ITIL Change Management

Nuno António Garcia Amarante

Dissertação para obtenção do Grau em Mestre em

Engenharia Informática e Computadores

Júri:

Presidente: Professor José Manuel Tribolet
Orientador: Professor Miguel Mira da Silva
Vogal: Professor José Delgado

Outubro 2009
Acknowledgements

First and foremost I would like to express my gratitude to my advisor for his patience and support throughout the realization of this thesis. I also thank everyone of the public organization that made this thesis possible. Also to my colleagues for all the cooperation and support we gave each other this last year, especially Diogo Nesbitt for his guidance and good advices.

Finally to my friends and specially my family for all the support and presence not only this last year but throughout my life.
Abstract

IT Service Management standards, like the IT Infrastructure Library and ISO 20000, are of increasing importance to organizations around the globe because nowadays businesses have high expectations towards the quality of services and these expectations change over time. ITIL is a set of best practices and although ITIL do describe what organizations should be doing it doesn’t say how. Little or no research exists on how to implement ITIL. An analysis of the scarce research to the date is made, and then we propose to thoroughly attack a dimension of this problem using Action Research methodology: discover how to really implement ITIL Change Management in medium sized organizations.

Keywords

IT Service Management, ITIL, IT Change Management
Resumo

Normas de gestão de serviços de IT, como o ITIL e o ISO 20000, são cada vez de maior importância para qualquer organização. Isto porque hoje em dia o negócio tem altas expectativas sobre a qualidade dos serviços e estas expectativas mudam ao longo do tempo. O ITIL é um conjunto de boas práticas que, apesar de descrever o que as organizações deveriam estar a fazer, não descrevem como. Pouca ou nenhuma investigação existe sobre como implementar o ITIL. Uma análise da pouca investigação que existe até à data é feita, e depois propomos abordar de forma completa uma dimensão deste problema usando a metodologia Action Research: descobrir como efectivamente implementar a gestão de alterações do ITIL em pequenas e médias organizações.

Palavras-chave

Gestão de Serviços de IT, ITIL, Gestão de Alterações do IT
Contents

Acknowledgements ......................................................................................... i
Abstract ........................................................................................................ ii
Keywords ....................................................................................................... ii
Resumo .......................................................................................................... iii
Palavras-chave .............................................................................................. iii
Contents ........................................................................................................ iv
List of Figures .............................................................................................. vi
List of Tables ............................................................................................... viii
List of Acronyms ............................................................................................ ix

1. Introduction ................................................................................................ 1
   1.1 Context .................................................................................................. 1
   1.2 Problem ............................................................................................... 2
   1.3 Research Methodology ......................................................................... 4
   1.4 Thesis Structure .................................................................................. 5

2. Related Work ........................................................................................... 7
   2.1 ITIL ...................................................................................................... 7
       2.1.1 IT Service Management .............................................................. 7
       2.1.2 What is ITIL? ........................................................................... 7
       2.1.3 V2 and V3 ................................................................................ 9
       2.1.4 ISO 20000 ............................................................................... 9
       2.1.5 ITIL Implementation ............................................................... 10
   2.2 IT Change Management ...................................................................... 11
       2.2.1 What is Change Management? .................................................. 12
       2.2.2 IT Change Management Implementation ................................ 13
       2.2.3 Other Related Work .................................................................. 18
       2.2.4 Commercial offers .................................................................... 18

3. Proposal ..................................................................................................... 23
3.1 Organizational context ................................................................. 24
3.2 Iterative implementation with Agile Development............................. 24
3.3 Diagnosing phase ....................................................................... 25
3.4 Action planning phase .................................................................. 25

4. Action Taking ................................................................................. 29
  4.1 Technology ............................................................................... 29
    4.1.1 Outsystems Platform .......................................................... 29
    4.1.2 Development Methodology ............................................... 30
  4.2 Configuration Management Application ......................................... 31
    4.2.1 Domain Model ................................................................. 31
    4.2.2 Functionalities ................................................................... 34
    4.2.3 Interface ............................................................................ 34
    4.2.4 Integration .......................................................................... 36
  4.3 Change Management Application .................................................. 37
    4.3.1 Domain Model ................................................................. 37
    4.3.2 Functionalities ................................................................... 38
    4.3.3 Interface ............................................................................ 39

5. Evaluation ..................................................................................... 43
  5.1 Evaluating phase ....................................................................... 43
  5.2 Specifying Learning phase ......................................................... 43
    5.2.1 Change Record and Process Description ................................ 43
    5.2.2 Change Decision ............................................................. 47
    5.2.3 CI attributes change and logging ........................................ 48

6. Conclusion .................................................................................... 53
  6.1 Future Work ............................................................................. 54

References .......................................................................................... 55
List of Figures

Figure 1.1 – Domain model of the IT of an organization (IT assets management data)..................3
Figure 1.2 – The Action Research Lifecycle [16]...........................................................................4
Figure 2.1 - Core components of ISO/IEC 20000 [28].................................................................10
Figure 2.2 – Change Management and other ITIL Processes [34]..................................................12
Figure 2.3 – Change Management Activities [34]........................................................................14
Figure 2.4 – Lifecycle of hardware CI proposal [35]....................................................................15
Figure 2.5 – Proposed States of Change [7]................................................................................15
Figure 2.6 – Perceptions about service quality improved over time [8]........................................17
Figure 3.1 – IT Department Organizational Chart.......................................................................24
Figure 3.2 – The Action Research Lifecycle................................................................................24
Figure 3.3 – Encountered Change Management Process ...............................................................26
Figure 3.4 –CMA Workflow Configuration..................................................................................27
Figure 4.1 – The ITIL Project Component Diagram.....................................................................29
Figure 4.2 – Outsystmes Platform Overview................................................................................30
Figure 4.3 - Traditional Waterfall model vs agile development [38]...............................................31
Figure 4.4 – Previous Configuration Management Domain Model (partial)..................................32
Figure 4.5 – Configuration Management CI Entities.....................................................................32
Figure 4.6 – Configuration Management Application Main Entities..........................................33
Figure 4.7 - View of the Hardware CI Tower-1............................................................................34
Figure 4.8 – Editing Hardware CI Tower-1..................................................................................35
Figure 4.9 – View of Server CI Server 001....................................................................................35
Figure 4.10 – Editing Server 001 Relations...................................................................................36
Figure 4.11 - Add/Remove references screen...............................................................................36
Figure 4.12 – Change Management Application Domain Model................................................37
Figure 4.13 – Change Management Application (CMA) main interface....................................39
Figure 4.14 - Change Record attributes default view.................................................................40
Figure 4.15 – Configuration Item Selection List.........................................................................41
Figure 4.16 – Workflow section ........................................................................................................41
Figure 4.17 - Change Wizard .........................................................................................................41
Figure 5.1 – Proposed Change Management Workflow ................................................................45
Figure 5.2 - The priorities of the Change ......................................................................................47
Figure 5.3 - Configuration Item lifecycle ....................................................................................49
Figure 5.4 - Edit CI attributes screen ........................................................................................49
Figure 5.5 - Change Log tab .........................................................................................................50
Figure 5.6 – Changes Summary Tab ............................................................................................51
Figure 6.1 - Traditional Waterfall model vs agile development [38] ..............................................53
List of Tables

Table 2.1 - Core components of ITIL Service Support [11]. .................................................................8
Table 2.2 - KPIs for ITIL processes Change Management and Release Management [10]. ..............17
Table 5.1 – Change Category to Change Privilege Correspondence......................................................48
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMA</td>
<td>Change Management Application</td>
</tr>
<tr>
<td>CMDB</td>
<td>Configuration Management Database</td>
</tr>
<tr>
<td>CMDBA</td>
<td>Configuration Management Database Application</td>
</tr>
<tr>
<td>CR</td>
<td>Change Record</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>ITIL</td>
<td>Information Technology Infrastructure Library</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Change</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Context

Motivation

A usual scenario for any organization: the IT service desk suddenly receive multiple calls from users reporting that a computer system is acting in a not expected way, hindering their work. The service desk staff research the situation but find no previous record of the indicated behavior, so they forward the incident report on to the system developers. They report back that a change had just been implemented, and that the system is not supposed to be working that way. Or actually it would eventually be discovered that another system had been changed, and that their alteration had had an effect on this system [1]. It’s too easy to make a seemingly innocuous change that take others hours to figure out and undo because it was attached to something that then in turn broke something else [2].

Nowadays business requirements change speed and the necessity for readiness to answer the ever changing conditions of business, demand IT to be capable of adapting itself rapidly and certainly without moving to a state where it does a worst job than it was doing before. Change then, must be moving from one state to another, but not just another, to something better. Better as in what the customer needs and what he expects to find.

Area

These days IT Management is focusing particularly on the ‘de facto standard’ Information Technology Infrastructure Library (ITIL), for implementing IT Service Management (ITSM). In doing this, deficits of the ITIL reference model are often being overlooked, benefits are merely assumed and misunderstandings spread [3]. Cater-Steel asserts that ITIL is not a complete approach in that it lacks a specific maturity model and a measurement system for process improvement [4]. Also ITIL can only be successfully implemented by understanding how those best practices apply, based on the IT organization’s targets, strengths and abilities [5]. Even generally speaking of frameworks, that best practices and standards, it is true that they are focused on logical level of processes which instruct what should be done. Also poor definition of information models corresponding to process descriptions is another ITIL weakness [6]. Even another author asserts that ITIL is simply not prescriptive about implementation [7].

