and to engage in constructive discussions are the prerequisites that make people perform well together. A belief that teamwork increases the chances of achieving goals is a must.

It is also vital to understand the importance of building and maintaining trust between the team members, as well as between the team and the company. Diversity is what makes us to disagree with each other’s understanding of a problem, forcing us to be more creative in finding a common solution. But a certain level of trust is needed to ensure that team members are willing to state controversial opinions, ideas, and thoughts. In a company with a highly hierarchical structure, where the value of one’s opinion depends on the position one is holding, innovative ideas from an ordinary junior engineer are most likely ignored.

A company must not be afraid to break up teams that do not deliver results due to bad communication or internal conflicts. A company must create a supportive environment for smart ideas, showing that they are of a great value, encouraging employees to deliver positive results, and to express their opinion openly. All potentially good ideas will not lead to beneficial results down the line. It is tempting to say that time was lost, but negative experiences and outcomes are important parts of the learning process. A diverse team is likely to analyze a bad decision in order to create a new one, becoming robust to learned vulnerabilities. This process is an important part of learning in a CAS containing humans. In particular, decomposing and rethinking of a negative consequence make people predict future outcomes more rationally [34].

Another condition is an active learning process. A team member could gain useful knowledge from other team members with different backgrounds and experiences. This is not only true within a member’s area of expertise. Life

experience, various working methods, and analysis techniques represent valuable learning opportunities. Training in explaining an idea, as well as becoming used to strong disagreements during a discussion enable people to think more critically and deliberately about new suggestions and ideas.

Learning from mistakes and ineffectual solutions, as well as following up successful ones, and handling challenges that emerge over time create a common knowledge base and help to anticipate weaknesses before they result in large negative consequences.

Feedback from the team members after a project is completed represents another learning opportunity. Today, companies actively promote the use of social networks at work. Whether it is a commonly used network like Facebook and Google Plus, or a specially tailored solution created to fit a company's needs, all employees have their own social network profile. There could be some differences, of course, but most profiles contain information about background, education, working experience, as well as strengths, weaknesses and personal traits as assessed by the account owners themselves. Management responsible for the creation of teams can combine social network data with feedback gathered from team members to facilitate the creation of future teams. They can adjust their choices with respect to both professional and personal characteristics that are of great consequence to specific projects. For instance, employees that work methodically, unhurried and are attentive to details could be combined with employees that work faster but pay less attention to details. The right mix of these two types of employees can create a team that delivers thoroughly validated solutions on time.

While creating a diverse team that performs well together is a difficult task, maintaining an organization consisting of many diverse teams is even more difficult. To facilitate the teams' healthy development, the teams must not be isolated. Members of diverse teams establish different workplace relationships outside their groups. These connections are a source of new information about the environment of projects under development, including technical novelties and new solutions to engineering problems. The teams can use the information to increase their competitiveness and productivity. The teams should also take advantage of the knowledge of new employees and invite external experts to move projects forward. Working on the same problem for a long time could make people think similarly and they can get stuck more easily. So a fresh look at a problem is always a great way to kindle an unexpected idea leading to a good solution.

During a traditional risk analysis of a project, an analyst allocates a likelihood and impact level to each incident on a list of possible future events created by the project management. Sometimes financial specialists also contribute incidents to the list since businesses can be badly affected by large financial losses. It might be a good idea to include engineers, planners, and purchasers in a risk assessment team to make the list of possible incidents more complete and to consider consequences more adequately. Such a team has the diverse knowledge needed to see multi-discipline dependencies. While an economist would search for the lowest priced offer from the vendors to reduce the risk of a budget overrun, an engineer is able to notice hidden design discrepancies. Working as a team, the engineers can find reasonable trade-offs, where the huge impacts of certain rare incidents, i.e. grey swans, are reduced.

To summarize, team diversity is a two-edged sword that should be handled very carefully. While big companies write about the diversity in their organization [35], emphasizing its demographical component (i.e. nationality, age, and gender) it is not the only component that makes diversity important. Differences in thinking, ability, vision, and attitude are what make diversity indispensable to engineering teams and other teams that solve problems. It is interesting to know that it is possible to create an award winning team by combining multiple smart but quite ordinary teams [36].

