CORE Metadata, citation and similar papers at core.ac.uk
Provided by Aaltodoc Publication Archive

Aalto University
School of Electical Engineering
Department of Automation and Systems Technology

Timo Schwarte

Evaluation and development of enterprise software deployment and change management processes

Master's Thesis
Espoo, September 9, 2013

Supervisor:	Docent Ilkka Seilonen, Aalto University
Advisor:	Olli Laine M.Sc. (Tech.)

Aalto University
School of Electical Engineering
Degree Programme in Automation and Systems Technology

ABSTRACT OF MASTER'S THESIS

Author: Timo Schwarte

Title:
Evaluation and development of enterprise software deployment and change management processes

Date:	September 9, 2013	**Pages:**	55 + 2
Major:	Information and Computer Systems in Automation	**Code:**	AS-116
Supervisor:	Docent Ilkka Seilonen		
Advisor:	Olli Laine M.Sc. (Tech.)		

Large enterprises are becoming increasingly dependent on complex information systems, as they play a vital role in the business of most companies. With ever increasing complexity of deployed information systems, management of these systems becomes equally more complex. IT departments face the challenges related to maintaining the often business critical software and information system deployments.

Maintenance and management of enterprise software deployments always includes the management of changes to existing deployments as well as the management of new deployments to environments which already serve users. In both cases, it is crucial to ensure that the tasks related to the management of a service do not disrupt the business, and thus do not cause harm to the company.

To be able to ensure reliable, failure-free operation of critical systems, structured, well planned and well implemented processes are needed for managing changes to business critical enterprise software deployments.

This thesis consists of two parts; first a literature review is conducted on IT service management standards and best practices, second the deployment and change management processes implemented at the case company are described and evaluated.

The result of this thesis is a comprehensive description of the management processes related to a specific IT service at the case company. Such description did not previously exist related to this specific service. Additionally some improvement opportunities were identified.

Keywords: enterprise software, software process, deployment process, change management, ITIL, process standardization

Language: English

Aalto-yliopisto
Sähkötekniikan korkeakoulu
Automaatio- ja systeemitekniikan koulutusohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Timo Schwarte		
Työn nimi: Yritysohjelmistojen käyttöönotto- ja muutoshallintaprosessien arviointi ja kehitys			
Päiväys:	9. syyskuuta 2013	**Sivumäärä:**	55 + 2
Pääaine:	Automaation tietotekniikka	**Koodi:**	AS-116
Valvoja:	Dosentti Ilkka Seilonen		
Ohjaaja:	Diplomi-insinööri Olli Laine		

Suuret yritykset ovat tänä päivänä yhä enenevässä määrin riippuvaisia monimutkaisista tietojärjestelmistä, jotka ovat avainasemassa monen yrityksen liiketoiminnassa. Järjestelmien kompleksisuuden kasvun myötä, myös näiden järjestelmien ylläpito on muuttunut monimutkaisemmaksi. IT-osastot kohtaavat päivittäin monimutkaisien järjestelmien ylläpitoon liittyviä haasteita.

Yritysten tietojärjestelmien ylläpito ja hallinta sisältää aina niin muutosten hallintaa olemassa oleviin sovelluksiin, kuin myös uusien sovellusten käyttöönoton mahdollisesti kompleksisessa ympäristössä. Kummassakin tapauksessa on ensiarvoisen tärkeää varmistaa, että palvelun hallintaan liittyvät tehtävät eivät häiritse yrityksen liiketoimintaa.

Liiketoimintakriittisten järjestelmien luotettavan, häiriöttömän toiminnan varmistamiseksi tarvitaan strukturoituja, hyvin suunniteltuja ja hyvin jalkautettuja prosesseja palvelun muutoshallintaan.

Tämä tutkimus koostuu kahdesta osasta; ensimmäisessä perehdytään alan parhaisiin käytäntöihin ja standardeihin kirjallisuudesta, jälkimmäisessä kuvataan ja arvioidaan tapausyrityksen käyttöönotto- ja muutoshallintaprosesseja.

Työn tuloksena saatiin kattava kuvaus tutkimuksen kohteena olleeseen palveluun liittyvistä hallintaprosesseista. Vastaavaa kattavaa kuvausta ei tutkimuksen kohteena olleesta palvelusta ollut aikaisemmin saatavilla. Lisäksi työssä tunnistettiin mahdollisuuksia tehostaa tutkimuksen kohteena olleita prosesseja.

Avainsanat:	yritysohjelmistot, ohjelmistoprosessit, käyttöönottoprosessit, muutoshallinta, ITIL, prosessistandardointi
Kieli:	Englanti

Acknowledgements

This thesis marks the end of a long journey through the educational systems of two countries. I would like to thank my employer, HiQ Finland, and especially my advisor Olli Laine for making it possible to write this thesis. I would also like to thank everyone whom I had the pleasure to work with during this assignment at ABB. Additionally I would like to thank my supervisor Dr. Ilkka Seilonen for providing guidance during the whole process. I am also very grateful for the support from my parents throughout my educational career. Last but not least, I'd like to thank Minttu Koiranen, who, by the means of bribery, blackmail and intimidation but also care and support, played a vital role in getting this thesis done.

Thank you!

Tämä diplomityö merkitsee pitkän matkan päättymistä läpi kahden maan koulujärjestelmien. Haluan tässä yhteydessä kiittää työnantajaani, HiQ Finlandia, sekä erityisesti diplomityöni ohjaajaa Olli Lainetta tämän työn mahdollistamisesta. Haluaisin myös kiittää kaikkia niitä ABB:n työntekijöitä, joiden kanssa minulla oli ilo työskennellä tämän työn aikana. Kiitokseni kuuluvat myös valvojalleni TkT Ilkka Seiloselle, joka tarjosi opastusta läpi diplomityöprosessin. Olen myös erittäin kiitollinen vanhemmilleni, jotka tukivat minua koko koulu-urani ajan. Viimeisimpänä muttei vähäisimpänä haluan kiittää Minttu Koirasta, joka oli lahjonnan, kiristyksen ja uhkailun, mutta myös välittämisen ja antamansa tuen, kautta merkittävässä roolissa tämän työn valmistumisessa.

Kiitos!

Diese Diplomarbeit bedeutet für mich das Ende einer langen Reise durch die Bildunssysteme zweier Länder. Ich möchte mich an dieser Stelle herzlich bei meinem Arbeitgeber HiQ Finland und besonders bei Olli Laine für das Ermöglichen dieser Arbeit bedanken. Ich möchte mich auch herzlichst bei allen Mitarbeitern von ABB mit denen ich die Freude hatte zusammenarbeiten zu dürfen bedanken. Dank geht auch an Dr. Ilkka Seilonen für die Leitung durch den Prozess dieser Diplomarbeit. Besonderer Dank geht an meine Eltern für ihre Unterstützung während meiner gesamten schulischen Karriere. Nicht zuletzt möchte ich mich auch bei Minttu Koiranen bedanken, da Sie mit Hilfe von Bestechung, Erpressung und Drohungen aber auch durch Unterstützung und Führsorge nicht unerheblich am fertig werden dieser Diplomarbeit beteiligt war.

Danke!

Ez a diplomamunka egy hosszú, két ország iskolarendszerén át haladó útnak a végét jelenti. Ebből az alkalomból meg szeretném köszönni munkaadómnak, HiQ Finlandnak, és különösen Olli Lainenak hogy ezt a diplomamunkát lehetővé tették. Köszönetem jár mindekinek ABB-nél is akikkel együtt dolgozhattam e munka folyamán. Sok köszönetem jár Dr. Ilkka Seilonennak is, aki tanácsaival segített a diplomamunka egész folyamatában. Nagyon hálás vagyok szüleimnek is, akik az egész tanulói karrieremben támogattak. Végül, de nem utolsó sorban, nagy köszönetem Minttu Koiranennak jár, aki vesztegetést, zsarolást és fenyegetést használva, de törődést és támogatást is nyújtva, nem kis mértékben járult hozzá e diplomamunka elkészüléséhez.

Köszönöm!

Espoo, September 9, 2013

Timo Schwarte

Abbreviations and Acronyms

CMDB	Change Management Database
CI	Configuration Item
RFC	Request for Change
BU	Business Unit
LBU	Local Business Unit
API	Application Programming Interface
REST	Representational State Transfer
ITIL	Information Technology Infrastructure Library
SOX	Sarbanes–Oxley Act

Contents

1 Introduction

1.1 Background

It is a widely accepted fact that software projects often fail to deliver as expected. The exact failure rate for software projects varies, depending on the study and the exact definition of failure. A renowned study by the Standish Group, called the Chaos Report, claims a software project cancellation rate of 15% – 40% and a rate of around 50% for "challenged" projects, meaning projects which fail to deliver expected functionality, on schedule, on budget, or any combination thereof (El Emam 2008, Jones 2008, Taylor 2004).

These high figures have been questioned, among others by Robert L. Glass (Glass 2005), who implies that the results of the Standish report are questionable to say the least. Nevertheless, software projects tend to be challenging for a plethora of reasons. A study by Dr. John McManus and Prof. Trevor Wood-Harper states that management issues cause 65% and technical issues 35% of software project failures. In their study, McManus and Wood-Harper find following management and technical factors causing project failure (McManus, Wood-Harper 2007) :

Management causal factors:

- Poor leadership in project delivery
- Poor stakeholder communication
- Poor competencies (and skill shortages)
- Poor stakeholder management
- Poor estimation methods
- Poor risk management
- Insufficient management support

Technical causal factors:

- Inappropriate and ill-defined software requirements
- Inappropriate technical designs

- Inappropriate development tools
- Inappropriate user documentation
- Poor test planning
- Poor technical support

Mark Keil et. al. in turn tried to implement a framework for identifying software project risks (Keil 1998). They found that the three most important risk factors are lack of top management commitment, failure to gain user commitment and misunderstanding the requirements of the project.

The purpose of this study is to focus on customized enterprise software which is developed based on the specific business requirements of the customer. More specifically, this study investigates software deployment processes on the example of a Microsoft SharePoint 2010 environment used for hosting customized business applications which are often integrated into third enterprise information systems, such as Enterprise Resource Planning, Manufacturing Execution, Document Management and Business Intelligence systems.

At the case company several of the issues listed above have been identified to frequently cause challenges in development and deployment projects of customized enterprise software. Standardized processes have been implemented to prevent some of above issues, mainly those related to technical requirements and restrictions of the hosting environment, requirements for the technical design of the software as well as issues in communication and management of the stakeholders (vendor, service provider, customer). The implemented processes do not take into account any issues regarding functionality requirements of the customized software, test planning, end user documentation and training or maintainability of the software, even though these factors might greatly affect the success of a software project. In other words, the purpose of the process is to ensure that

- the customized software can be deployed on the given application hosting platform,
- the customized software will not cause defects in previously deployed software or the hosting platform itself
- the technical design of the software is reasonable and maintainable

- the customized software is documented well enough for the purposes of the service provider.

The processes do not take into account if the software provided by the vendor does actually meet the functionality requirements of the customer. Ensuring all required functionality (and required performance) is left to the responsibility of the customer and the vendor.

1.2 Case environment

This study is conducted in the environment of ABB Finland, a large, international industrial company working mainly with power and automation technologies. ABB Finland consists of five main divisions:

- Power Products
- Power Systems
- Discrete Automation and Motion
- Low Voltage Products
- Process Automation

Each division again consists of multiple business units (BU). The organizational hierarchy of ABB is represented in Figure 1. Business units are represented locally by the local business units (LBU). Each business unit can have local business units in multiple countries. Each local business unit has its own Information Systems (IS) organization which, to some extent, can independently make decisions and purchases. This adds to the complexity of managing services shared between LBUs, such as the SharePoint service which serves as the case scenario for this thesis.