Potgieter defends that although the OGC claims that the use of ITIL improves customer satisfaction and Service Quality, this was not scientifically proved and published to date, but rather inferred [8]. The same study alleges to be the first one to conclude that, customer satisfaction and operational performance improve as the activities in the ITIL framework increase (more on this study on section 2.1.5 ITIL Implementation). We further question that the several variances and options that an
organization can undertake when implementing ITIL will have a strong influence on customer satisfaction and quality service. Additionally to what Potgieter states, we ascertain that further study is needed to find what consequences the different paths taken in ITIL implementation have on customer satisfaction and quality of service [8].

Furthermore, a large array of authors state that there is little or no published research that concerns ITSM [3, 8-12]. There is a world of knowledge, secretly possessed by Consulting Firms and Organizations, but that knowledge is contained there, in exclusivity, and is not based on worldwide scientific construction. There is some research related to areas like automatic scheduling of IT Changes, but none concerning the actual way to implement and do IT Change Management. There are studies that conclude that a majority of organizations give priority to implement Change Management [13] but there are even less research about the actual implementation of this particular process.

1.2 Problem

Controlling and regulating the change of IT assets in any organization is recognized to be of utter importance because, simply put, it is very easy to make a seemingly innocuous change that takes others hours to figure out and undo because related components were affected in an unexpected and unplanned manner, compromising the service availability [2].

Nowadays the fact that businesses have high expectations towards the quality of services and these expectations change over time, demands that IT must capable of adapting itself rapidly and certainly without moving to a state where it does a worst job than it was doing before.

The IT assets of an organization consist of a very strongly connected network, in which any change can have unforeseen repercussions on any other part of the network. Figure 1.1 depicts an example domain model of an organization.
Figure 1.1 – Domain model of the IT of an organization (IT assets management data)

This example illustrates how complex and difficult to navigate and understand such network is. The partially depicted domain model is actually the domain model of the Change Management Database that was already used previously to the beginning of this thesis, in the target organization where this thesis was undertaken.

A study by Cater-Steel on ITIL implementation concludes that [13]:

- There exists substantial variation among the organizations in ITIL implementations
- Change Management is consistently treated as a high priority
- Incident, Problem and Configuration Management are also early choices
- Sequence of adoption of IT Processes are dictated by specific business strategy and benefits sought
- In some cases, previous attempts had been abandoned due to lack of senior management commitment and resources

From these several points we would like to underline the first one, which states that a substantial variation exists among the organizations in ITIL implementations. ITIL is a set of best practices and although ITIL does describe what organizations should be doing it doesn’t say how. ITIL isn’t actually a set of best practices it’s truly a set of common general practices, from which organizations must choose and adapt to better fit their specific circumstances.

It’s important to note that there is indeed a world of knowledge about ITIL implementation, but those ‘inside stories’ are hard to get as Spremich states [10].

It’s a fact that IT must be capable of adapting itself rapidly and certainly without moving to a state where it does a worst job than it was doing before. Change then, must be moving from one state to another, but not just another, to something better.
Being so the problem of this thesis is: in spite of its recognized importance, it is not known how to implement ITIL Change Management in any context of an organization. This is because there is little or no research on implementing ITIL and even less specifically on ITIL Change Management.

1.3 Research Methodology

Qualitative research methods were developed in the social sciences to enable researchers to study social and cultural phenomena. In Information Systems research there has been a general shift away from technological to managerial and organizational issues, hence increasing interest in the application of qualitative research methods [14].

Action research encourages researchers to experiment through intervention and to reflect on the effects of their intervention as well as the implication of their theories. It contrasts with classic quantitative research methods because of the need of researchers to understand the ill-structured, fuzzy world of complex organizations. [15].

Besides reading extensively about a subject, observing systems working in organizations, building case studies and devising a theory, this methodology suggests that a researcher should try out the theory with practitioners, in real situations, gain feedback from this experience, modify the theory as a result of this feedback, and try it again. Each iteration of the action research adds to the theory improving it [15].

Action Research details a five phase cyclical process depicted in Figure 1.2. The approach first requires the establishment of a client-system infrastructure or research environment.

![Figure 1.2 – The Action Research Lifecycle [16]](image-url)
• **Diagnosing:** The first step is diagnosing in which the researcher identifies the visible problems in the organization, responsible for the organizations desire to change. It’s in this phase that a theory about the nature of the problems is developed.

• **Action Planning:** Researchers and practitioners collaborate in this phase in which the specification of the actions needed to solve the first encountered problems takes place.

• **Action Taking:** Implements the planned action. The researchers and practitioners collaborate in the active intervention in the organization.

• **Evaluating:** After the action is concluded, the researcher and the practitioners evaluate the results.

• **Specifying Learning:** In this phase the knowledge acquired throughout the whole cycle is specified.

### 1.4 Thesis Structure

In the first chapter, Introduction, the general context of this thesis is laid out, by setting the Motivation and Area of the thesis and also by explaining the problem and research methodology.

The second chapter, Related Work, identifies the context of the work in the international scientific community, and also explains the main concepts of the thesis area.

The third chapter, Proposal, presents the initial proposal, the starting point of the work, and how it relates to the executed work, this chapter contains the Diagnosing and Action Planning phases of the Action Research cycle.

The fourth chapter, Action Taking, describes the nominated Action Research cycle, consisting of the general architecture of the solution, for a better understanding of the underlying proposal and the contributions.

The fifth chapter, Evaluation, describes the two final Action Research cycles, the Evaluating and Specifying Learning.

Finally, the sixth and last chapter, Conclusion, consists of the final main conclusion of the thesis and proposals for future work.
2. Related Work

In this chapter we present the concepts related to Change Management, its implementation, and the context in which this work is developed.

However this chapter is not as extensive as it should be expected to be, because there is little or no scholarly work on this area, [3, 8-12] and also most academic institutions appear to be reluctant to include ITSM in their curriculum [17]. This despite the fact that a great array of commercial solutions and software are offered by several companies [18], despite the significant growth of ITSM practice in the industry [9] and despite the obvious challenges associated with adoption and implementation of ITIL [11]. The majority of publications about ITIL implementation itself is in a context of university IT departments [3, 19-21], which further indicates this division between the academic universe and other organizations. There is a world of knowledge about ITIL implementation, but that ‘inside stories’ are hard to get as Spremic states [10].

2.1 ITIL

2.1.1 IT Service Management
ITSM is a subset of the Services Science discipline that focuses on IT operations delivery and support [9]. In the past, many IT organizations were internally focused and concentrated on technical issues. Nowadays businesses have high expectations towards the quality of services and these expectations change over time. This means that for IT organizations to live up to these expectations they need to concentrate on service quality and a more Customer oriented approach [22]. ITSM is often simply associated with ITIL [9]. The first international Standard on ITSM is ISO 20000.

2.1.2 What is ITIL?
The Information Technology Infrastructure Library (ITIL) is a set of documents consisting of several volumes [22-24] of IT Management concepts, processes and methods, currently administrated and developed by the IT service provider of the British government, the Office of Government Commerce\(^1\) and is internationally promoted by itSMF\(^2\). ITIL consists of a framework of best practices intended to facilitate the delivery of high quality services. It documents industry best practice guidance and, being a framework, it describes the contours of organizing Service Management.

ITIL is becoming one of the most widely accepted approaches to IT Service Management in the world [3]. ITIL is being widely adopted by organizations throughout the globe, and has become a ‘de facto standard’ in the arena of IT Service Management. [18, 25].

---

1 http://www.ogc.gov.uk/
2 http://www.itsmfi.org/
ITIL books are meant as guidance that can be adapted to suit a specific environment. What ITIL provides is a common vocabulary and common processes and philosophies of operation that can bridge the gap between IT organizations. It also provides the much needed discipline, especially in areas such as Change Management: it is very easy to make a seemingly innocuous change, that takes others hours to figure out and undo, because related components were affected in an unexpected and unplanned manner, compromising the service availability [2]. The Configuration Management Database (CMDB) is something that underlies all the processes, it is where the usual asset management data resides, but the CMDB also holds the sources on information on the resources used by each service and their dependencies. When a Change needs to be executed, the CMDB will show which components are attached to the altered component or service so that any consequences and problems associated to the change are always known [2]. The CMDB stores Configuration Items (CIs) which can be any component that needs to be managed in order to deliver an IT Service. CIs typically include IT Services, hardware, software, buildings, people and formal documentation such as Process documentation and Service Level Agreements [23].