5.3 End-user as a customer

New business goals or requirements due to government regulations often necessitate additional application functionality. The goal is to implement the functionality within the allotted budget and time frame. One of the important stages in the software life cycle is the design process, in which end-user experience is frequently ignored. There is also a mandatory stage of functional and non-functional testing, searching for possible flaws in the design, verifying software performance and security, and checking for correct business workflows. There are many talented software developers, designers, and testers, but creating a complex user interface with specific functions, even if those were described in detail in the use cases, requires end-user involvement. The user behavior need not be identical to an application's real workflow. To design a great user interface requires a deep understanding of the actual work done by users. Johnson [37] describes different methods of how to unite design rules and human perception to design a user interface with an adequate level of usability.

Seeing end-users as internal customers of an in-house IT department can have a positive effect on users' everyday performance. Due to the way software is developed, it could be hard to involve diverse user groups at every design stage. However, conducting surveys prior to introducing new features in an upcoming application release does not require much effort. Such an approach could dramatically decrease the amount of negative feedback and enhance the user experience.

Case study: New TIME release

To complete their work, many user groups (i.e. engineers within different disciplines, planners, purchasers, task managers, and document controllers) in the Company rely on specific functionality in the application TIME. Almost all user groups would like TIME to support better filtering operations. Many requests were sent to the application owner, who interacts with the end-users and delivers feedback to the development team. After a long wait for enhanced filtering possibilities, curious users finally got to read the release note "Records filtering improved." The only visible difference between the old and new interfaces was a colorful indicator above the results window with gradation marking from 1 to 9 and the "running man" button (Fig. 5.2).

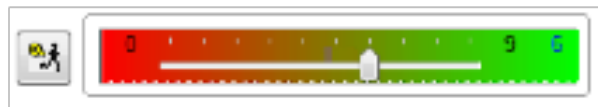


Figure 5.2 – Added filter in new TIME version

None of the users managed to utilize this new feature as described in the "how to" guidelines written by the application owner. In fact, even the application owner was unable to carry out the operations described in the guideline. After trying undocumented ways to use the new filtering options, it was decided to send a request to the developers asking for an explanation. The first answer was just a reference to the supporting documentation, which we had already read. All succeeding answers just emphasized the simplicity of the feature and the lack of time to investigate the issue due to a tightly scheduled project for an external customer. After approximately two months of correspondence between the end-users and the developers, the bug was finally found. A part of the source code supporting the functionality of the new filter function was accidentally left out in

final release version. The development team promised to add the functionality as soon as they had time available.

The described situation happened more than two years ago. The so-called filter is still a nonfunctioning decoration of the interface of TIME. This situation is abnormal and is clearly based on a human error or lack of quality control. But it could have been easily avoided by involving the users even at the final stage of testing. For those working with the TIME application every day, the only bright spot in the interface would definitely have aroused their curiosity during a test phase.

The main challenge in one of the Company's projects was to design a human-machine interface. The developers and human factor specialists were interacting with the end-users at every design stage. The same attitude to the Company's other internal projects and iterative design processes could minimize mutual frustration and increase users' satisfaction and productivity.

The Company has learned a lesson from its many projects. Before brand-new releases or transitions between software versions, a pilot user group is formed. The group members use an application for a couple of months, report possible bugs straight away, and answer a detailed survey at the end of the testing period. A group is supposed to consist of randomly chosen employees, working in different departments and offices all over the world, to provide diversity and to test non-identical user needs. Unfortunately, the same colleagues from Bergen have been included in every pilot group since the testing method was first introduced.

5.4 Summary

Despite continuous improvements of various technologies to detect anomalies and make decisions, in particular machine learning techniques, people are still needed to make important decisions. Any successful company must create an environment that not only value human knowledge, but also ensure that the knowledge is used to further the development of the company and improve on its products.

This chapter has argued that engineers should be involved in several projects at the same time, not only to improve their skill set but also to avoid idle time when an important software application goes down. Furthermore, end-user involvement during different stages of the software development process increases the level of end-product satisfaction and facilitates the creation of a suitable user interface with the right set of features. Finally, end-users should

be regarded as customers by software developers. Effective ways of communication must be found since the software developers need correct feedback from the users to improve an application.