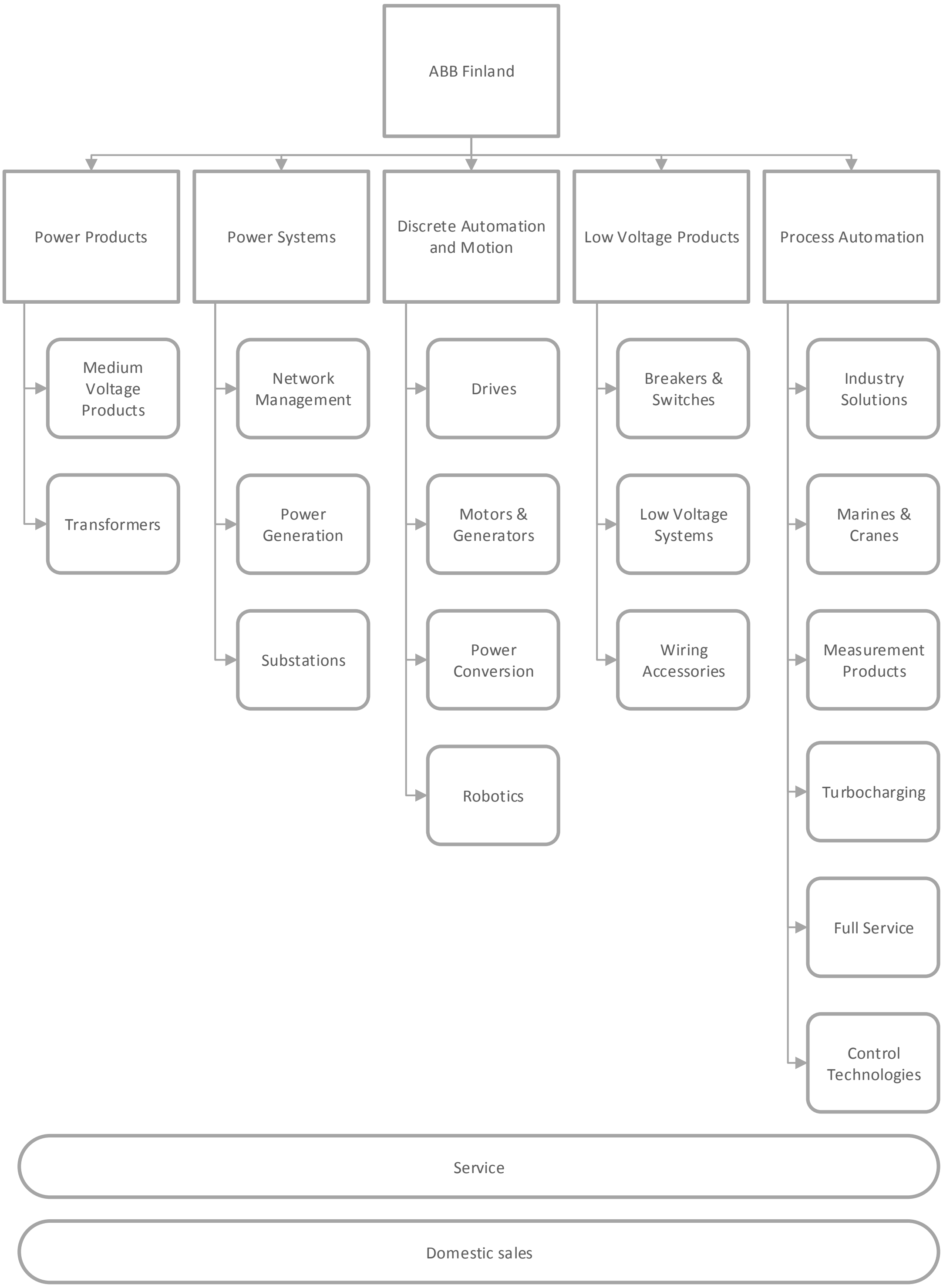

Figure 1 ABB organizational hierarchy

The author of this thesis works for HiQ Finland, an IT and management consulting firm specializing in communciations, software development and business-critical IT. This study has been conducted by the author in an assignment by ABB Finland through HiQ Finland.

1.2.1 Service hosting provider

The SharePoint services which are the subject of this thesis are provided by the ABB Group to the different country organizations of ABB. Service hosting provided by ABB Group includes server maintenance, platform updates and maintenance, installation of customized line-of-business applications, and other day-to-day operations. The service excludes development of customized applications. Development of customized applications is left to the responsibility of the business units requiring a solution to fit their business requirements.

1.2.2 Multivendor environment

ABB's SharePoint environment can be described as a multivendor environment. No single IT vendor is responsible for all service, development, maintenance and support tasks. Different areas of the service have been outsourced to a group of vendors, both internal and external to ABB. To be able to deliver the required services, vendors have to co-operate. In a typical scenario, a vendor is contracted by a local business unit to plan, design and implement a customized application to be deployed on top of the SharePoint platform. This vendor has to work closely together with other vendors, such as the hosting provider (to meet technical requirements set by the hosting provider), a support provider (to ensure the implemented application can be properly supported after deployment) and often vendors of third systems (such as ERP systems) with which the new application is required to be integrated.

One of the earliest descriptions of IT service outsourcing to multiple suppliers comes from British Petroleum (Cross 1995). In the early 90's, British Petroleum outsourced all of its information technology operations in an effort to cut costs and gain flexibility. Unlike most IT outsourcing projects at that time, BP did not outsource all of its IT services to a single supplier, nor did it divide the required services into discrete pieces which could then be outsources separately. BP hired three contractors to deliver all required IT services as if they came from a single supplier. To achieve this, BP required the contractors to work together. The contractors had to divide the services amongst themselves and provide a joint proposal of how they would provide services to BP.

Another example of multivendor environments is described by Beck et. al., who studied the outsourcing of the development of an online banking software in Germany. The goal of this project was to migrate an old online banking system to newer technology which would allow the customer to utilize a broad supplier base in the future (Beck, Schott, Gregory 2011).

One of the typical challenges in multivendor environments is communication (Lee 2010). Vendors in different countries, different cultural and technical backgrounds have to communicate with each other to deliver expected services. One of the goals of this thesis is the implementation of a process, including standardized deliverables and means of communication to ensure successful implementations.

1.3 Technology

As stated before, this thesis was performed on the example of a Microsoft SharePoint 2010 environment at ABB. SharePoint is a collaboration product and web application platform. SharePoint, as an out-of-the-box product, contains vast amounts of features for sharing and collaborating on files, setting up blogs, websites, and workspaces for teams and even features for so-called self-service business intelligence. Most of these features are available out-of-the-box for end users without requiring extensive technical expertise. Users can setup websites, customize look-and-feel, add or remove features without writing a single line of code. SharePoint does however also provide several means for building sophisticated customized line of business applications.

SharePoint is built on the Microsoft .NET framework, and it provides APIs which can be used in to utilize SharePoint functionality in custom built, line-of-business applications. SharePoint APIs can be used locally, by custom code running on the SharePoint servers themselves, as well as remotely, by running custom applications on remote servers or client devices and utilizing industry standard web-service and REST interfaces for communication with SharePoint servers. One supported type of SharePoint customizations are so-called “farm solutions”. Farm solutions can contain application logic, user-interface elements, as well as templates and definitions used for expanding the capabilities of the SharePoint product. As SharePoint farm solutions are installed and run directly on the SharePoint server, they may contain elements which might harm the environment, compromise security or cause other problems, such as slowness and service interruptions. On the other hand, SharePoint farm solutions provide powerful means for exploiting and expanding the functionalities built into SharePoint. These risks, as well as the need for structured and organized handling of the shared server environment in use, created the requirement for well-defined, controlled and monitorable deployment and change management processes.

1.4 Objectives

In this thesis, the author will study and evaluate current deployment and change management processes used by ABB for managing the service around custom built SharePoint applications. The processes will be evaluated asking the following questions:

- What kind of change management and deployment processes are used at the case company?
- Do the processes fulfill their purpose?
- What kind of problems or challenges occur in the studied processes?

The study will be conducted in three parts. First, a literature review of industry standard change management and deployment processes, as well as a review of previous studies related to this field will be conducted. Second, the current state of processes used will be examined and analyzed, and the processes will be described in detail. Third, the performance of the processes will be evaluated by performing analysis on quantitative and qualitative data which was gathered during this study.

1.5 Authors background

The author of this thesis has been involved in several projects around ABB's SharePoint environment for over two years at the point of writing. During these projects he has gained extensive knowledge about processes and procedures around the SharePoint platform at ABB. He has also learned to know most stakeholders, decision makers and employees working on providing the SharePoint service to their customers. This knowledge has turned out to be crucial in effectively communicating with, and developing new processes for ABB.

This study was preceded by a one-year long project which was carried out to migrate and upgrade ABB Finland's previous SharePoint 2007 environments into a shared SharePoint 2010 environment provided by the ABB Group for multiple country organizations. The author worked on this project as a project manager, overseeing the whole migration process, as well as managing changing business requirements, managing application adaption done by multiple vendors and managing co-ordination and communication between project stakeholders in multiple countries.

1.6 Thesis background

It was discovered during the migration project, that the overall condition of the source environment was unsatisfactory. The overall condition of the server environment lead to major technical challenges during the project, some of which could have been avoidable. These challenges were caused, among others, by

- exceeding limits and boundaries of the product (e.g. in database size)
- undocumented configuration
- unavailable or outdated source code
- violation of good practices in application source code.

These encountered challenges made it clear, that a well-defined and monitored process would be needed to ensure reliable operation of the target environment even after prolonged time of use. The purpose of this process would be to ensure that changes to the SharePoint environment are well documented, follow product guidelines and practices, and do not violate rules set by the hosting provider or vendor or the product.

2 IT Service Management overview

The fast development of information technology in recent years has had an immense effect on the way businesses utilize IT. Hardware becomes more powerful, software more versatile and high speed networks allow worldwide connections between different systems and organizations. This development is said to mark the transition from the industrial age to the information age (van Bon, de Jong et al. 2007, Arcilla 2008).

An organization that delivers IT services to their customers (whether internal or external) needs some structure to achieve efficient delivery. This structure is increasingly formed around processes and customers, instead of being formed around functions and technical capabilities. This development is further supported by requirements for efficiency and requirements for supporting the demands of legislative regulations, such as the Sarbanes-Oxley Act of 2002 (Van Bon et. al. 2007).

The IT processes at the case company are mainly built around the ITIL framework. Other frameworks, such as the Microsoft Operations Framework and COBIT, exist, these are however not further examined in this thesis (Galup, Dattero, Quan, Conger, 2009).

ITIL was developed in the 1980s by the British government as a response to the level of IT service quality provided to them by both internal and external IT companies. The CCTA (Central Computer and Telecommunications Agency, since renamed into the Office of Government Commerce, OGC) was instructed to develop a vendor independent, standard approach for efficient and effective delivery of IT services. The result of this development is the Information Technology Infrastructure Library (ITIL), which is made up of a collection of best practices.

2.1 Business processes in general

A process, according to ITIL is “a structured set of activities designed to accomplish a specific objective”. A process has one or multiple inputs and turns them into defined outputs, and it may include any roles, responsibilities, tools and controls which are required to reliably deliver an output. A process may also define policies, standards, guidelines activities and work instructions if needed (Cannon, Wheeldon 2007).