The two primary components of the ITIL v2, service delivery and service support, consist of core processes, listed and briefly described in Table 2.1, that IT organizations must put in place in order to provide quality IT services for their customers. Change Management, the focus of this thesis, is one of these processes.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Desk</td>
<td>The single point of contact for service providers and users.</td>
</tr>
<tr>
<td></td>
<td>Manages incidents, service requests, and handles communication with the users.</td>
</tr>
<tr>
<td>Incident Management</td>
<td>Manages the lifecycle of all incidents; restores normal service operations as quickly as possible.</td>
</tr>
<tr>
<td>Problem Management</td>
<td>Manages the lifecycle of all problems: prevents incidents from happening and minimizes the impact of incidents that cannot be prevented.</td>
</tr>
<tr>
<td>Change Management</td>
<td>Controls the lifecycle of all changes: enables beneficial changes to be made with minimum disruption to IT services.</td>
</tr>
<tr>
<td>Release Management</td>
<td>A collection of hardware, software, documentation, processes or other components required to implement approved changes to IT services.</td>
</tr>
<tr>
<td>Configuration Management</td>
<td>Maintains information about configuration items required to deliver an IT service, including their relationships.</td>
</tr>
</tbody>
</table>
There is also another study that has used a KPI based approach to empirically prove and compare the results of an ITIL implementation, comparing the results before and afterwards the implementation (more on this study in the end of sub-section 2.2.2 in Metrics and management reporting). The summary of the results were the identification of direct and indirect benefits from the evaluated ITIL implementations [10]:

- Implementation of ITIL improved the quality of IT Services which the Company provides to customers.
- After an ITIL process implementation, day to day work procedures were improved. People get concrete tasks and roles in the process. The use of the knowledge base helps them solve most incidents on first level support and let specialist free of simple and repeating issues.
- Management is a key point of ITIL implementation.
- Before buying software for automation processes, IT has to make thorough IT audits, which means deep feasibility and functionality analyses of the software and its possibility to integrate with current IT infrastructure.

### 2.1.3 V2 and V3

ITIL v3 pares the paper down to five main volumes, plus an introductory book, and rearranges the processes to reflect the IT service lifecycle [2].

The differences between V2 and V3 reflect changes in business focus over the years, for example [2]:

- Where V2 talked about business and alignment, V3 emphasizes business and IT integration.
- Where V2 talked about value chain management, V3 emphasizes value network integration.
- Where V2 talked about linear service catalogue, V3 emphasizes dynamic service portfolios
- Where V2 talked about collection of integrated processes, V3 emphasizes holistic service management lifecycle

Several authors, like Spremic, defend that the ITIL V3 processes remain the same in its core as in ITIL V2 [10].

### 2.1.4 ISO 20000

The first international standard for IT Service Management is ISO 20000 and, effectively serves as the basis for certification audits and assessments of how well ITIL advice has been adopted [26]. It is divided into two parts: ISO 20000-1 [27] and ISO 20000-2 [28]. It consists of thirteen processes allocated across 5 areas as indicated in Figure 2.1.

- Part 1: specification. Promotes adoption of an integrated process approach to effectively deliver managed services to meet the business and customer requirements [27].
Part 2: Code of practice. Provides guidance and recommendations based on industry consensus to service providers planning service improvements and/or seeking to be audited against [27].

Figure 2.1 - Core components of ISO/IEC 20000 [28].

ISO 20000 was based on BS 15000, which in turn is fundamentally the British Standard for ITIL, BS 15000. It took less than 14 months from the submission of BS 15000 to become ISO 20000. This highlights the importance of an international standard for service management [26].

ISO 20000 creation further demonstrates the importance and relevance that ITSM has. The creation of an international standard supports and influences organizations to improve IT Service Management, implementing ITIL, to obtain accreditation. ISO 20000 will shift the emphasis from certification of individuals to the certification and audit of organizations. [29]. Cater-Steel goes further declaring that ITIL has evolved from a company standard to it is rectification as an International Standard, affirming that as such, growth adoption is guaranteed to accelerate.

2.1.5 ITIL Implementation

Implementation of ITIL Projects varies between smaller and larger organizations. The context of this work is small and medium sized organizations.

Research indicates that for large organizations with high-risk projects, serious management is required and a formalized approach is necessary [30]. For small organizations, a more adhoc process is used and usually depends on the type of customers and projects, and on team leaders or managers. Attempts are made to keep processes simple and efficient, minimize paperwork, promote computer-based processes and automated tracking and reporting, minimize time required in meetings and promote training.

A Cater-Steel study raises the question whether ITIL should be implemented as a project or business as usual (seen as continuous process improvement) [11]. Cater-Steel finds some advantages and
disadvantages in both, but concludes that the institutional context is important for this choice, specifically relative level of institutional support from senior management, historical factors in relation to previous projects and the reasons motivating ITIL adoption.

Cater-Steel has developed a unique and extensive 2 years research of ITIL Adoption in Australia, through several surveys and case study analysis [4, 31, 32]. The final critical success factors identified are [13]:

- Commitment from senior management
- Champion to advocate and promote ITIL
- Ability of staff to adapt to Change
- Quality of IT staff allocated to ITIL
- ITIL training for IT staff

Cater-Steel also concludes that concerning ITIL implementation that:

- There exists substantial variation among the organizations in ITIL implementations
- Change Management consistently treated as a high priority
- Incident, Problem and Configuration Management also early choices
- Sequence of adoption of IT Processes are dictated by specific business strategy and benefits sought
- In some cases, previous attempts had been abandoned due to lack of senior management commitment and resources

Other author [5] also indicates that these are the main reasons for ITIL implementation failure:

- Lack of management commitment
- Spending too much time on complicated process diagrams
- Not assigning process owners
- Concentrating too much on performance while ignoring quality
- Being too much ambitious (e.g. implement all or many ITIL processes at once)
- Failing to maintain momentum and allowing departmental demarcation

2.2 IT Change Management

In this section we first present a brief description of the main concepts of IT Change Management and then a summary and discussion of relevant research out of the very few scientific publications focused on IT Change Management.

[33] research shows that 80 percent of mission-critical application service downtime is directly caused by people or process failures (“unmanaged changes”). The other 20 percent is caused by technology
failure, environmental failure or a disaster. The complexity of today’s IT infrastructure and applications makes high availability systems management enormously difficult.

The basic concepts of Change Management are principally process related and managerial, rather than technical (whereas Incident Management is primarily technical, with a strong emphasis on the mechanical nature of some of the processes) [22].

2.2.1 What is Change Management?
As ITIL puts it: change is the process of moving from one defined state to another [22]. Change Management is intended to enable organizations to implement an effective and efficient process to identify, plan and manage changes to their infrastructure. It should provide users with functionality to identify and mitigate risk associated with changes so that they can be implemented with confidence.

ITIL Service Support book states that Change Management ensures that standardized methods and procedures are used for efficient and prompt handling of all changes, in order to minimize the impact of change-related incidents upon service quality, and consequently to improve the day-to-day operations of the organization [22]. Figure 2.2 shows the interfaces of Change Management Process with other ITIL Processes.

![Diagram of Change Management and other ITIL Processes]

**Figure 2.2 – Change Management and other ITIL Processes [34]**

According to ITIL, Change Management is responsible for managing the Change processes involving:

- Hardware
- Communications equipment and software
- System software
- ‘live’ application software
- All documentation procedures associated with the running, support and maintenance of live systems
Furthermore, changes to any components that are under the control of projects are subject to project Change Management Procedures, not under general Change Management procedures. The Change Management team will, however, be expected to liaise closely with project managers to ensure smooth implementation and consistency within the changing management environments.

It is the Change Management process that produces approval (or otherwise), for any proposed Change. While Change Management makes the process happen, the decision authority is the Change Advisory Board (CAB), which is made up for the most part of people from other functions within the organization [22].

2.2.2 IT Change Management Implementation

Activities

Figure 2.3 shows the activities involved in ITIL Change Management Process.

There is a research that proposes an interesting and somewhat detailed approach to the Change Management activities. The author states that it is only similar to the ITIL version but had to be adapted and tailored specifically for the business needs of the respective organization of the Case Study. A summary of the process follows [35]:

1. Request for Change: change originator requests for a change by using a web form in the intranet or by sending e-mail to the Service Desk. Change initiator can be anyone. RFC (Request for Change) ticket is created;
2. Change Coordinator reviews the RFC;
3. Change Coordinator evaluates the RFC, classifies it, and categorizes it based on the impact and urgency. Change Coordinator does the initial planning of the change implementation by describing the change and by assessing the change necessity and feasibility;
4. The change request is approved for implementation either by the Change coordinator or the Change Advisory Board (CAB). Need for CAB approval depends on the change impact for the business, the scope of the investment, the amount of resources needed for implementation, and the amount of risk related to the change implementation;
5. Change Coordinator plans the change implementation by defining the tasks needed to complete the implementation, the work effort, resources, schedule budget and acceptance criteria. Change implementer plans the change implantation from technical viewpoint by defining technical solution, test and back out plans.
6. The change is implemented according to the implementation plan and “Build and Test” and “Release to production Methods”
   - “Build and Test” is the change preparation process where impact of the change is verified and also what will be done and how. It also includes testing if feasible.
   - “Release to Production” is the process of taking changes to production. It also involves defining if the change can be considered successful or not and deciding if implementation of back-out plan is needed.