The chapter also discussed the important question of how to pick members of a team, especially an engineering team. It is proposed several principles for creating teams, that not only get the work done, but do it in effective way, learning from previous experience and from each other.

6 Conclusions

In a company with a hierarchical leadership structure, top level management is responsible for global problems such as low work efficiency, delivery delays, and lack of communication between departments. To mitigate such problems and improve the company's competitiveness, management often initiate major changes to its software systems without knowing all the contributors to these global problems. That is why it is important to involve employees at all levels of a company as early as possible when making changes to important software systems.

This thesis has provided a bottom-up view of an engineering company's large distributed software system and associated everyday work routines. The complexity of the company's system has grown since it was first designed nearly 20 years ago. The main reason for the current system's large complexity is the tight integration of its many applications. Moreover, old unused functionality is not removed, making the system even more difficult to understand. Today, it is hard for the employees to keep track on all the system changes and few, if any, of the employees have a complete overview of the system.

The conducted analysis showed that dependencies between the system's applications, as well as dependencies between the users and the applications are often hidden and reveal themselves only in challenging situations. Insufficient understanding of these dependencies has led to unwanted user actions, delayed work flow, and suboptimal solutions.

Risk assessments inside the company were performed in a rather old-fashioned way. The assessments tended to examine system modules separately and did not take into account their mutual influences. Considering software systems as complex adaptive systems provides an alternative way to analyze systems and remove fragility to undesirable events. No matter how low the probability of negative event is, the incident always happens at an inappropriate moment, and no one is ready to face its aftermath. Hence, it is important to make systems robust to consequences of incidents and to learn from the incidents how to improve the systems over time.

A large part of the thesis has argued that humans constitute an inseparable part of any complex human-machine system. The many interactions between stakeholders and the technical part of a complex system has a huge influence on its global behavior and must be taken into account when modelling the system.

Many improvements were proposed, concerning both the technical part and the human part of the company's system. Some of the improvements can be adopted without major costs, while others need significant investment of resources. Simplicity and openness are needed to identify why modules of a system fail and to prevent these local failures from spreading over the whole system. Team diversity can be an important source of bright ideas to improve systems. Close cooperation between software developers and end-users is beneficial to both parts, and a smart input validation can dramatically reduce erroneous entries. Finally, evaluating systems from different perspectives, rethinking existing work processes, and improving the internal culture of a company can make a huge difference to the quality of the final products.

7 Bibliography

- [1] "Statistisk sentralburå," 2016. [Online]. Available: <http://ssb.no/en/nasjonalregnskap-og-konjunkturer/statistikker/nr/aar/2016-05-12?fane=tabell&sort=nummer&tabell=265613>.
- [2] R. Moriarty, K. O'Connell, N. Smit, A. Noronha and J. Barbier, "A New Reality for Oil & Gas: Complex Market Dynamics Create Urgent Need for Digital Transformation," Cisco Systems, 2015.
- [3] L. Fedulova, "Organizational Management," Lybid, Kyiv, 2004.
- [4] S. Alter, "Defining Information Systems as Work Systems: Implications for the IS Field," *European Journal of Information Systems*, vol. 17(5), pp. 448-469, 2008.
- [5] S. Page, *Diversity and Complexity*, Princeton University Press, 2010.
- [6] H. Simon, "The Architecture of Complexity," *Proceedings of the American Philosophical Society*, vol. 106, no. 6, pp. 467-482, 1962.
- [7] S. Lloyd, "Measures of complexity: a nonexhaustive list," *IEEE Control Systems*, vol. 21, no. 4, pp. 7-8, 2001.
- [8] M. Mitchell, *Complexity: A Guided Tour*, Oxford University Press, 2009.
- [9] D. Helbing, "Systemic Risks in Society and Economics,," *SFI WORKING PAPER*, vol. 12, no. 044, 2009.
- [10] N. Taleb, *Antifragile: Things That Gain from Disorder*, Random House, 2012.
- [11] K. Hole, *Anti-fragile ICT Systems*, Springer Open, 2016.
- [12] "Informatica," [Online]. Available: <http://www.informatica.com/>.
- [13] "SAP," [Online]. Available: <http://go.sap.com/index.html>.
- [14] "Safran Software Solutions," [Online]. Available: <http://www.safran.com>.
- [15] M. Stone, "Information Visualization: Challenge for the Humanities," [Online]. Available: https://www.clir.org/pubs/resources/promoting-digital-scholarship-ii-clir-neh/stone11_11.pdf.