2.1.1 Process redesign

The first studies around process standardization and redesign date back to 1990. In their study "Information Technology and Business Process Redesign" Davenport and Short define business processes as "a set of logically related tasks performed to achieve a defined business outcome". According to their definition, processes have two important characteristics (Davenport, Short 1990):

1. Processes have customers, that is, processes have defined business outcomes and there are recipients for the outcomes. The customers of a process can be both internal and external to the company.
2. Processes cross organizational boundaries, that is, normally they occur across or between organizational subunits. Processes are generally independent of formal organizational structure.

Davenport and Short also identify four objectives for redesigned processes, beyond what they call "rationalization", i.e. eliminating obvious bottlenecks and inefficiencies. These objectives are:

- Cost reduction
- Time reduction
- Output quality
- Quality of work life / learning / empowering

They claim, that cost reduction alone is not a sufficient goal by itself, and that excessive attention on cost reduction results in tradeoffs which are usually unacceptable to process stakeholders. Cost reduction should however be considered when planning other objectives of the process redesign.

Time reduction is an important aspect in today's economy, in which an increasing number of companies compete on the basis of time. Reducing the time it takes for a process to complete can in some cases give a company the needed competitive advantage.

Output quality, according to Davenport and Short, has frequently been the focus of process improvement in industrial manufacturing environments, but should be regarded as equally important in service industries. For example freedom of defects can be used as a measure of output quality.

Quality of work life is a frequently neglected objective of process improvement, as in most companies the strongest pressure is to produce benefits for the organization. Optimizing all of these objectives simultaneously is rarely possible, according to this particular study, yet managers seem to believe in the value of learning and empowerment.

2.1.2 Process standardization

It has been suggested that standardized business processes can enhance efficiency, quality and control of the process in question. According to Beimborn, a business process consists of several sub-processes or activities that are (logically) ordered, and outputs trying to achieve a defined business goal. Their study, focusing on inner-organizational process standardization, lists several benefits of standardized processes. Process standardization can improve operational performance and reduce processing cost by eliminating errors, achieving economies of scale, and by facilitating communication. As companies tend to use several variants of business processes simultaneously, standardizing these variants can enable identifying the process with the highest performance and applying this as the “process standard”. Additionally, business process outsourcing is facilitated by standardized processes (Wullenweber 2007). As a downside, process standardization can create inefficiencies created by bureaucracy and the centralization of authority (Beimborn 2009, De Vries 2006, Münstermann et. al. 2010).

From a point of view focusing on output quality, a standardized process offers compliance with regulations (e.g. SOX) and can thus result in improved customer confidence. Guidance within a standardized process can help not to miss for example certain elements of data collection, which could otherwise lead to an inferior outcome in terms of quality (Beimborn 2009).

Additionally process standardization has been found to reduce complexity in business processes, thus allowing better coordination and monitoring of the process and also leading to higher transparency (Lahtela 2010). Beimborn admits however, that their study has some limitations and might lack generalizability, as it was conducted on a small sample size, on just a single process in the banking industry in Germany.

Complete implementation of standardized process frameworks, such as ITIL, is not necessarily required to achieve the goals of process standardization. Often opportunities for improvement can be found in a small set of processes, depending on the case of the organization (Deutscher, Felden, 2010, Fry 2010). The requirements for a successful process standardization also depend on the field of business, e.g. the IT Service Management business has its very own requirements for processes (Dorling 1993)

2.2 The Information Technology Infrastructure Library (ITIL)

The first version of the Information Technology Infrastructure Library (short ITIL) was developed in the 1980s by the Central Computer and Telecommunications Agency, which has since been merged into the Office of Government Commerce in the UK. The need for development of a standardized approach for efficient and effective delivery of IT services was identified by the UK government, which at the time found the quality of IT services delivered to them to be lacking. ITIL consists of a collection of 'best practices' for IT service providers (Yao 2010).

ITIL offers a systematic approach to the delivery and quality of IT services. It describes most important processes in IT organizations, and includes checklists for tasks, procedures and responsibilities which can be used as a basis for tailoring to the needs of individual organizations (van Bon, de Jong et al. 2007).

ITIL is used by organizations world-wide to improve their capabilities in IT Service Management. ITIL is closely related to the ISO/IEC 20000 standard, which provides formal certification of IT Service Management capabilities of an organization. ITIL offers a body of knowledge useful for achieving the standard (Lacy, Macfarlane 2007).

ITIL v3 consists of 26 processes and four functions. The four functions are each described in their own publications: Service Strategy, Service Design, Service

Transition, and Service Operation. Additionally ITIL provides guidance on service improvement in the Continual Service Improvement publication.

In the following section, the function of all 5 books will shortly be covered. Bearing in mind the subject of this study, the Service Transition and Service Operation books are of more interest than the other parts of the ITIL framework.

2.2.1 The ITIL Life-cycle

The ITIL lifecycle represents the different stages in the lifecycle of every IT Service. At the core of the ITIL lifecycle is the Service Strategy, which provides guidance on designing, developing and implementing Service Management. Service Design covers design principles and methods for converting strategic objectives into a service portfolio. Service Transition covers transitioning new and changed services into operations. Service Operation embodies guidance on achieving effectiveness and efficiency in the delivery and support of services. Continual Service Improvement aims to create and maintain value for customers through better design, introduction and operation of services (Lacy, Macfarlane 2007). The *Service Design*, *Service Transition* and *Service Operation* functions are considered to evolve around the overall *Service Strategy*, thus generating a cycle around the delivered IT Service. The *Continual Service Improvement* function aims to improve all of the stages of a service life-cycle, and is thus separated into its own set of practices and processes.

2.2.2 Service Operation

Service Operation, according to the ITIL publication, is the ‘business as usual’ phase in the lifecycle of a service. The Service Operation book provides best-practice and guidance on all aspects of managing day-to-day operations of an organization’s IT services. The goals of Service Operation are to co-ordinate and fulfill activities and processes required to provide and manage services for business users and customers (van Bon, de Jong et al. 2007).

Service Operation staff members should be aware that they are providing services to the business, thus staff should be trained not only how to provide and support the service, but also teach the attitude with which to provide these services (van Bon, de Jong et al. 2007). Success can be achieved by establishing long-term relationships with the business, as well as suppliers (Hall 2002).

2.2.3 Service Transition

Service Transition, as defined in the ITIL, includes the management and co-ordination of the processes, systems and functions required for the building, testing and deployment of a ‘release’ into production, and establishing the service specified in the customer and stakeholder requirements.

2.3 Change Management according to ITIL

According to the ITIL Framework, Change Management (in Service Transition) is the process responsible for controlling the lifecycle of all changes. The primary objective of Change Management is to enable beneficial changes to be made, with minimum disruption to IT Services (Lacy, Macfarlane 2007). The goal of the change management process is to ensure that changes are deployed in a controlled way, evaluated, prioritized, planned, tested, implemented and documented (Van Bon et. al. 2007). The ITIL Change Management framework defines numerous roles, responsibilities and processes which can be used to facilitate and control Change Management. The roles, responsibilities and processes, as defined by ITIL, are described in this chapter.

2.3.1 Roles

ITIL defines a Role as a set of responsibilities, activities and authorities granted to a person or a team. A single person or team may also incorporate multiple roles. Some of the roles related to the change management process are the Change Manager, Change Advisory Board and Application Management Team and the Service Manager. The roles which are introduced here, are the ones implemented and used in the case company.

2.3.1.1 Change Manager

ITIL V3 defines the responsibility of the Change Manager as follows:

The Change Manager receives all change requests and in collaboration with the initiator assigns a priority to a change. The Change Manager presents all Change Requests to the Change Advisory Board for consideration, and based on the advice from the Change Advisory Board, authorizes acceptable changes (Van Bon et. al. 2007). The Change Manager also summons Emergency Change Advisory Boards for urgent changes, chairs all CAB meetings and issues Change Schedules. In co-operation with all parties involved, the Change Manager co-ordinates change building, testing and implementation, updates the change log with all progress, and in the end reviews all implemented changes to ensure that the changes meet their objectives (Lacy, Macfarlane 2007). The responsibility stays within this one role throughout the change (Pfitzinger, Jestadt 2011).

2.3.1.2 Change Advisory Board

The purpose of the Change Advisory Board is to support the authorization of changes and to assist Change Management in the assessment and prioritization of changes. Members of the CAB should be able to ensure that changes are adequately assessed from both a business and a technical point of view. The ITIL framework suggests, that the Change Advisory board should include people from the whole range of stakeholders and could potentially include:

- customers
- user managers
- user group representatives
- developers
- specialists or technical consultants
- operations staff
- contractors and other third parties

It is also emphasized, that the CAB should be composed according to the changes being considered, and that it should involve suppliers if useful or appropriate.

CAB meetings can be arranged face-to-face or online, with all participants engaging in the meeting, or “electronically” via support tools and e-mails. Arranging CAB meetings electronically, often facilitates participation for CAB members, however communication has been found to be more efficient in face-to-face meetings.

The CAB meeting should have a standard agenda. The agenda should include going through failed, unauthorized and backed-out changes. Change Requests to be assessed by the CAB should be processed in a structured order, according to the priority of the changes and changes possibly processed prior to the meeting should be checked through. New changes should be scheduled and previously drawn schedules should be updated. The CAB should also review implemented changes, as well as the change management process itself and any changes proposed to the process.

2.3.1.3 Application Management Team

The Application Management Team or department is usually dedicated to a specific application or set of applications. Its main purpose is to ensure day-to-day operation of the services and applications managed by the team. It is also responsible for identifying knowledge and expertise required to manage their applications, and recruit, contract or insource resources with the required skillsets. Training, both for resources in the team, as well as end-users can be designed and delivered by the Application Management Team. The team also defines standards and principles used in the design of new application architectures, and is involved in the design of the architecture of new applications or services. Additionally to the involvement in the design phase of development projects, it is also involved in the continuous improvement, such as updates and changes to existing applications.

The technical knowledge and expertise of the Application Management Team is used to evaluate changes and in many cases changes are implemented by the Application Management Team. It also ensures that all system and operating documentation is up to date and complete, and that Application Management staff is familiar with the content of the documentation. Together with the teams performing software development, it is also involved in defining and maintaining documentation related to the managed applications.

Traditionally, application development and application management are performed by separate teams in an organization. While the teams performing application development are focused on building new functionality for their customer, and are more concerned with the functionality of the application than day-to-day operations. The Application Management Team however focuses on existing functionality, how to deliver it, and how to ensure that the application is stable and meets performance requirements.

Application development is usually done in projects, with specific goals and specified schedule and budget. It is often difficult for developers to consider ongoing operation of the application after the development phase, especially if they are not involved in the operation of the application. Application Management is performed as an ongoing process, and staff is usually less involved in development projects.

The main goal of Application Management is to ensure availability and stability of applications, thus they are rewarded for consistency and preventing unexpected events and preventing possible unauthorized functionality implemented by the development team. Development teams however are rewarded for creativity and completing projects, with less focus on the impact on daily operation.

2.3.1.4 Service Manager

The Service Manager is responsible for managing the end-to-end lifecycle of an IT service by co-ordinating the development, introduction and evaluation of products and services. The Service Manager is responsible for achieving company goals, financial, customer, vendor and inventory management as well as benchmarking his service.

ITIL defines some key activities for the Service Manager. These activities include analysis of service management processes, setting targets for efficiency improvements and setting targets for service improvements. Additionally the Service Manager is responsible for ensuring that all approved actions are completed and that they achieve the desired result. This is particularly important in change management and deployment processes, as the Service Manager is ultimately accountable for changes being made to the service.

2.3.2 Activities

An activity, as defined by ITIL, is a set of actions designed to achieve a particular result. A process consists of activities which have to be performed to achieve the desired outcome of the process. In this chapter, activities related to the Change Management process are described in further detail.