7. The change implementation success and impact are reviewed in cooperation with the Change Originator. The RFC ticket and related information is updated, and the change ticket closed.
Configuration Item Lifecycle

ITIL suggest only some examples of the status of an RFC ‘logged’, ‘assessed’, ‘rejected’, ‘accepted’, ‘sleeping’. Mattila also proposes a Lifecycle is thesis [35]. He suggests the lifecycle of a Configuration Item to be as illustrated in Figure 2.4. Note: the state lend is when the CI is used by a subcontractor or minority shareholder but maintenance is done by the IT Department.

![Figure 2.4 – Lifecycle of hardware CI proposal [35]](image)

Other author still suggest the states depicted in Figure 2.5 [7]. Ward introduces the concept of protected stages. In these stages any change made do the Configuration Item necessitates that an RFC be associated with them. This validation capability recognizes explicitly that there are life-cycle states in which a greater degree of control is required over the way in which they can be modified.

![Figure 2.5 – Proposed States of Change [7]](image)

Change Advisory Board

[35] also describes yet another variant on the ITIL Change Management process: the organization of the Change Advisory Board (CAB). It consists in the Back End Services team of the data center in question. CAB meetings are held as a part of regular BES meetings. If authorization from Business
application owner is needed, CAB meeting participants are considered separately. In this case CAB consists of IT Managers, Business IT Managers, CI Owners and Change Implementers from BES team, depending on the change request. These meetings can be held as a part of business application control meetings.

Emergency Change Advisory Board (ECAB): Accepts major changes which are prioritized as critical and which could cause substantial harm to business if not applied immediately. ECAB consists of service key user, service owner and if service has related services their services owners. ECAB members are contacted by email or in extremely urgent cases by phone. If a person is unavailable his superior is contacted for approval of the change. After emergency implementation change returns to normal Change Management process.

Other research asserts that too few people in the CAB make it easier to meet, but may exclude some of the essential stakeholders of a particular change [36]. So Dorst proposes the creation of several area CABs. This has the benefit that it is easier to accommodate existing decision and communication forums. The logical separation between the areas (which aligned with the business model) made it easier to appoint the correct people in these roles, both from a management (authority) and an acceptance perspective. In order to overcome the silo-structure that was created (albeit business rather than technology focussed), they defined a new role, the change Controller, someone who specifically (horizontally, across all CABs) would review, authorize, coordinate and control the changes from a technical (IT) perspective.

Change Categories

ITIL Change Management process suggests that change can be categorized in one of four categories: standard, minor, major and significant. [37] states that by doing this the process is fundamentally flawed because the most seemingly trivial things can have a monumental impact on a system and the trivial is governed by perception, so if people believe something to be trivial they will treat it as such.

[35] divides change into 3 different categories: Normal, Major and Emergency Change.

However, not every request for modification is treated as a change. Daily routine tasks that have a small impact on business are handled as Service Request tickets. Service requests are handled by Helpdesk or 2nd level specialists and the tasks are conducted by procedures and related instructions created by the 2nd level support, IT service key users or IT service owners who are also responsible for training them to the Helpdesk and 2nd level support specialists.

Normal changes are planned and implemented by a predefined procedure. Typically these changes are implemented by IT 2nd level support. Change is applied according to predefined procedures (instructions how to apply change), which have been created by the 2nd level support or the IT service key use. The procedures have to be pre-approved by the IT service owner. Normal changes are repetitive with known outcomes and known staff who are authorized to implement.
Any change which is not a Normal change that will or has the potential to interrupt multiple business critical services is a Major Change. All Major changes have to be approved by the CAB prior to detailed planning and implementation. Some major changes have such a big impact on the business that CAB has to escalate them to the IT board for an approval. Finally a Major change is considered an Emergency change if it may cause a severe risk for the service continuity if not implemented immediately.

**Metrics and management reporting**

There is a study used a KPI based approach to empirically prove and compare the results of an ITIL implementation, comparing the results before and afterwards the implementation. A summary of the results for the Change and Release Management are shown on Table 2.2.

<table>
<thead>
<tr>
<th>KPI</th>
<th>Before ITIL Implementation</th>
<th>After ITIL Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of changes which are realized as planned</td>
<td>25%</td>
<td>80%</td>
</tr>
<tr>
<td>% of released changes but not approved</td>
<td>10%</td>
<td>95%</td>
</tr>
<tr>
<td>% of urgent changes</td>
<td>60%</td>
<td>35%</td>
</tr>
<tr>
<td>% of unsuccessfully realized changes</td>
<td>18%</td>
<td>6%</td>
</tr>
<tr>
<td>% of used software which are unauthorized</td>
<td>22%</td>
<td>8%</td>
</tr>
<tr>
<td>% of wrong releases</td>
<td>13%</td>
<td>10%</td>
</tr>
<tr>
<td>% or urgent releases</td>
<td>32%</td>
<td>20%</td>
</tr>
</tbody>
</table>

![Table 2.2 - KPIs for ITIL processes Change Management and Release Management [10].](image)

Figure 2.6 – Perceptions about service quality improved over time [8].
These surveys were based on a Customer Satisfaction Survey that is applicable to Information Systems Field. Although the survey is in essence a quantitative measurement device, qualitative statements are associated with the rating scale.

Other author proposes the following metrics [37]:

- Number of changes within a specified period broken down by type
- Percentage of changes completed on schedule
- Percentage of changes completed within budget
- Number of aborted changes within a specified period
- Average duration broken down by type
- Number of incidents attributed to changes completed within a specified period

### 2.2.3 Other Related Work

Mainstream management information systems research is dominated by the empirical variable approach – gathering data across groups and organizations and linking variables by linear models using statistical methods [9]. Examples of this research is decision support tools to optimize scheduling of changes, configuration complexity model and its application to a change management system, automating the change management process with electronic contracts, which are all very low level, and heavily empirical, mathematical and algorithmic based.

Galup suggests that an alternative to the mainstream variable approach is the process approach [9]. The argument for using the process approach for ITSM research is that it would facilitate the creation of focused set of paradigms, greater understanding of how things work in IT Organizations, and hopefully more conclusive results to research studies.

### 2.2.4 Commercial offers

Consulting firms and software companies offer many different solutions and implementations that are not shared with the scientific community. Following is a brief analysis made by Forrester Research [18]. This analysis is important and relevant as it outlines not only what are the characteristics of such a system or solution but also in what ways and tools they are present in the current state of the art.

In the study there is a comment that gives special relevance to the importance of Change Management, it is referred that the process of managing change to the infrastructure is compounded by size, complexity, business impact, and new regulatory requirements, and that modelling such workflows with the proper approval and change controls requires industrial-strength process and workflow management tools that allow secure, auditable and controlled processes.

A summary of the more relevant findings of the study follows. It is important as it can be useful to further be aware of what problems might rise when implementing Change Management.
Large or complex organizations

Tools appropriate for large organizations are characterized by their ability to:

- Handle large call volume: Organizations with 5000 to 10000 or more employees can generate upwards of 100000 individual tickets per year with multiple database records for each ticket. Can the product handle these volumes?
- Model large and complex workflows: large organizations usually have a well-defined processes and governance structures. Can the tool model complex organizations into logical, business-centric services? Can the tool model workflows, handle diverse change management approval processes, and be used to implement an automated service request management process? Can the tool support the operations of a global enterprise with critical business systems?
- Integrate with common systems and asset management tools: Managing the IT infrastructure is more than just handling calls from end users. Do the tools integrate with the infrastructure management tools and vendors currently in place?
- Support the installation and support needs of large, global enterprises: Installing and customizing a full service management suite in a large organization can take six to twelve months. Does the vendor have the experience, size, and partnerships required to make these costumers successful in the appropriate geographies?

Figure 2.3 – Large Enterprise Service Desk Management [18]
The evaluation uncovered a market served by:

- Industrial-strength enterprise solutions (Axios, BMC Remedy, CA, HP, IBM): All have comprehensive suites of tools covering a full range of service management requirements tied to robust workflow engines and overlaying top-tier CMDB architectures. Additionally, out-of-the box configuration options can also make them appropriate for much smaller organizations that require robust tools to manage their needs.

- Robust mid tier options (FrontRange, iET, Numara Software FootPrints, Service-now.com, Symantec, Touchpaper, USU AG): are competitive options for organizations with between 2,500 and 8,000 employees or those with less demanding workflow or scalability requirements. These products can be implemented more easily than their larger competitors and, depending upon requirements, may be the best choice for some of the largest organizations. All of these vendors have reference customers among the largest corporations.

- Smaller solutions for special needs (BMC Service Desk Express, FrontRante HEAT); are appropriate only for the lower end of the large enterprise segment.

**Smaller Organizations**

These organizations have many of the needs as their large counterparts, however they are more likely to look at tool suites that can be installed with fewer expert resources and lower customization requirements, provide a best practices framework to catalyze the maturing of the service desk organization, and integrate or provide the tools that manage a more PC-centric infrastructure.