- [16] D. Volkov, "Happiness is in Money," *Buisness Journal*, vol. 2, pp. 81-85, 2011.
- [17] W. Kroger, E. Zio and e. al., *Vulnerable Systems*, Springer, 2011.
- [18] "Downtime, Outgages and Failures - Understanding Their True Cost," [Online]. Available: <http://www.evolver.com/blog/downtime-outages-and-failures-understanding-their-true-costs.html>.
- [19] N. Taleb, *The Black Swan: The Impact of the Highly Improbable*, Random House, 2007.
- [20] "Når tradisjonell risikohåndtering ikke strekker til," 2016. [Online]. Available: <https://hackcon.org/uploads/222/03%20-%20HackCon%20-%20Risikoh%C3%A5ndtering.pdf>.
- [21] Ø. Rideng, "Dynamisk risikostyring i petroleumssektoren," 2013. [Online]. Available: http://heim.ifi.uio.no/~ketils/kst/Seminars/20131107-HvordanAggregereRisikoOgRisikoanalyser/Foredrag/131107_rideng.pdf. [Accessed 2016].
- [22] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication 800-145, September 2011. [Online]. Available: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>.
- [23] S. Dekker, "Employees: A Problem to Control or Solution to Harness?," *Professional Safety*, August 2014.
- [24] S. MacKenzie, "Input Validation," [Online]. Available: <http://www.cse.yorku.ca/~mack/1011/InputValidation.PDF>.
- [25] NORSK ELEKTROTEKNISK KOMITE, "Standard.no," 2016. [Online]. Available: <https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=796600>.
- [26] 2010. [Online]. Available: <http://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=443200> .
- [27] D. Winterfeldt and T. Husted, "Validating User Input," in *Chapters on Tiles and Validation*, 2002.
- [28] S. Dekker, *The Field Guide to Understanding Human Error*, Aldershot, U.K.: Ashgate Publishing Co., 2006.

- [29] F.Ruzic, "Openness of Information-Communications Systems: The Rescue Tool for Preserving Information Age Heritage," *Informing Science: the International Journal of an Emerging Transdiscipline*, vol. 16, 2013.
- [30] E. Mannix and M. Neale, "What Differences Make a Difference?," *American Psychological Society*, vol. 6, no. 2, pp. 31-55, 2005.
- [31] S. Page, "Making the Difference: Applying a Logic of Diversity," *Academy of Management Perspectives*, November 2007.
- [32] A. Woszczyński, P. Dembla and H. Zafar, "Gender-based Differences in Culture in the Indian IT Workplace," *International Journal of Information Management*, 2016.
- [33] R. Ely and D. Thomas, "Cultural Diversity at Work: The Moderating Effects of Work Group Perspectives on Diversity," *Administrative Science Quarterly*, vol. 46, pp. 229-273, 2001.
- [34] T. Wilson, J. Meyers and D. Gilbert, "Lessons From the Past: Do People Learn From Experience That Emotional Reactions Are Short-Lived?," *Personality and Social Psychology Bulletin*, vol. 27, no. 12, pp. 1648-1661, December 2001.
- [35] "Apple: Diversity," [Online]. Available: <http://www.apple.com/diversity/>.
- [36] E. Buskirk, "Bellkors Pragmatic Chaos Wins 1 Million Netflix Prize," 2009. [Online]. Available: <http://www.wired.com/2009/09/bellkors-pragmatic-chaos-wins-1-million-netflix-prize/>.
- [37] J. Johnson, *Designing with the Mind in Mind*, Morgan Kaufmann Publishers, 2010.