2.3.2.1 Change request

A change request is a formal way to request a change to one or more existing configuration item (CI). A change request can be delivered e.g. as a ‘Request for

Change' document, a service desk call or by using other means, such as online chat or e-mail.

2.3.3 Records

A record is a document containing the results of an activity or process. Records regarding the change management process are stored in the change management database. Each step in the change management process must produce a record as evidence that a certain activity has been completed.

2.3.4 Types of changes

The ITIL Service Transition (Lacy, Macfarlane 2007) describes three basic types of changes; normal service changes, standard changes and emergency changes. An emergency change is a change intended to repair an error in an IT service that has significant negative impact on the business. Emergency changes are implemented as soon as possible, and to allow prompt reaction to an emergency, some details of emergency changes may be documented retrospectively. Emergency changes are reserved only for fixing unplanned, unforeseen errors in an IT service. Changes which are meant to introduce immediately required new business functionality to a service are handled as normal service changes. Emergency changes are not discussed further in this thesis. This thesis will focus on normal changes and standard changes, as well as their implementation in the case company in the case service.

A normal service change is defined by ITIL as the addition, modification or removal of an authorized, planned or supported service or service component and its associated documentation. The purpose of change management processes is to ensure that standardized methods and procedures are used for efficient handling of all changes, that all changes are recorded in the Configuration Management System and that the overall risk to the business is optimized. The objective of the change management process is to ensure that changes are recorded, evaluated, authorized, prioritized, planned, tested, implemented, documented and reviewed in a controlled manner.

Figure 2 shows an example process flow of a normal service Change (Lacy, Macfarlane 2007).

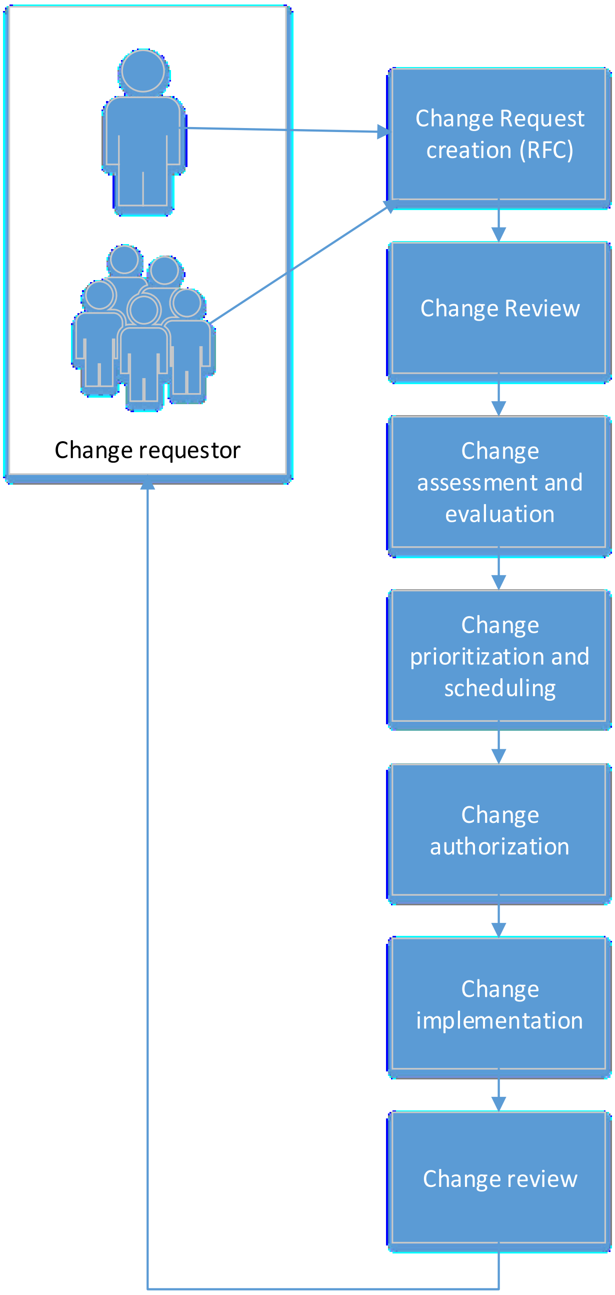
Change requestor
Change Request creation (RFC)
Change Review
Change assessment and evaluation
Change prioritization and scheduling
Change authorization
Change implementation
Change review

Figure 2 Example change management process workflow

The third type of changes, the standard change (also called pre-authorized change) is a change to a service or infrastructure for which the approach is pre-authorized by change management. The changes in this category include only tasks which are well known, documented and proven and the risk of these changes is usually low and well understood. In the context of application services, these changes are low impact, routine changes.

2.3.5 Change procedures

A normal change should follow a certain, pre-defined procedure. This change procedure, as defined in the ITIL framework, is described in this chapter. This change procedure is also visualized in Figure 2.

In the beginning of the change process, a change request has to be raised by the change initiator. The change initiator might be an individual or group inside the organization requiring the change. For major changes a change proposal might be required, containing full description of the change as well as business and financial justification for the change. Changes requiring change proposals are not covered by the change management processes studied in this thesis, and will thus not be further evaluated.

A change request is raised by completing a “Request for Change” document (RFC). RFCs should be documented and logged using a chosen procedure. It is usually advisable to use an integrated Service Management tool for logging RFCs, as these tools contain the possibility to store also data on assets, Configuration Items (CI) and the relationships between them. Multiple methods can be used for submitting an RFC, such as paper forms, e-mails or using a web based interface which might be directly connected to the Service Management tool. The RFC should include a description, a list of items to be changed and a reason (i.e. business case) for the change. The RFC should also clearly identify the affected CIs, contain contact information of the requestor, as well as propose a change priority and estimated impact.

After the request for change has been raised, the RFC is reviewed. The purpose of the review is to filter out incompletely filled change requests, repeats of earlier RFCs which might be still under consideration and change requests which are outside of the scope of the service or seem totally impractical. If a RFC does not fulfill the requirements, it should be returned to the requestor with brief details of the reason. This initial review does not assess the RFC any further. In-depth evaluation is handled in the next phase.

After the initial review, RFCs are assessed and evaluated. The ITIL framework proposes the “seven Rs” of change management as a starting point for assessing the possible impact of a failed change, as well as the impact on assets and configuration of the service. These severn Rs are:

- Who Raised the change?
- What is the Reason for the change?
- What is the Return required from the change?
- What are the Risks involved in the change?
- What Resources are required to deliver the change?
- Who is Responsible for the build, test and implementation of the change?
- What is the Relationship between this change and other changes?

Following the assessment of the possible impact of the change, the change is evaluated and prioritized. During the evaluation, each change assessor should indicate whether they support the implementation of the change. Once a change has been approved, it is prioritized. The change priority establishes the order in which changes are implemented.

Change prioritization is followed by scheduling of the change to a release window. Release windows should be pre-agreed and established to help the organization plan change throughput. For example a release window of one hour each week may be agreed for minor releases. Larger releases requiring significant downtime should be scheduled together with the business, to avoid unnecessary business disruption.

Before the final implementation of a change to productive environments, the change has to be authorized. Change authorization can be done by a role, person or group of people, largely depending on the organizational culture. Additionally, changes affecting larger parts of the business might require authorization at a higher level than smaller changes.

After a change has been authorized, it is passed to the relevant technical groups for building the change. Change Management bears the responsibility for ensuring that changes are implemented as scheduled. Additionally, Change Management ensures that all changes are thoroughly tested before implementation to live environments.

The process described in Figure 2 ends with a review of the implemented change and closure of the change record. A change review (also called post implementation review, PIR) should be conducted to confirm that the change meets business requirements and that the stakeholders of the change are satisfied with the implementation (Jin 2008).

3 Empirical study

Every study has a purpose or goal, thus studies can be categorized based on their respective goal or goals. The goal of a study in turn has a great impact on the research method. A division of research goals in four main categories has been suggested in (Hirsjärvi et. al. 1997). These four categories are:

- Surveying study: The goal of a surveying study is to observe events, try to find new points of view or previously unknown phenomena, increase knowledge about less-known phenomena or create hypotheses.
- Explanatory study: The goal of an explanatory study is to seek explanations to a given situation or problem, usually searching for cause-effect-relationships.
- Describing study: The goal of a describing study is to generate detailed descriptions of individuals, events or situations and to document their most interesting and central features.
- Predicting study: The goal of a predicting study is to generate predictions about upcoming events or the future behavior of people caused by some phenomenon.

This study aims to be a surveying and describing study, focusing less on seeking explanations to events that already occurred or trying to predict future events, and focusing more on trying to survey and describe existing processes, and seeks for improvement opportunities therein. It is suggested, that qualitative research methods, such as case studies are well suited for surveying and describing studies.

3.1 Research method

This study was conducted as a case study. A case study is a research method which aims to gather detailed and intensive data about a single case or a small set of interrelated cases. Typical features of a case study are the selection of a single incident, situation, or set of incidents. The study is usually performed on an individual, a group or community. Incidents and events are studied in their natural surroundings, where the incidents and events are part of a larger entirety. Information and knowledge can be gathered using multiple methods, including observation, interviews, and study of documents. The object of interest of the study are often processes (Hirsjärvi et. al. 1997).

Robert Yin describes three steps for designing a case study (Yin 2011). The first step is to define the "case", which is usually a bounded entity, such as an organization or a social phenomenon. Yin points out, that the case might be redefined after collecting some data. The case serves as the main unit of analysis, it might however, depending on the nature of the studied phenomenon, contain nested units within the main unit.

The second step is to select the type of the case study. A case study might consist of a single or multiple cases (single- or multiple-case study). Each case might be holistic, or it might contain subcases, leading to four different types of case studies. The four different types of case studies are shown in Table 1.

Table 1 Case study types

	Number of cases studied	
Number of units of analysis	Single-case, single unit of analysis	Multiple-case, single unit of analysis
	Single-case, multiple units of analysis	Multiple-case, multiple units of analysis

The third step is to decide, whether or not to use theory in developing research questions, selecting the case or cases, or defining the relevant data to be collected. Yin suggests that someone less experienced in case studies would benefit more from basing his or her research on previous accepted research, but also says, that a case study which deliberately avoids any theoretical perspective has the potential to produce a new and truly different insight into the field of research.

The goal of this thesis is to study and improve processes in an IT organization, thus, it can be concluded that a case study is a suitable research method for the task on hand.

During the study, information has been gathered in different ways. Documentation describing the processes of the organization were studied, processes were discussed in face-to-face meetings with all stakeholders of the processes. Documented processes were compared to the actual actions performed. The day-to-day work routines of involved individuals were observed.

Additionally to the empirical data gathered using observation of and conversations with people involved, data gathered in the Change Management System over the duration of this study was used. The Change Management System contains both numerically analyzable data, such as throughput time and numbers of failed and successful changes, as well as empirical data about the interaction between different stakeholders. The dataset acquired over the course of this study is relatively small, and does not allow statistically significant analysis. It does however provide a sufficiently large amount of data for analysis of the performance and improvement opportunities of the processes at hand.

3.2 Case design

The case selected for this study are the management processes related to providing a global IT service at the case company. More precisely the deployment and change management processes related to changes in a global, multivendor SharePoint environment. Even though there are two main processes involved (initial deployment and changes to existing deployments), the nature of these processes has been found to be very similar during initial observations.

Instances of the processes to be studied can take anywhere between a few days and multiple months to complete. During the course of this study a total of 59 instances of the processes which are subject of this study were initiated. According to the categorization suggested by Robert Yin this is thus a “single-case, multiple units of analysis” study.