![Small Enterprise Service Desk Management Tools](image-url)

*Figure 2.4 - Small Enterprise Service Desk Management Tools [18].*
As result, service desk management tools appropriate for smaller organizations are characterized by:

- Lower platforms demands: In a smaller enterprise or one with fewer global requirements, it is likely that a single instance of the service desk management application will be capable of supporting the organization’s ITSM needs. Does the product support the server and database options most likely found within a smaller enterprise?
- Lower requirements for training and expert resources
- Desktop life-cycle capabilities: Does the suite include or tightly link to PC discovery, inventory, asset management, remote control, and software installation tools?

The evaluation uncovered a market in which there is very little to distinguish basic product functionality:

- Large enterprise vendors ate the top: By their sheer breadth of available functionality and capabilities.
- A large, highly capable group follows: the bulk of the evaluated products are all likely candidates to meet the service management needs of the average enterprise. It is hard to go wrong with any of the products for a robust service desk management tool that can be expanded to provide basic service management functionality across the other key functional areas.
3. Proposal

After an extensive and complete bibliographic research about general IT change management and specific ITIL change management, we now describe the proposed solution for the thesis problem. This chapter includes a clarification of the context in which the thesis proposal is built upon, the initial assumptions and beginning proposal. Finally the Diagnosing and Action Planning phase of Action Research are described, because, as stated by the Action Research definition, it is exactly during these two phases that the researcher works in conjunction with the practitioner to diagnose the problem the organization has, and devise a plan, consisting of the proposal to solve the identified problem.

Briefly put the initial proposal of this thesis is to describe how to really implement ITIL Change Management in medium sized organizations.

By being present on the target organization, and observing the day to day operations of the IT department, we built a real working prototype tailored to not only the target organization but also any medium sized organizations. Therefore part of the challenge was also to build the prototype so that it could be adapted to other organizations and not only this one.

At the beginning of this project there was already underway, and near finishing, the implementation of the Configuration Management Process and of the Incident Management Process. The existing applications are called Change Management Database Application (CMDBA) and Incident Management Application, more on these applications in Chapter 4 Action Taking.

The development of the Change Management Application (CMA) was undertaken with an agile development methodology. Small amounts of work were done, and then feedback obtained by the practitioners. The ITIL proposal was followed as a guideline, and a recurrent debate theme, was that the majority of times, as the practitioners stated, ITIL simply wouldn't fit the needs and specificities of the way they worked there. Several times after some deliberation, the normal conclusion was that indeed the “ITIL suggestions” made sense, and should be adopted, even if it implied to change the usual way of doing things. The result in the end is a working process and prototype, that makes real and practical all the ITIL guidelines and best practices, based on direct observation on the field, and the suggestions, input, and discussions from demonstrations and tryouts with the practitioners.
3.1 Organizational context

The target organization was a Portuguese public organization. The IT department of this organization is functionally structured as shown in Figure 3.1.

![IT Department Organizational Chart](image)

**Figure 3.1 – IT Department Organizational Chart**

This IT Department is responsible for providing IT Services to the entire organization, which represents a total of 650 users, geographically dispersed around the nation.

This is the organizational context where the experiment will be undertaken. It takes places in a wider research project of ITIL implementation, where a Service Desk, Configuration Management and Incident Management Action Research experiments are underway.

3.2 Iterative implementation with Agile Development

One action research cycle was completed in the context of this thesis. However, several agile development cycles were executed between the Action Taking phase and the Action Planning phase can be seen illustrated in Figure 3.2.

![The Action Research Lifecycle](image)

**Figure 3.2 – The Action Research Lifecycle**
The initial proposal was incrementally modified and growing as a consequence of the constant interaction with practitioners. This interaction was made by being on the field specifying the process, domain model and the prototype while observing the problems and needs of the organization. Furthermore several more formal meetings took place for an appraisal of the process and prototype development. As stated before the development methodology used was agile, not only for the development of the prototype but, we can affirm, the research itself.

3.3 Diagnosing phase

First and foremost we had to had a complete and wide understanding of the available scientific knowledge concerning IT Change Management. Being so, the beginning of this phase is described in Chapter 2 Related Work. Then we applied the principles and main conclusions of relevant research to analyze and understand the necessity to change on the target organization, as it is stated by the Action Research principles that should be done in the Diagnosing Phase.

The diagnose was mainly based on the research documents, data and experiences from earlier researchers of the project. We identified a strong need for a better change management process. The need was such that IT Change Management had already in some way been adapted and wrongly attempted to be done through the Incident Management process.

Also, for the organization, the successful implementation of the Incident Management makes Change Management the next logical step. Change Management is currently done in an informal and unstructured process. The advantages of the control, transparency and service quality improvement are a motivation to the continuation of the implementation of ITIL best practices in the organization.

Several important aspects that, in reality, belong to the diagnosing phase are better described in the next section, so actually both phases were undertaken simultaneously (the end of the diagnosing phase, overlaps the beginning of the planning phase).

3.4 Action planning phase

The main scope and users of the Change Management System were the System Administrators according to what the IT Director in conjunction with the research team decided. More formal meetings took place with the 3 practitioners, the System Administrators. But the infield observation took place with the whole service desk staff.

When the diagnose in the organization began, we found that in the target organization there was an unstructured but existing Change Management process. In the planning phase we tried to discover, specify and layout in what exactly consisted the already present, unplanned and unstructured change management process, not only to better understand what was the starting point, but also to better
explain to the practitioners what exactly was Change Management, how we could help them improve it, and why it was important do to so. We briefly describe such process next.

The service desk, at the time, was constituted by a team of 8 technicians. They constituted the first line of the service desk. The second line, were the team of sysadmins (system administrators), constituted by 3 employees. This hierarchy has been previously depicted in Figure 3.1 of Section 3.1.

Figure 3.3 depicts the part of the layout of the IT department floor, were the workstations of the Service Desk were placed. Figure 3.1 of Section 0 also contains the legend of Figure 3.3, matching the symbols of the System Administrators and of the Technicians. As can be seen in Figure 3.1 there were two open space offices adjacent to each other: one for the system administrators (service desk second line) and another for the technicians (service desk first line). It was the responsibility of the system administrators to implement the majority of the changes of IT, for instance server configurations. The change management process consisted of the requests for change coming from either the IT Director, from the Technicians (representing users trough the service desk), or from the system administrators themselves (either by some need or problem identified, or because a user request).

![Figure 3.3 – Encountered Change Management Process](image)

The decisions to and when implement changes where done by the System Administrators, in an informal way, and if, in any kind of change they had some kind of doubt, they went to the IT Director office and orally ask for confirmation.

The change implementation announcements was also orally executed, as depicted in Figure 3.3. The announcements illustrated with a balloon in the figure, consisted of several examples like the following:

- “Unfortunately the backups didn’t end during last night, so will still be running them this morning. Tell us if someone complaints and we will cancel them”.

26
• “If anyone calls during this morning about the email service tell them that it’ll only be available in the afternoon.”

Giving these facts, and the reality encountered, we can assert that one of the greatest challenges of this thesis, concerning it’s planning, was to try to find a balance between the following:

• Their understanding of the previous change management process (different for each sysadmin);
• The actual to be change management process (or rather the researchers understanding of the process);
• The ITIL recommendations;
• The scarce related work recommendations;
• System administrators focus on the interface of the prototype (where each button should go);
• The existing Configuration Management Database and Configuration Management Process, that were not ready to incorporate the Change Management Process.

When referring to discussion or debate between the researchers and the practitioners from now on, it can be read to be as a discussion of all the previously listed topics, and the respective search for balance between them.

Fundamentally the Action Planning was concurrent with the Action Taking as already stated in Section 3.2. The Action Planning was also undertaken through several formal meetings that took place between the practitioners and the researchers were a discussion of the planning took place. Also there was a constant and attentive field observation of the day to day work of the organization. At the same time, we executed the Action, entering the Action Taking phase: the prototype application was being implemented.

To better explain the cycle between planning and action taking, there is an important feature that we must mention. It’s a feature that enables the application to be customized in run-time. The administrator of the application (researchers) can change, for example the transitions, the state names and the mandatory fields to proceed to another state. Figure 3.4 depicts the screen shot of the workflow configuration screen.

![Edit Transition from 'Review' to 'Evaluate'](image)

**Figure 3.4 –CMA Workflow Configuration**

This feature enabled to conduct semi-informal meetings with the practitioners, where they would tryout the application in their workstations. They would input a Request for Change, then proceed to update it and navigate through the workflow as if they were implementing it, other times they would
insert already implemented changes. Thereupon, as suggestions were made, and discussed with the researchers, they were implemented in run-time through the workflow configuration interface, to reflect and try out the recent conclusions of the discussion. Evidently, several times, the discussion went on to be more about the interface rather than the actual process, but the researchers brought the discussion, and therefore the research and purpose of the meetings, as near as possible to the process of managing change whenever possible.
4. Action Taking

In this section we describe the two main Applications involved in the development of this thesis: the Configuration Management Application (called in the organization Configuration Management Database Application: CMDBA) and the Change Management Application (CMA). The CMDBA already existed and was modified to better integrate with and accommodate the needs to implement the CMA. The Change Management Application was developed and built from scratch. Concurrently with the development of the applications, we determined the process of managing change in medium sized organizations, and several other contributions that are further explained in section 5.2 Specifying Learning phase.

Both the applications are used by the Service Desk technicians. The Change Management Application was developed with the objective to be customizable to any organization it might be deployed.

4.1 Technology

4.1.1 Outsystems Plataform

The technology used for the prototype implementation was Outsystems Platform. The main objective of this technology is rapid and agile development of web based applications. The main reason for using this platform was because of the necessity of the integration with the CMDB. Also because the Incident Management Process already was implemented and integrated with Outsystems.

![Diagram](image)

Figure 4.1 – The ITIL Project Component Diagram

Still the referred reason was not the principle motivation to use the Outsystems Platform, it was the substantial advantages presented by this Platform. The main advantages and reasons for using this
technology are: very rapid development and prototyping; lean learning curve and because the result is a web based application, allowing users to access it anywhere they are.

Service Desk Technicians for example, when needed on the user’s location, could access the Incident Management Application on the user Workstation to consult or update information. The Outsystems Platform is programmed through visual development, allowing for workflow programming. It has implemented several design patterns, that save developers time. This enables the programmer to concentrate on the business, on the real requirements of the project at hand, not worrying about the technology underneath.

4.1.2 Development Methodology

We believe it’s important to stress the fact that the software tool is not the objective. There are authors that refer that it merely helps to have friendly software, and also that very successful ITIL workflow were implemented with a simple Lotus Notes forms application [2, 12, 21].

Being so, the prototype was developed with other colleagues on the scope of the ITIL Project being conducted in the target organization. As stated before at the beginning of this project there were already underway, and near finishing, the implementation of the Configuration Management Process and of the Incident Management Process. All these applications were also developed with agile programming methodology in the Outsystems Platform, the consequence is that the time spent actually developing is minimal, the development sprints are very short and there is constant feedback from the users.
The Change Management Application (CMA) was a new component added to a wider research project (depicted in Figure 4.1) that involves various other researchers.

The advantages of the agile development are:

- Frees developers do devise the most ingenious solutions throughout the project, as little time is wasted in repetitive or wrong tasks.
- Small, collaborative teams of developers are able to share tacit knowledge about development processes.
- Flexible deliverable and schedule: it allows the presentation of functional in a short amount of time.

![Figure 4.3 - Traditional Waterfall model vs agile development](image)

### 4.2 Configuration Management Application

The Configuration Management Prototype description is only done to the extent of understanding the Change Management Application, and of describing the contributions made to it in the context of the thesis: some changes were made not only to better the Configuration Management Application but also to facilitate the integration with the Change Management Application.

#### 4.2.1 Domain Model

Figure 4.4 contains a part of the previous domain model of the Configuration Management Application (CMDBA). Each configuration Item was an independent entity and each type of relation also. One of the main challenges of the Change Management Application (CMA) was to easily make temporary copies of the proposed alterations in the CMDB, so that they could be later revised, committed or reverted given whatever happens throughout the Change Implementation, more on this feature and its implementation on section 5.2.3. It would be very complex to implement this feature
with the domain model depicted in Figure 4.4, so we adapted the previous CMDBA domain model depicted in Figure 4.4 to the domain model depicted in Figure 4.5 and Figure 4.6. Figure 1.1 from chapter 1 of this thesis is actually a more complete picture of the previous CMA domain model depicted partially in Figure 4.4.

**Figure 4.4 – Previous Configuration Management Domain Model (partial)**

The domain model presented in Figure 4.5 constitutes the more relevant Entities to understand the adaptations that were made in order to integrate Configuration Management Application with the Change Management Application.

**Figure 4.5 – Configuration Management CI Entities**
There is a super-class like entity, the CI (Configuration Item) entity that contains the common information of all the types of Configuration Items. Every CI created in the CMDB has both the CI and a sub-entity corresponding to the proper type of CI. The CIs used in the Public Organization are:

- Application
- Database
- Communication Device (PDA)
- Rack
- Software (Software, Updates)
- Staff
- Workstation (Laptop, Desktop)
- Server (Virtual Server, Server)
- Router
- Hardware (Tower, Monitor, Keyboard)
- External Device (Printer)

For each type of CI there is also at least one auxiliary entity, the Meta CI, which contains the name the subtype of the CI (for example, the Workstation Type CI, has two subtypes: the Laptop and the Desktop, and the Application CI only has the Application subtype). It is the MetaCI entity that specifies the relation types that exist in the CMDB. The relation types must be defined between the MetaCIs (for instance: the Laptop has a relation to the Software called installation, and the Desktop also has another relation to the Software with the same name). All relations are bidirectional, and have a name in the new version. The main advantage of the new implementation is that there is a data abstraction layer that facilitates the copying of CIs and their relations. In the previous version there were unique identifiers for each type of CI, and now there is just one generic CI Identifier type that points to the CI entity table, unique among all types of CIs. This facilitates, for example, the storing of a list of CIs that belong to a Change Record.

![Diagram of Configuration Management Application Main Entities](image)

**Figure 4.6 – Configuration Management Application Main Entities**
4.2.2 Functionalities

The CMDBA functioned has a Configuration Management Application since the beginning of the project in the Public Organization. It still maintains this functionality because there may be future client organizations interested only in the CMDBA and the Incident Management, and not the CMA.

It functions as a relatively simply Database, where it is possible to create and edit the CI properties, and create and edit the CI instances and relationships.

4.2.3 Interface

The interface of both applications are based on the style guide template that Outsystems makes available and advises developers to follow. This enables even more rapid development because all of the main design options and problems are already taken care of by Outsystems. The style guide template is based on usability best practices, and heavily tested design patterns that enable the ease of use.

![Hardware CI Tower-1](image)

**Figure 4.7 - View of the Hardware CI Tower-1**

Figure 4.7 shows how a CI can be consulted. Notice how the Edit Hardware and Edit Relations links in the top menu are available. The whole CMDBA interface isn’t accessible when the client also has installed the CMA, in which case, only with Administration privileges can the CMDBA then be accessed. Figure 4.8 shows the edit menu for the same CI.
Figure 4.8 – Editing Hardware CI Tower-1

Figure 4.9 depicts a view of an example CI that has a relation. The previous example, depicted in Figure 4.8, didn’t have any relations, so the corresponding tab didn’t appear.

Figure 4.9 – View of Server CI Server 001

It’s important to note that all these interface elements, were revamped because of the new domain model and were reutilized in the CMA interface. They consist of an Outsystems concept denominated Web Blocks, that enable reusing on any number of other web pages. To make proposals of change in a CI in the CMA, either to the attributes or the relations, the same interface Web Blocks seen in Figure 4.8 (edit attributes) and Figure 4.10 (edit relations) are used in the CMA interface.

Finally Figure 4.10 shows the interface to edit relations, where the user can link or unlink the CIs.
4.2.4 Integration

The integration between the Configuration Management Application (CMDBA) and the Change Management Application (CMA) is very similar to the one between the CMDBA and the Incident Management Application (the latter falls out of the scope of the thesis).

Outsystems enables integration between the projects based on the public declaration and exposure of Web Blocks, and Functions.

Figure 4.11 - Add/Remove references screen

Figure 4.11 shows the Add/Remove references screen of the CMA project. We call to attention the ListCIs web block (that enables listing of any type of CIs), the CIEdit, CIShow and ShowRelation web blocks (as is implied by the name that are for viewing and editing the CIs), and finally the public functions LURelation (to link and unlink any given relation between to CIs) and the UCCI (to Update or Create a CI).
4.3 Change Management Application

4.3.1 Domain Model

The main entities of the Change Management Application (CMA) are depicted in Figure 4.12.

- ChangeRecord – represents the change. Every Change Record is created being a Request For Change, and then after the end of the change cycle it's kept has a permanent record of the change.
- CCcategory, CPrivilege, CPriority, CImpact and CUrgency – represents the various entities that determine the respective change categorization, privilege, priority, impact and urgency.
- CI – Configuration Item, this entity is shared with the CMDBA.
- CRCI – Change Record Configuration Items, this entity stores the CIs that are assigned to the change, and the respective copies, where the change proposals are stored.
- CState – represents and stores each state of the change management workflow.
- CStateCR – stores all complementary information related to the state, namely which attributes of the change record should be enabled or not. For example the fields that assign the Impact and Urgency to the change are not enabled after the Authorization state, this information is stored here.
- CStateTransition – represents and stores each transition possible of the change management workflow.

Figure 4.12 – Change Management Application Domain Model
• CStateTransitionCR – stores all the complementary information related to the Transition, for example to make a transition to the Authorized state it’s mandatory that the Risk and Impact Analysis attribute of the Change Record is filed, this information is stored here.

It’s the complementary entities, CStateCR and CStateTransitionCR that enable true customization of the Change Management Workflow which is an important feature of the prototype. Everything is customizable in run time, except for the attributes of the change, as we described previously in the end of Section 3.4.

4.3.2 Functionalities

The Change Management Application (CMA) is a very minimalistic application. The top-level main functionalities are in fact only the creation and edition of a Change Record. There are also all the configuration functionalities related to the customization of the workflow.