The purpose of this study is to evaluate and possibly develop the processes currently used in the case company. IT service management has been a major field of study for multiple decades (e.g. Arfa 1991, Curtis 1993, Kazlaukas 1994) and large quantities of previous research and well-established ITSM frameworks exist. Additionally the fact that ITIL based ITSM frameworks are widely used at the case company made the evaluation against some of the existing frameworks a natural choice.

3.3 Research questions

The goal of this thesis is to evaluate deployment and change management processes related to the service in question. The following questions will be evaluated in this thesis:

1. What are the change management and deployment processes currently in place?
    a. What are the objectives of each process?
    b. What roles are involved in each process?
    c. What are the steps or activities of each process?

The purpose of this research question is to describe the existing processes. The question is answered by describing the objectives, roles and activities of the studied processes in detail.

2. What is the performance of the processes used?
    a. What are success and failure rates of the processes?
    b. What are typical causes of failure?

The performance of the processes used is evaluated through material gathered through various change management databases and systems during the course of this study. The goal is to not only analyze success/failure rates, but to also gain insight into the causes behind possible failures. Based on the knowledge gathered about challenges of the processes, improvement opportunities could be suggested.

3. Do the processes fulfill their intended purpose?

The outcome of the processes is evaluated against their respective objectives. A process might fulfill its goals e.g. even in case of high failure rates. The stated goals of the processes are compared against the data gathered about process instances during this study.

4 ITSM processes at the case company

The ITIL Framework provides a set of roles, responsibilities, functions and processes, not all of which have to be implemented in every IT organization. The purpose of the framework is to provide a set of best practices and guides which can be implemented and adapted according to the requirements of the organization. In this chapter, the roles, stakeholders and processes as they are implemented in the case company are described.

4.1 Roles

The service in this case study is provided by a company internal IT department, which provides similar services to all country organizations of the case company. There are however several other groups affiliated with this service. These groups and stakeholders are described in this chapter in further detail.

4.1.1 Global Service Provider (GF-IS)

The Global Service Provider (called Group Functions – Information Services) is a separate unit responsible for providing services related to information systems. It is the responsibility of the service provider to run and maintain a platform for running business applications. Business applications are (in this case) not developed or maintained by the service provider. Inside of the organization of the service provider, there are again several roles, which will be described next.

4.1.1.1 Onboarding manager

The onboarding manager bears responsibility for the process of deploying new business applications and making changes to existing business applications. The onboarding manager has an overview of all ongoing deployment processes on a general level. Regular meetings for following up on the status of ongoing processes are arranged by the onboarding manager with customer representatives. The onboarding manager is responsible for providing necessary information and guidelines to customers and development partners.

4.1.1.2 Operations team

The operations team is responsible for installation and maintenance of the application platform. The operations team is responsible for monitoring platform servers, making changes to the platform, adding new servers and services to the platform and overall smooth operation of the environment. Additionally, the operations team is also responsible for installation of Business Applications developed by Local Business Units or (more often) their development partners (vendors). The operations team also monitors each deployed business application for availability. In case of disruption of the availability of any deployed Business Application, it is the responsibility of the Operations team to ensure that the disruption is not caused by a problem in their platform services. Any problems caused by e.g. faults in the application code, are not the responsibility of the operations team, and have to be handled by a support vendor. The operations team is both in responsibilities and tasks very similar to the Application Management Team described in the ITIL framework (chapter 2.3.1.3).

4.1.2 Local IT Department (Customer)

4.1.2.1 Service Manager

The Service Manager is responsible for the Service delivered to stakeholders in general. The Service Manager arranges regular meetings with the Service Provider and customers (end-users) of the system. The role of the Service Manager is very similar to the role of the Service Manager as defined by ITIL (see chapter 2.3.1.4)

4.1.2.2 Change Manager and Change Coordinator

In the case scenario, the Change Manager and Change Coordinator role are held by a single person. This person is responsible for receiving, logging and allocating RFCs, authorizes changes, issues change schedules and updates the change log with all progress. Unlike in the process defined by ITIL however, the Change Manager is not responsible for coordinating change building, testing and implementation. The coordination of these tasks is left to the application owner, or someone assigned by the application owner for these tasks.

4.1.2.3 Change Advisory Board

In the case scenario, the Change Advisory Board does usually not assess the kind of changes which are the subject of this study. The Change Advisory Board is responsible for making higher level decisions regarding the Service. In rare cases when a change affects the whole service, and thus all customers of the service, the Change Advisory Board might be convened.

4.1.3 Development Partner

Development partners develop solutions ("Applications") to be deployed on a platform ("Service"). Development partners are also used for implementing changes to already deployed solutions. The Development Partner has no access to Stage or Production environments of the System. The Development Partner develops in separate development environments. Changes (i.e. the source-code, documentation, scripts) are reviewed and approved by the Operations team of the Service Provider. Only changes implemented according to the rules and guidelines of the Service Provider, which have subsequently approved by the Service Provider can be deployed to Stage and Production environments. Development Partners, depending of the scope of a solution or change can have one or multiple of the following roles. One person can also embody multiple roles. In smaller development projects, all roles might be embodied by one person.

4.1.3.1 Project Manager

The Project Manager is responsible for a development project in the big picture. The Project Manager manages resources used in a development project, handles communication about the project with the customer and plans and ensures schedules.

4.1.3.2 Architect

The Architect is responsible for the overall technical architecture of a developed solution. The Architect might also participate in the development, but his main responsibility is the overall technical solution.

4.1.3.3 Developer

Developers implement solutions and changes according to customer requirements. Developers work in close co-operation with the Architect and Project Manager to ensure the fluent progress of a project.

4.1.4 Local Business Units

In the case organization, a local business unit is an organizational entity with some independency regarding IT purchases. In the bigger picture, Local Business Units (LBUs) are subordinates of Business Units, which are in turn subordinates of Divisons. This hierarchy will not be further taken into account in this thesis, but this thesis will rather focus on the LBU level.

4.1.4.1 Application Owner

The Application Owner is a business user holding ownership of a business application. The Application Owner is usually not a technical person, and thus not familiar with the technical details of the implementation of the application. The Application Owner often works in co-operation with vendors to implement changes to an application, and changes are planned according to the business requirements drawn by the Application Owner. It is also the Application Owners responsibility to initiate the Change Management process for implementation of a Change to productive environments. The Application Owner communicates with the Change Manager regarding any actions in the Change Management process.

4.2 Tools

4.3 Deployment management processes

In the course of this study, a deployment management process for business applications built on Microsoft SharePoint 2010 was implemented in the case company. The different phases of this process as well as all stakeholders are described in this chapter.

4.3.1 Background

Applications built on top of SharePoint are usually developed by 3rd party contractors, the so called Development Partners. The development is conducted in separate development environments, usually owned and maintained by the development partner. These development environments are not a part of the logical or physical network of the case company. Application development is conducted using the preferred development project management methodologies of the customer (usually a Business Unit) and the contractor. An overview of the process used to deploy applications developed by 3rd party contractors is shown in Figure 3.

4.3.2 Objective of the process

The goal of the deployment management process is to introduce a new business application which has not been previously deployed to the target environment. This deployment should cause minimal disruption (e.g. required downtime) and should not affect previous deployments. Additionally it is the goal of the process to conclude the deployment during a single service windows, and to avoid the need for additional, unplanned deployments or changes to the deployment.

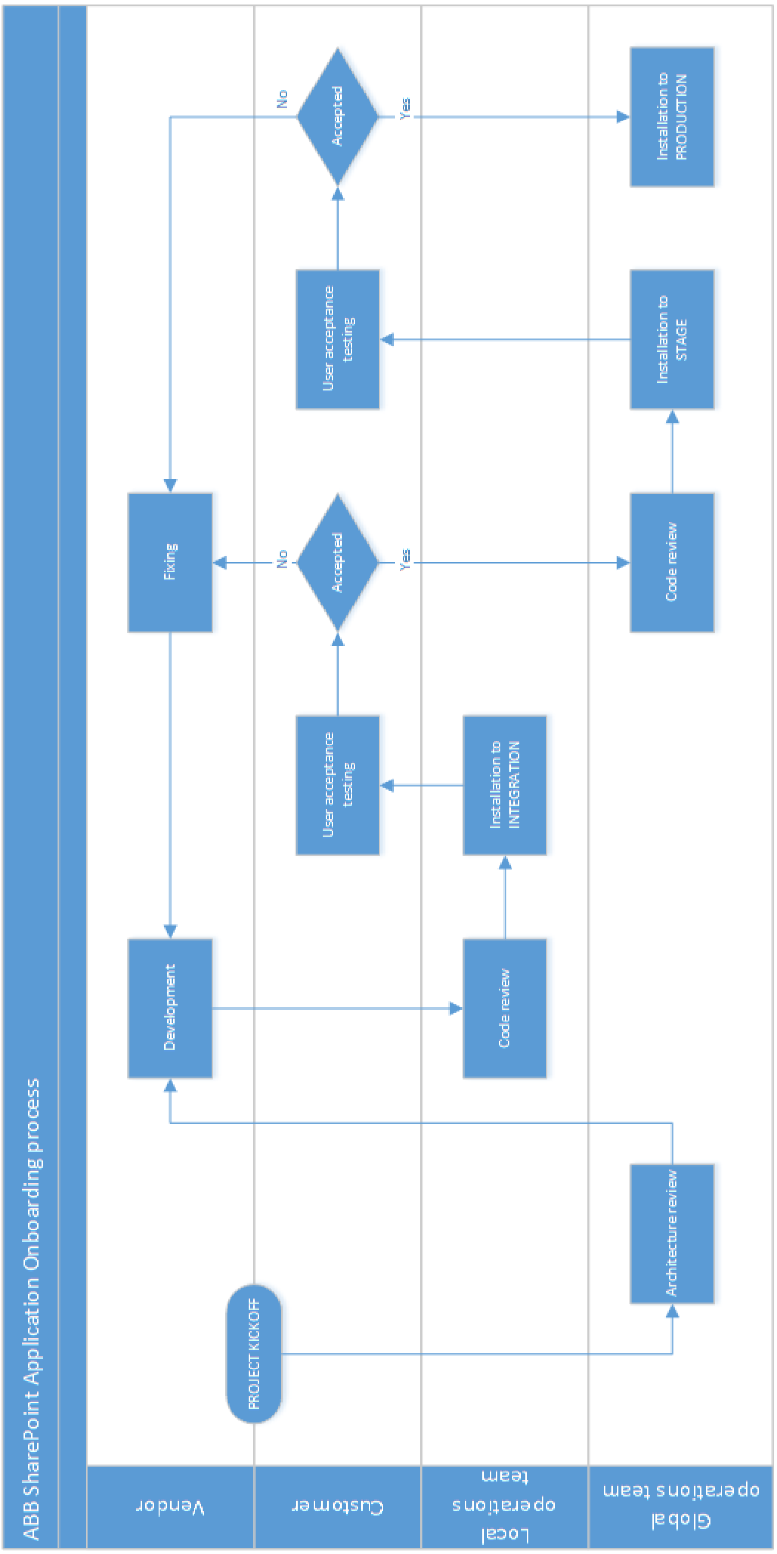

Figure 3 Application deployment process

4.3.3 Initiation of the process ("Kick-off")

Every development project of a new application which is developed to be deployed to the global Custom Applications SharePoint environment is required to participate in a so called Kick-off meeting. This meeting has multiple purposes.