• Main functionalities
  o List Change Records
  o Create Change Record
  o Edit Change Record
    ▪ Change attributes of the Change Record
    ▪ Add, remove CIs
    ▪ Edit CIs attributes (only edits a copy, the changes only are permanent when the change is committed, more on 5.2.3)
    ▪ Consult the Change log

• Configuration functionalities
  o Create and edit states
    ▪ Customize the change record attributes that are enabled on each state
    ▪ The tooltip help text that appears next to each attribute of the CR
    ▪ The question that appears on the interface, when in the current state, for the user to select the next state
    ▪ The answer that appears on the interface when in a state that has a transition to this one, for the user to select the next state
  o Create and edit transitions
    • To what state should the transition go
    • If the transition should commit or discard the change stored change proposals
    • What attributes of the change record are mandatory to be filled to use this transition
  o Create and edit categories, privileges, priorities, impacts and urgencies
4.3.3 Interface

The same principles described on section 4.2.3 regarding the Configuration Management Database Application (CMDBA) interface apply to the Change Management Application (CMA) interface.

![Change Management Interface](image)

**Figure 4.13 – Change Management Application (CMA) main interface**

The interface design was as simple as it could possibly be, as there would be introductory training to the users of this application. Also, it was planned to be developed a Single Point of Contact (SPOC) as depicted in the Component Diagram of Figure 4.1, which would function as a dashboard. Because of this, the entry point and main screen of the application as illustrated in Figure 4.13, was kept as simple as possible. In this screen the user can make a Request For Change, creating a new Change Record, and also consult the following tables:

- “Changes awaiting your input”: these are changes that were assigned to the current user in particular. It can be, for example, a change record that the user started to work on, but didn’t finish.
- “Changes awaiting input from Techs”: these are the changes that were assigned to any Technicians but to no Technician in particular. The user can open them, and assign the change record to himself. When the user does so, the change record moves to the table “Changes awaiting your input”. This contains, of course, the changes assigned to the current user team, whatever the team might be, and only users from that team can assign the change to them. In section 5.2.1 the concept of teams, actually called change privileges, and the states for which they are responsible are further explained.
• “All Change records”: as the name states, lists all the Change Records currently in the CMA. It contains both change records that are assigned to another teams, or to another users. This doesn’t include the closed changes, there is another table “Closed Changes” for that purpose.

After a user selects one Change Record he enters the Edit Change Record screen where it’s possible to have a global view of all the different components of a that Change. The different parts of this view are presented and describe in the remainder of this section.

![Change Record Attributes Default View](image)

**Figure 4.14 - Change Record attributes default view**

The Change Record tab contains all the Change Record attributes and these attributes are grouped by type, and expandable as can be seen in Figure 4.14, so it’s easier for the user to find what he wants or needs to edit. Figure 4.14 depicts the General section expanded, and all the other sections collapsed. A complete description of these attributes is given in section 5.2.1.

Figure 4.15 proceeds to show the Configuration Items tab located right next to the Change Record tab. In this tab it’s possible to enter the Change Wizard, where the user can select the CIs that will be contained in this Change, and apply whatever changes to them, that are not persistent, until the change is implemented, and reviewed.
Figure 4.15 – Configuration Item Selection List

Finally Figure 4.16 shows the workflow tab where the user can select to which state do proceed. In the depicted case, the user can either authorize or not the change, making a transition to the authorized or rejected state, respectively.

Figure 4.16 – Workflow section

Figure 4.17 shows the first change wizard screen where the CIs to be added to the current change are selected and added or removed. The following screens of the wizard consist of comparisons between original attributes and relations of each CI and the changes proposed. More on this important feature, including interface screen shots, on section 5.2.3.

Figure 4.17 - Change Wizard
5. Evaluation

5.1 Evaluating phase

The evaluation was definitely the less worked component of this thesis, as we were not able to do a complete test of the proposed solution with the practitioners at the time of the conclusion of this thesis. Still it was done and mainly consisted of the several already referred meetings with the System Administrators and Technicians, and of their tryout and feedback. Several discoveries were made only after the practitioners used the prototype and therefore the process, and came to new conclusions, about how they conducted their work, and how our proposed process based on ITIL could help them improve their work (concerning Change Management).

Further work is definitely needed to improve on this component. What was initially planned, and we propose that should be done next, is to select a number of KPIs similar to the proposed and applied in the studies referred in end of section 2.2.2, very much like the ones proposed in Table 2.2. In such way it would enable to analytically evaluate to what extent the solution effectively influenced the Change Management process in the target organization.

Being so the actual evaluation isn’t an analytical one, is rather the conjunction of all the discoveries and relevant results described in the following section.

5.2 Specifying Learning phase

The main consequences and discoveries of the action are described in this section. The of the more relevant results, are actually unplanned ones, that we didn’t propose on the beginning but only discovered later working together and thinking with the practitioners.

5.2.1 Change Record and Process Description

Change Record

The Change Record attributes described here is the final version of what were several versions and iterations.

- General
  - Change Requester
  - Request Date
  - Change Description
  - Change Agent
- Priority
- Impact
- Urgency
- Category

- Authorization
  - Risk and Impact Analysis
  - Change Authority
  - Authorization Date

- Implementation
  - Change Builder
  - Implementation Plan
  - Back out plan

- Scheduling
  - Scheduled Implementation Start
  - Scheduled Implementation End
  - Scheduled Duration
  - Actual Implementation Start
  - Actual Implementation End
  - Actual Duration
  - Delay

- Review
  - Review Date
  - Review Results

- Rejection
  - Rejection date
  - Rejection Reason

**Process**

The process described here is the final version of what were several versions and iterations of processes.

Throughout the process, in each state it can be seen between brackets the privilege level that is needed to undertake the execution of the state. Only users with the defined privilege are enabled to enter and modify the change. This was already described in the beginning of 4.3.3, when the CMA main screen interface was shown and explained.

- **Received:** Default start state. When a Request for Change (RFC) is initiated this is the state they start. Any user of the Change Management Module with any privilege can do this.
- **Review RFC**: The Change Manager then must briefly review all RFCs in order to verify the filling consistency and standardization of the Change Record. Such as respecting the policies of the organization. In case of rejection the Change Manager fills the rejection reason. The Change Manager may also accept the status of emergency change or not. The Change Manager is responsible for confirming the Category of the Change, it cannot be changed onward. This makes the Change Manager the responsible person to assure that the Change is later authorized by the correct person in the organization.

Figure 5.1 – Proposed Change Management Workflow
• **Evaluate:** The Technician then proceeds to make the necessary evaluations to determine with more precision what is needed and how much time is needed to implement the Change. The Tech much fill has much information as necessary to make sure the Change is approved in the next step. He may also start filing the implementation plan, and must fill out the Implementation Schedule.

• **Authorization:** According to whatever levels are defined in the organization, different privileges can authorize the change or not. When not authorized the Rejection Reason must be filled.

• **Authorized:** In this state the Change awaits for a Tech to begin the implementation.

• **Build Change:** The Change Implementation Start Date is automatically filled. The Tech, or team of Techs, proceeds to implement and test the change.

• **Post Implementation Review:** Is carried out to confirm that the change has met its objectives, that the initiator is happy with the results, and that there are no unexpected side effects. It’s only after this state when the Changes to the CMDB are made persistent.

• **Closed:** If the review is successful the Change is closed.

• **Rejected:** In this state the Tech has the responsibility to report to the initiator that the Change Request was rejected and the reasons for being so, which are filled out in the Change Record by the person that rejected it in the previous state. Then they decide if there should be an appeal, and with what updates and modifications to the Change Request Description, so that it should not be rejected this time.

• **Standard:** This state allows bypassing the whole process of evaluation and authorization. It’s only used in predefined scenarios that must be approved before hand. Should a Tech try to implement Changes that are not Standard this way, the Change Manager can always come to attention of this, with the Post Implementation Review, and then take proper action.

• **Emergency Change:** Bypasses only the authorization part of the process. There are no templates with this type of change, therefore it can only be selected by the Change Manager himself.

• **Implement Fallback:** When there are unexpected results, or the objectives of the Change aren’t accomplished, the fallback is implemented when it’s possible. If not the Change must be closed, and there must be a new Request For Change, possibly with an emergency status, if the problems are disruptive. Only after this state the Change proposals are discarded.

• **Canceled:** The final state when a Change was requested but not approved in some phase, or when the Fallback was successfully implemented (rendering the implementation has if it had not happened).
5.2.2 Change Decision

The seven Rs of Change Management

The seven Rs of Change Management as stated in ITIL v3 Service Transition Book are[23]:

- Who RAISED the Change?
- What is the REASON for the change?
- What RETURN will the change deliver?
- What RISKS are there is we do or do not carry out the change?
- What RESOURCES will be required to perform this change?
- Who is RESPONSIBLE for this change being performed?
- What RELATIONSHIPS are there between this and other changes?

These are, accordingly to ITIL the answers that should always be answered to authorize a Change. We had them in consideration since the beginning, and tried to discover how we could convince the practitioners of the importance of these questions and how they should answer them. In this subsection, and the following two sections we explain the tools that are intended to enable the responsible person for the authorization to better answer the seven Rs of Change Management, and make the correct decision.