First and foremost, the Kick-off meeting is arranged to ensure that all stakeholders are aware of the development project. At minimum the participants of the Kick-off meeting are:

- The Service Manager, who is required to be aware of all ongoing development projects related to the Service. The Service Manager uses the information about ongoing projects to forecast resource requirements in the operations department. Additionally the Service Manager maintains an overview of all development projects to ensure that projects do not largely conflict with the purpose and principles of the Service.
- The Change Manager, who authorizes the deployment of the developed application and manages the overall process from initiation of the development project to deployment to productive environments, however mostly excluding involvement in the development project itself..
- The Onboarding Manager, who is responsible for ensuring that the newly developed application can be accepted to the service. Onboarding a new business application might require the deployment of additional servers or resources (such as disk space or computational capacity), changes to existing appliances or even the hiring of additional personnel.
- A technical architect from the operations team, who in the end of the onboarding process reviews the technical solution of the application, based on source code and documentation review. As acceptance of the technical solution by the architect is required for deployment to productive environments, the architect should be available to provide necessary information to the developing party as early as possible in the development project.
- The operations team lead, who is responsible for the Service from a technical point of view.

- A customer representative, usually the future Application Owner of the product of the development project. The Application Owner knows the business requirements of the application and can shortly describe both the problem to be solved by the application and the way the application is supposed to solve this problem from a business point of view.
- A technical representative of the development partner, usually a technical project manager, architect or developer familiar with the whole solution. The technical representative can roughly describe the architecture of the planned solution to the extent which is already known.

During the kick-off meeting, different aspects of the planned application are discussed. The agenda of a typical kick-off meeting is attached in Appendix A.

First, the application owner describes the business purpose and business functionality of the application. Important factors, such as the groups of users (e.g. company internal or external users), amounts of predicted users and business criticality are discussed shortly. The purpose of the presentation of the planned solution is to give a broad overview of the solution. The presentation is recommended to be performed only verbally and no formal documentation or presentation is required at this phase. The practice was chosen, as it was noticed that formal presentations tend to shift the focus of the presentation away from important matters to small details which are irrelevant at this phase of a project, and are additionally bound to change during the course of the project.

After a short description of the business perspective, the technical representative of the vendor describes the planned technical implementation of the application, again in broad terms, as details are likely to change and would not affect the outcome of the meeting. Important aspects to be described at this early phase are possible integrations, dependencies to other applications or tools (both upcoming and already deployed) and implementation type of the solution. The main architectural components should be listed, and impact on the hosting environment (such as required storage space for data accumulated by the application, required computational capacity and bandwidth requirements) should be roughly estimated.

It is the goal of this presentation to make it possible to recognize bad architectural decisions in an early phase of the development process and to avoid applications being developed with false assumptions about the target environment. Often developers unfamiliar with the target environment might make decisions which later make it costly or impossible to implement the developed solution to productive environments.

If the developer or customer is not yet familiar with documentation available for supporting the development of applications, such as coding guidelines, environment descriptions, documentation requirements and templates, and process documentation, then this documentation is shortly presented and access to this documentation is made available.

Altogether this meeting should not take longer than an hour. Kick-off meetings lasting longer than an hour are usually caused by discussion about small details or absence of one of the main stakeholders from the meeting, resulting in guessing and making assumptions during the meeting.

4.3.4 Architecture review

The architecture review is a review conducted based on the architecture plan documentation which has to be produced in the beginning of each development project. The purpose of the architecture review is to assess the suitability of the planned architecture to the target environment. During this phase only documentation is evaluated, as extensive development has usually not been done before this phase. The goal of the architecture review is to provide the possibility to intervene early in the development process e.g. if the plans for implementation do not meet requirements of the environment. By providing the possibility for early intervention in the development process, costly and time consuming redesigns which might otherwise be necessary in case of unsuitable solutions can be avoided. This step is critical for the overall success of a new deployment.

4.3.5 Development phase

During the development phase the development team works on the implementation of the proposed solution. During this phase, the developers and business stakeholders usually work on their own and do not require assistance from the platform supplier.

Depending on the scope of a project, the development phase usually lasts from a few weeks to over a year. Development is usually conducted in project specific development environments, which do not necessarily resemble the target environments in all aspects. Important differences which can occur which were identified during this study are:

- Different versions of the underlying platform used
- Custom deployments missing from development environments

Using a different of the platform during development can lead to incompatible solutions and custom deployments missing from a development environment can lead to unexpected side-effects when a solution is deployed to the actual target environments.

4.3.6 Code review

Before accepting any change or new application for the implementation to Stage or Productive environments, the source code of the application is reviewed by a representative of the hosting provider. In the code review, the code is checked for possible problems which might affect the whole hosting environment, such as memory leaks, and other issues, such as conflicting file names, unclear naming schemes, and improper exception handling which might pose a risk to the platform or other applications. The code review does not check if the implemented application actually meets business requirements, it is rather a purely technical review to ensure maintainability of the environment after installation of the change or application.

4.3.7 Deployment to staging environments

New applications and changes to existing applications are deployed to a staging environment before deployment to productive environments. Stage deployments are usually conducted on Wednesdays (except for public holidays).

4.3.8 User acceptance testing

The Application Owner is responsible for conducting comprehensive testing in the Staging environment before accepting the Change for installation to Production environments. In case of small changes testing is often conducted by the Application Owner personally. In case of larger changes affecting possibly hundreds or thousands of users, User Acceptance Testing is often conducted with larger user groups, to gather sufficient knowledge about the change. User Acceptance Testing is conducted to ensure that the business requirements of the Change are met. During the testing phase in the staging environment, the Change is monitored additionally by the Operations Team running the Production environments to ensure that there is no negative impact on the environment. A Change negatively affecting the environment might be cancelled by Operations, even if it meets business requirements.

4.3.9 Deployment to productive environments

Deployments to productive environments are conducted after the deployment has been performed in a staging environment, tested and accepted by the respective owner and accepted in prior reviews. Deployments to productive environments are conducted during Saturdays to minimize the impact on business functions.

4.3.10 Post implementation review

After deployment to productive environments, the deployment has to be formally accepted by the application owner. This formal acceptance is based on a review conducted by the application owner, the content of which might vary depending on the scope of the deployment.

4.4 Change management processes

One of the biggest concerns regarding the Deployment management process described in chapter 4.3 during this study was related to possible performance of the process in situations where only small changes are needed, or changes to existing deployments need to be installed with a tight schedule to meet business requirements. For these scenarios a second change management process was implemented. The process involves most of the steps and phases described in chapter 4.3, simplifying however where unnecessary bottlenecks could be identified. Additionally this process omits certain steps which are required for large implementation projects, such as financial approvals, due to the nature of changes which are usually implemented with a relatively small budget in a tight timeframe. This chapter describes the change management process in detail.

A “change”, as defined in this process is a change to existing functionality of an application, usually required by one of the following:

- changes in business requirements
- application bug preventing utilization of certain functionality
- changes in interfacing systems

A change, as used in this process, does not alter the overall architecture of the application already deployed for productive use. Additionally the code-base of the application does usually not change significantly.

A change management process for previously deployed applications is shown in Figure 4.

4.4.1 Objective of the process

The goal of the change management process is to deploy a change to a previously deployed business application, with minimal disruption to the users of the targeted application as well as all users of the underlying platform. As business applications which are targeted by such changes are usually in heavy use, successful introduction of the change without interrupting service outside of pre-defined service windows is crucial for a successful change.

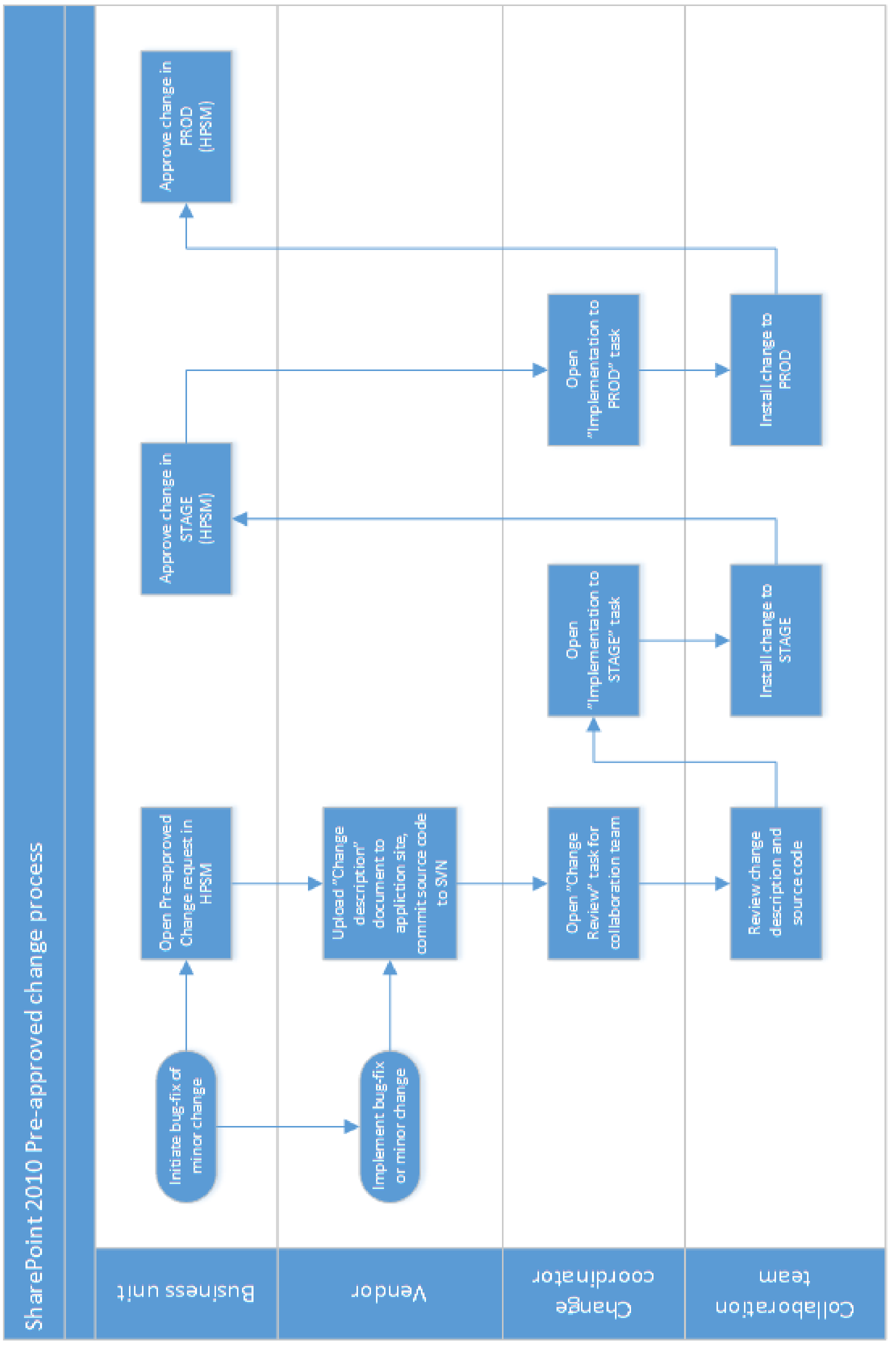
SharePoint 2010 Pre-approved change process
Business unit
Vendor
Change coordinator
Collaboration team
Initiate bug-fix of minor change
Implement bug-fix or minor change
Open Pre-approved Change request in HPSM
Upload "Change description" document to appliction site, commit source code to SVN
Open "Change Review" task for collaboration team
Review change description and source code
Open "Implementation to STAGE" task
Install change to STAGE
Approve change in STAGE (HPSM)
Open "Implementation to PROD" task
Install change to PROD
Approve change in PROD (HPSM)

Figure 4 Change management process for deployed applications

4.4.2 Change initiation

The requirement for a change is usually identified by a user or owner of the application. The application owner is responsible for initiation of the implementation of a required change with a development partner. Similarly to the deployment process described in chapter 4.3, the change management process does not take responsibility for selecting a vendor and coordinating the implementation of the change required by the business stakeholders of the application.

After the requirement for one or more changes has been identified, the application owner initiates the selection of a vendor to implement the change. Most often the vendor is already familiar with the application, and has previously implemented changes to this particular application.

4.4.3 Change implementation

The implementation of the required changes is coordinated by the application owner. The application owner (or a person appointed by the application owner) defines the requirements and is the primary contact person for the developer working on the implementation.

The party responsible for hosting the application is not familiar with the business requirements and use-cases of the application, and is thus not able to provide assistance in matters regarding the business functionality. The hosting party might however be consulted for information on the hosting environment which might affect the implementation of the change.

4.4.4 Change review

The deployment process described earlier contains a two-phase review process in the beginning of any development project. First an architecture review, which is conducted based on documentation provided, and second (after implementation of the application) a source-code review. This two-phase review makes sense for newly developed applications, as the purpose of the first architecture review is to identify possible substantial flaws in the application architecture which might make it costly or even impossible to deploy to the targeted environment. The second phase (code-review) is conducted to ensure that the application is implemented according to coding guidelines, and that the application is implemented according to the architecture described earlier. Each of these reviews might take several business days. This is not a problem in development projects, the duration of which is usually several months.

For smaller changes, the implementation of which might only take a few hours or days, this two-phase review process was deemed too burdensome and time-consuming. An application change (per definition) does not alter the architecture of an existing application, making a full architecture review redundant. The change management process thus includes a single change review, instead of the full two-phase review procedure of the deployment process.

For the change review, the development partner is required to submit a short technical description of the change, as well as the application source code. Both are reviewed in conjunction by the architect performing also architecture and code reviews.

This practice saves time, as only one review is conducted and thus allows more rapid deployments. Additionally a single change review saves time and money, as less resources are required both for conducting the reviews, as well as preparing necessary documentation.

4.4.5 Deployment to staging environments

After acceptance, the change can be deployed to the staging environment. Staging installations are scheduled on a single day each week. This reduces the possibility of unauthorized changes, as changes are deployed according to a pre-defined process, at certain times by designated members of the operations team.

4.4.6 User acceptance testing

User acceptance testing is the responsibility of the application owner. Testing can be performed by the application owner, or any user or group of users designated by the application owner for this task. The application owner is responsible for reporting the final result of the user acceptance tests. Changes are not deployed to production environments before they are officially approved by the application owner.

4.4.7 Deployment to productive environments

Once the application owner has approved the change, it is deployed to production environments. Deployment to production environments is usually done during pre-defined service windows once a week. Deployments outside of this weekly service window are only performed in emergencies, e.g. when a disruption in the system causes significant loss to the business. Deployments to productive environments are, just like deployments to stage environments, performed by designated members of the operations team.

4.4.8 Post implementation review

After deployment to productive environments, the application owner is responsible for reviewing the change and officially approving the change in the CMDB, just like in the deployment process.

5 Results

This study was conducted over the period of one year, between June 2012 and May 2013. In this section the results of this study are analyzed. Numerical data form the Change Management System was gathered over the period of one year. Additionally observations were made by the author in meetings and discussions surrounding the change management process.

5.1 Configuration Items

At the time of writing (May 2013) 52 Configuration Items were registered in the Change Management System. Each Configuration Item represents a business application built for and deployed on the SharePoint platform. Each business application has been developed to meet defined business requirements and usually facilitates business processes.

5.2 Initiated Changes

During the period of the study, a total of 59 Changes were initiated by application owners. At the time of writing 38 of the initiated changes were completed. The remaining 21 changes were waiting for additional actions before completion.

5.2.1 Distribution of changes over time

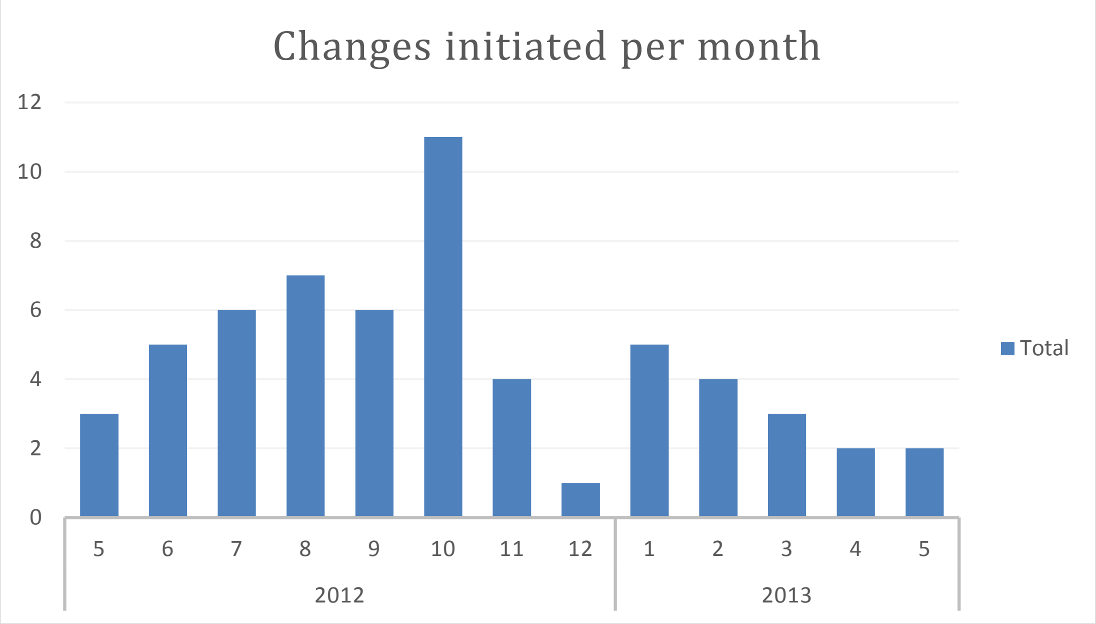

Figure 5 Changes initiated per month

The distribution of initiated changes over time is shown in Figure 5. Usually only a few changes are initiated per month, with the average being 4,75. As the number of changes is relatively low, and the required amount of work per change is relatively low, only a few persons deal with the Change Management process every day. For most stakeholders (such as application owners), the Change Management process is something they have to deal with only rarely (a few times a year). Thus, even though required and trained to follow a certain process, application owners tend to require guidance every time they need to initiate a change process.

5.2.2 Changes per Configuration Item

Changes were initiated for 31 Configuration Items. This represents a large portion of the total amount of 52 Configuration Items. Additionally 8 Changes were logged without a corresponding Configuration Item. The reasons for Changes being logged without a corresponding configuration item are shown in Table 2.

Table 2 Reasons for changes logged without corresponding CI

Reason	Number of changes
CI not yet created (new application)	3
Change opened by untrained personnel	4
CI not apparent from requirements	1

It is worth noting that changes opened directly by application owners, without the involvement of e.g. the Change Manager, tend to contain insufficient information for the implementation of the Change (such as missing information on the affected Configuration Item). This observation was made on several accounts during the course of the study. Changes being opened with insufficient information can lead to several problems. The most usual effect is the change being delayed, as additional information has to be gathered for the change to be implementable. If a change is initiated with seemingly sufficient, but technically inadequate information, it may lead to the change being implemented but failing (not meeting business requirements, creating problems in the CI or other CIs, or even creating problems in the platform).

For all Configuration Items which had changes logged during the period of this study, only a small number of changes were logged.

Table 3 Number of changes per CI

Number of changes initiated	Number of CIs
1	21
2	4
3	2
4	4

As can be seen from Table 3, for 21 of the 31 Configuration Items which had changes logged between June 2012 and May 2013, only one change was initiated. The largest number or changes initiated for a single CI was 4 (for a total of 4 CIs). The relatively even distribution of Change Request over all Configuration Items also induces, as stated before, that application owners are not familiar with the process, and are not able to effectively drive the process from the beginning to the end.

5.2.3 Change duration

The duration of a Change was measured from opening a Change to the completion of the Post Implementation Review. Additionally, the duration of changes which are not yet closed was measured until the date of writing.

The average duration of all changes closed to date was 68 days, the median duration for closed changes was 62 days. There is however a large distribution in the duration of changes. Only 4 changes were implemented from opening of the Change Request to Change Request closure in less than two weeks. 9 Changes took over 100 days to implement with the longest implemented Change taking 197 days.

The very long average duration of a Change Request can be easily explained. A Change Request should be initiated by the Application Owner, as soon as the business requirements for the change are known. After the initiation of the Change Request, the change proposal is reviewed by the Operations team, implemented by a vendor, tested in several test environments, and only then implemented to productive environments. This kind of approach is necessary to ensure that all steps of the process are completed and accepted as required by different stakeholders.

Change Requests currently still open show an even wider distribution of durations. Naturally, recently opened Change Requests only have a few days or weeks of duration to date. There are however several changes open with durations of over 300 days (2 Change Requests) and durations between 200 and 300 days (6 Change Requests). These Change Requests all represent larger development projects which are monitored by the Change Management process.

5.2.4 Change success rate

As described earlier, each Change Request is divided into a number of sub-processes, such as the Architecture Review, Code Review and Implementation to Test environment. In the Change Management System these sub-processes are called “Tasks”. Most of the tasks have a binary output: either they are successful or they fail. During an optimal Change Request process no Task has to be repeated. If necessary any Task in the Change Management process can be repeated until successful. Each revision of a Task comes however with a delay in the final implementation of the Change to productive environments.

Of the 38 Changes which were initiated and completed, only 2 went through the whole Change process without any of the Tasks related to that particular Change being repeated. All other completed Changes required repetition of at least one of the Tasks in the process.

The different types of failures which occurred during this study (including Tasks from all 59 Change Requests) are shown in Table 4.

Table 4 Types of failures in Change implementations

Type of failure	Number of affected Changes
Failed implementation to Stage	11
Failed architecture review	10
Failed code review	8
Failed implementation to Production	10
Other failures	9

Different reasons can be found for failed Tasks in a Change process. Failed architecture reviews are usually caused by insufficient information provided about the planned architecture of a new application or change to an existing application. In some cases, especially shortly after the introduction of the new Change Management process, architecture reviews were requested, but no architecture documentation was provided or even existent.

Failed code reviews were usually caused by application source code not following coding guidelines provided by the customer, code not following best practices of the SharePoint platform, application logic containing possibly dangerous code (e.g. possible major performance impact) or installation files containing filenames overlapping with previously deployed files.

Failed implementations to Stage can be divided into two main categories. The first category contains changes which fail to install as instructed (e.g. the installation ends with an error). The second category are Changes which are implemented technically, but do not meet business requirements. These changes have to go through another iteration of development, reviews and implementation to Stage.

The most alarming failures are failures in implementations to Production. The purpose of Stage implementations is to ensure that Changes can be deployed without problems. In some cases however, the implementation to Production still fails. Reasons include failed installations, misconfigured implementations and implementations causing malfunction in other parts of the system. A large number of failed implementations to Production indicates low performance of quality assurance during implementation and implementation to Stage.

Failures marked as “miscellaneous” contain different types of failures, including failed implementations to Stage or Production and failed reviews. Miscellaneous Tasks were only recorded in the early introductory phase of the Change Management processes. Later Changes do not contain Miscellaneous Tasks.

5.3 Performance of current processes

From the results presented, several conclusions can be drawn. First, the implemented processes have proven their necessity. 48 failures of different kinds were recorded during the implementation of 59 Change Requests (Table 4). 38 of these 48 failures took place before implementation to productive environment, thus not causing any harm to the operation of business critical systems. The remaining 10 failures which were recorded during implementation of a change to productive environments were relatively minor, and mostly caused disruption only in the targeted application.

Architecture reviews failed in 10 cases, meaning that the architecture had to be adjusted before allowing implementation. Failed architecture reviews probably prevented the costly implementation of some non-suitably planned applications, which would have required major changes before a possible implementation for productive use, or which might have caused problems with the platform or other applications in the future after prolonged use. In some cases a failed architecture review has also lead to some reconsideration regarding the planned application architecture, thus making possibly more robust and future proof designs possible.

Code reviews failed in 8 cases, most of which were caused by conflicting naming schemes used in the application code or inappropriate memory handling which poses a risk to all applications running on a shared platform. Problems caused by memory leaks are often hard to identify and might appear only after prolonged use. A business application which has been running for a long time often becomes business critical, making it impossible to shut down or remove an application which starts to cause problems. Thus, it is very important to identify such risks before they materialize. Possible problems caused by conflicting naming schemes are most often more immediate (e.g. overwriting of existing files), the problems might however not be visible in the newly deployed application, but rather come to light in previously deployed applications. In such case identifying the root cause of a problem can be very hard and catching these types of problems is thus very important.

Stage implementations failed in 11 cases. A failed implementation to Stage often means that instructions, installation files, or other required deliverables were not provided. Other, more critical, causes for failure of stage implementations are actual problems caused by the deployment, such as overwritten files or changed configuration causing problems in other applications.

A second conclusion which can be drawn from these results is that the implemented processes are familiar only to a small number of people. A large number of failures and delays were caused by issues in communication between different parties involved and misunderstandings regarding the expected service. In some cases the studied processes were misinterpreted as a means for ordering changes from a vendor, even though the process only aims to ensure successful deployments and changes. The familiarity of application owners with the presented processes increased greatly during the course of this study. Additionally it was found that application owners who are responsible for multiple applications, and are thus more frequently involved in the change management processes, were able to follow through the process with significantly less errors than application owners unfamiliar with the matter. Dedicated resources thus ensure reliable operation of the IT service delivered.

5.4 Further improvement opportunities

One of the more frequent problems with the presented processes was the throughput time of a change from initiation to productive implementation. Long throughput time of the process was in some cases “by design”, as the process covers the whole change cycle from initiation, through design and development (which might take months), to productive deployment. In a large number of cases however, changes could have been implemented with a faster schedule, as changes had to be delayed due to more-or-less trivially avoidable issues, such as missing instructions or other deliverables.

As suggested earlier, it was found that dedicated resources who are familiar with the processes greatly improve the performance of the process, as they tend to minimize the need for repeating certain stages of the process which tend to be challenging to resources less familiar with the matter. It would thus be important to ensure that all application owners are trained for their task and are familiar with the processes presented here, as well as familiar with other important aspects of the IT service they are responsible for. It was for example found, that resources possessing knowledge about the technical implementation of their application could handle changes as well as incidents more effectively. Work on improving the skills of application owners has been started, and has shown good results, it does however still provide the most potential for improvement.

6 Conclusions and suggestions for further research

6.1 Conclusions

In this thesis, the processes for change and deployment management of customized SharePoint applications at the case company ABB were described and evaluated. Previously no comprehensive description of the processes used in the management of the case service existed. The produced description of these processes and other material generated during the work on this thesis has been widely used to deploy the processes in the IT organization of the case company.

Additionally to describing the processes for use in training and documentation, a simplified change management process was planned and implemented during this thesis. The simplified change management process provided the possibility for more rapid deployments of changes, e.g. in cases were bugs or defects are fixed but the overall architecture of the application is not changed.

The performance of the described processes was evaluated. It was found, that the implemented processes serve their purpose as intended, catching risky changes before implementation to productive environments and making the overall service more structured and reliable than previously. It was also found that people unfamiliar with the processes form a major cause of challenges, and that training appropriate resources can improve the performance of the process by reducing the number of failures in changes.

The processes described in this thesis follow a lot of the best practices suggested by the ITIL framework, they have however been adjusted to meet the requirements of the service in question. Based on the results of the evaluation of the processes, it can be concluded that ITIL provides a good starting point for implementing IT service management processes, but also that adjusting the processes to the specific requirements of a certain service leads to better results than strict implementation of a standard process.

Overall, the implementation of standardized processes for managing the IT service which was subject of this study has proven beneficial. Some improvement opportunities remain, such as the throughput time of changes, but the overall effect of the implemented processes has been positive. The service is now very well structured and maintained, especially when compared to the situation before this study was conducted. During the upcoming months and years, new types of challenges and new requirements will most probably arise. The current situation provides a good foundation for dealing with challenges, and processes around the service could be adjusted to meet changing business requirements.

6.2 Suggestions for further research

As stated in the conclusion, the implemented processes were found to meet their respective goals by preventing potential problems and preventing business disruption. However, the number of failures and potential issues identified by the studied processes was alarmingly high.

This study focused on describing and evaluating the performance of the studied processes. For a better overall performance of the case service, it would be beneficial if the number of failures could be reduced, thus possibly improving the throughput time of the processes and the overall service performance. A study focusing on means to improve the “first time right” rate of changes and deployments could help improve the processes in this regard.

Another potential improvement possibility lies in the roles and responsibilities related to the studied processes. The impact of e.g. combining roles or appointing specialists for certain tasks which were found to be challenging for resources unfamiliar with the matter could be studied.

Existing literature dealing with IT service management processes in general was found to be very mature. Only little studies exist however on ITSM processes in complex multivendor environments, such as the environment at the case company. A study on the impact of multiple vendors working on shared processes could be conducted to better understand the overall service.

7 Bibliography

ARCILLA, M., 2008. A Solution for Establishing the Information Technology Service Management Processes Implementation Sequence. A Solution for Establishing the Information Technology Service Management Processes Implementation Sequence, 16.

BECK, SCHOTT, GREGORY, 2011, Mindful Management Practices in Global Multivendor ISD Outsourcing Projects. Scandinavian Journal of Information Systems, 2011, 23(2), pp. 5–28

BEIMBORN, D., 2009. The Role of Process Standardization in Achieving IT Business Value. System Sciences, 2009.HICSS '09.42nd Hawaii International Conference on, , pp. 1.

CANNON, D., WHEELDON, D., 2007 ITIL – Service Operation. Second impression 2007 edn. The Stationery Office.

CURTIS, B. and PAULK, M., 1993. Creating a software process improvement program. Information and Software Technology, 35(6–7), pp. 381-386.

CROSS, J., 1995, IT Outsourcing: British Petrolems Competitive Approach. Harward Business Review, May-June 1995, pp. 94-102

DAVENPORT, T.H. and SHORT, J.E., 1990. The new Industrial Engineering: Information Technology and Business Process Redesign. Sloan Management Review, 31(4), pp. 11-11-27.

DE VRIES, H.,J., 2006. Best practice in company standardization. International Journal of IT Standards and Standardization Research, Jan-June, 2006, Vol.4(1), p.62(24), 4(1), pp. 62-June,.

DEUTSCHER, J.-. and FELDEN, C., 2010. Concept for implementation of cost effective Information Technology Service Management (ITSM) in organizations, Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP 2010, pp. 167-168.

DORLING, A., 1993. SPICE: Software process improvement and capability determination. Information and Software Technology, 35(6–7), pp. 404-406.

EL EMAM, K., 2008. A Replicated Survey of IT Software Project Failures. Software, IEEE, , pp. 84.

FRY, M., 2010. ITIL Lite - A road map to full or partial ITIL implementation. First edition edn. London: The Stationery Office.

GALUP, S.D., DATTERO, R., QUAN, J.J. and CONGER, S., 2009. An Overview of IT Service Management.

GLASS, R., 2005. IT Failure Rates--70% or 10-15%? IEEE Software, 22(3)

HALL, J., 2002. IT Groupware services--making them effective. Management Services, 46(7), pp. 22-23.

HIRSJÄRVI, S., REMES, P. and SAJAVAARA, P., 1997, Tutki ja Kirjoita, Gummerus Kirjapaino Oy

JIN, K., 2008. Business-Oriented Development Methodology for IT Service Management. Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, , pp. 99.

JONES, D.T., 2008. Rethinking IT. Management Services, 52(2), pp. 36-37.

KAZLAUSKAS, E.J. and WEIR, A.D., 1994. Information services for innovative organizations. San Diego, Calif: Academic Press.

KEIL, M., 1998. A framework for identifying software project risks. Association for Computing Machinery.Communications of the ACM, 41(11), pp. 76-83.

LACY, S. and MACFARLANE, I., 2007. ITIL - Service Transition. Second impression 2007 edn. The Stationery Office.

LAHTELA, A., JANTTI, M. and KAUKOLA, J., 2010. Implementing an ITIL-Based IT Service Management Measurement System, Digital Society, 2010. ICDS '10. Fourth International Conference on 2010, pp. 249-254.

LEE, R.P. and KIM, D., 2010. Implications of service processes outsourcing on firm value. Industrial Marketing Management, 39(5), pp. 853-861.

MCMANUS, J. and WOOD-HARPER, T., 2007. Understanding the Sources of Information Systems Project Failure. Management Services, 51(3), pp. 38-43.

MÜNSTERMANN, B., ECKHARDT, A. and WEITZEL, T., 2010. The performance impact of business process standardization. Business Process Management Journal, 16(1), pp. 29-56.

PFITZINGER, B. and JESTADT, T., 2011. Orchestration of service design and service transition, Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on 2011, pp. 541-544.

TAYLOR, S., 2004. The way to get IT right. Management Services, 48(4), pp. 22-23.

VAN BON, J., DE JONG, A., KOLTHOF, A., PIEPER, M., ROZEMEIJER, E., TJASSING, R., VAN DER VEEN, A. and VERHEIJEN, T., 2007. IT Service Management - An Introduction. First Edition edn. Zaltbommel: Van Haren Publishing.

WULLENWEBER, K., 2007. An Empirical Exploration of How Process Standardization Reduces Outsourcing Risks. System Sciences, 2007.HICSS 2007.40th Annual Hawaii International Conference on, , pp. 240.

YAO, Z., 2010. An ITIL based ITSM practice: A case study of steel manufacturing enterprise. Service Systems and Service Management (ICSSSM), 2010 7th International Conference on, pp. 1.

YIN, R., 2011, Applications of Case Study Research, SAGE

Appendices

Appendix A

SharePoint custom application kick-off agenda

To be invited (minimum):

- FI ABB:
    - Service Manager
    - Change coordinator
- LBU:
    - Application owner or Project manager
- Vendor:
    - Technical representative (e.g. architect)
- Collaboration team:
    - Onboarding manager
    - Operations manager
    - Architect

Agenda (total time: max. 1 hour):

1. Introduction (everyone)
2. Short, verbal description of the business purpose of the application (Application owner)
    a. Functionalities
    b. Users (external, internal, etc.)
3. Technical description of the **planned** solution (Vendor)
    a. Rough architecture description
    b. Possible integrations
    c. Possible internal (inside SharePoint) and external dependencies
    d. Used Service Applications (e.g. Search, Managed Metadata, etc.)
    e. Solution type (Farm/Sandboxed)
    f. Questions to Collaboration Team (if any, from Vendor)
    g. Questions to Vendor (if any)
4. General information (FI ABB, if necessary)
    a. Show location of guidelines and documents
        i. Architecture guideline
        ii. Coding guideline
        iii. Templates
        iv. Environment descriptions
        v. Process description

5. Questions and answers

Note: This meeting should be a discussion. Vendor will prepare Architecture description for review AFTER this meeting. No lengthy presentations should be held.