Priorities

![Priorities Table]

**Figure 5.2 - The priorities of the Change**

The public organization already had a very personalized priority system, implemented for the Incident Management Process. Figure 5.2 shows part of the taxonomy used by the organization. These categorizations were also defined in an Action Research context, in conjunction with the researchers responsible for the Incident Management Process, and was at the time, under revision. As the Action Research principles dictate, the definitions shown in Figure 5.2 were determined in conjunction with the practitioners. This was defined to be out of the scope of Change Management for the time being, the Change Management Application simply implemented this taxonomy.
Categories

Once again as we stated before concerning other aspects, ITIL also only makes suggestions about what the actual categories of change there should. As for all the other decisions made, after thorough discussion, the researchers and the practitioners agreed on the model depicted in Table 5.1.

<table>
<thead>
<tr>
<th>Change Category</th>
<th>Change Privilege</th>
<th>Public Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minor</td>
<td>Technician</td>
<td>Service Desk Technician</td>
</tr>
<tr>
<td>Significant</td>
<td>Change Manager</td>
<td>System Administrators</td>
</tr>
<tr>
<td>Major</td>
<td>Senior Manager</td>
<td>IT Director</td>
</tr>
</tbody>
</table>

The category is what determines who will authorize the change or not, in the authorization state. If a change is deemed to be “minor”, it means that it only has to be approved by a technician, and he’s the one that takes responsibility. If it is “significant” that it must be the change manager to approve even he is not the one to implement the change, finally when a change is classified as “major”, it requires the approval of the senior manager. The security of the categorization is assured because all changes must be reviewed by the change manager, as can be seen in Figure 5.1, which has the responsibility to make sure that the change is correctly categorized.

5.2.3 CI attributes change and logging

Configuration Item Lifecycle

The chosen configuration item lifecycle is a simple one and consists of only three states. Previous to the beginning of the work of this thesis there were only two states, actually implemented by a Boolean: live or deleted. The new proposed states are:

- **Change** - for every CI that is added to a Change Record, immediately a copy of such CI is created with this state. It is on this copy that all the proposed changes are temporarily stored. If the change is implemented successfully, when the workflow transitions to a certain preconfigured state, the changes are said to be committed. While a change is underway, if any user should consult the CI, it would see the original, whatever the proposed changes maybe. All CIs start with this state, because they are always created in a context of a Change Record as if they were a copy (without the original). When the change is committed they pass to the state live, effectively entering the CMDB.

- **Live** – the default state for the CI
• **Deleted** – As previously stated, already before the beginning of the Change Management Implementation there already was this state, to implement a simple soft delete. It was maintained.

![Figure 5.3 - Configuration Item lifecycle](image)

**Change Proposal Algorithm**

To make possible what was described in the previous section, it was implemented a complex algorithm that enables changes to be done as if they were being done directly to the CI, but are stored in a temporary copy that only exists in the scope of that Change Request. If the change is approved then all the modifications are committed, if not, they are discarded. This applies also to the relations between CIs. If there was a cardinality constraint of one to one between two types of CIs, and if the user wanted to disconnect from one CI to connect to another, he would have to add the three CIs to the Change Request, unlink the CI’s first, and only then make the new link. All this modifications and proposed modifications aren’t visible in the Incident Management for any of the CIs, until they are all committed, or otherwise discarded.

![Current CI Details](image) ![Proposed CI Changes](image)

**Figure 5.4 - Edit CI attributes screen**

Figure 5.4 shows the edit configuration item screen, where the user can compare both the original configuration item on the left (state: LIVE), with its corresponding copy on the right (state: CHANGE), that only exists in the scope of the current Change Record. It’s not possible to edit the original CI, it’s only present for comparison. In the proposed workflow described in 5.2.1, it’s only possible to edit the CIs attributes until the Authorization state, because once it is authorized there can be no further alterations proposals.

As previously stated in the end of the section 4.2.3, the screen of the Change Management Application depicted in Figure 5.4 was developed reusing the Configuration Management Application screens
depicted in Figure 4.8, Figure 4.9 and Figure 4.10. This is why the modifications to the CMDBA are included in the scope of the thesis; they were implemented also for the integration with the CMA.

Change Log and Summary

Another relevant related learning is a feature that truly facilitated the user to make informed decisions when consulting a Change Record. At any phase of the process he is be able to see a history of all the changes and modifications done throughout the whole change management process. This was a feature that was suggested by the practitioners as a consequence of their use. We must call to attention that this is different from the Configuration Management Log, which was already implemented and logs all changes to any Configuration Item, either done through the CMDBA or the CMA. The change log tab can be seen in Figure 5.5.

![Change Log tab](image)

**Figure 5.5 - Change Log tab**

The change log has two main groups represented in tables the changes made to the Change Record attributes (for example authorization, evaluation, implementation dates, adding or removing CIs, etc.) and the changes made to the attributes of the CIs that were included in this Change Record. Also, all this information is grouped in expandable sections, each containing the whole log of a state of the change process. The user can browse through the states, and consult in each state what was done and by who, to either the Change Record or the respective CIs. Figure 5.5 depicts the “Build Change” state expanded, and it can be seen that in this state the Change Record attribute “Actual Implementation End” was filled in, and that there weren’t any changes done to the Configuration Items.

The practitioners concurrently identified another very important information, fundamental to be made available, so that the workflow and control of change in the organization could be effectively done in a
truly informed way: a compact list of only the final changes done to the CIs. This was placed in the “CI Changes Summary Tab” illustrated in Figure 5.6.

![Changes Summary Tab](image)

**Figure 5.6 – Changes Summary Tab**

Throughout the process, different users can propose alterations to the CIs of the Change Record. The same attribute for the same CI can even off course be altered more than once (for example change the network name from PC-000-001 to PC-001-001, and then later to PC-002-001). These changes, done possibly in different states of the process, are listed in Changes Summary tab. But this tab only shows the last alteration done to the attribute (in the example it would only show PC-002-112 for the network name). So this is actually the summary of the final alterations that are proposed in this Change.
6. Conclusion

IT Change Management is of vital importance for the delivery of quality services to the business. Discovering how to do this, and publish these results is an important step to promote and discover more about IT Service Management Implementation, and trying to unveil the great knowledge that many consultants possess and do not share.

There are industry sectors in which change management is handled efficiently and professionally and without incident, for example in the nuclear and pharmaceutical sector. These industries have matured to a zero tolerance of change failure approach. One can conceive the possibility of the impact of a poorly upgraded cooling system at a nuclear power generation facility [37]. If other industries are able to do this, so can be expected of IT Management.

Should it be possible to remake this work from scratch a total new approach should be done. Instead of trying to complete an acceptable prototype before the effective implementation of the process, a new action should, at the very beginning, start to be utilized in production immediately with a minimalistic prototype and Change Management process, as long as it would work. Then, continuing from there, we could improve and further investigate with the live action prototype, the change management process and prototype itself. What we suggest is illustrated in Figure 6.1. Instead of only deploying the prototype and process when in version 1.0 (arrow A), it should be deployed before (arrow B), as it would truly facilitate the evaluation of the proposal, present results more quickly, and more importantly: it would enable the research to progress with input from the actual use of the prototype.

![Image of traditional waterfall model vs agile development](image)

**Figure 6.1** - Traditional Waterfall model vs agile development [38]

Release Management is hardly referred to in this document. Release Management, simply put, gathers changes as release packages and thus take control over releases. Accordingly to the success of Change Management implementation this might be a future work to be developed.
There are two distinct endings of a spectrum that we believe to be important to explain. When the diagnosing begun, what we found in the target organization was an unstructured but existing change management process, this corresponds to the reality, to the practical situation we encountered. In the other end of the spectrum, there was the famous and important ITIL Change Management. The real contribution of this thesis was to begin to understand and show how to bridge them both, and try to implement the “utopical” standard that ITIL actually is for such organizations, as the public Portuguese organization where this thesis took place.

Another very relevant conclusion is the fact that we believe it wasn’t given enough attention to the definition of the scope of the project. Maybe if a smaller scope was defined, with better smaller objectives, more results would have been achieved. This is different from what was explained with Figure 6.1. It’s actually about understanding what to do and not how to do it. ITIL states what organizations should do, and we assumed we could go for it, to what ITIL states, when we first should be researching and deciding, since the beginning, given what ITIL recommends, what can we do, in the scope of the thesis and in the context of the organization were the work was developed. We affirm this conclusion to be very important because it possibly is what normally goes wrong in the organizations when implementing ITIL: is not the problems normally listed, as we list in section 2.1.5, these are the different causes of the fundamental problem: the fact that small and medium sized organizations don’t define exactly the goals to achieve, and the scope to undertake when they say to be “implementing ITIL”.

### 6.1 Future Work

One very important and inevitable future work to do is to discover and determine how the concept of the CAB (Change Advisory Board) can be useful to this scale of organization. In spite of not getting to actually finish and implement the CAB, it was planned, and briefly discussed with the practitioners. What was planned was to execute face to face CAB meetings, of which main conclusions would be included in a proper field in the Change Record, for future reference.

For the Incident Management Application there already was implemented and functioning a dashboard that allowed the consulting of all the current incidents, and their state, and what each of the technicians was working on. This was displayed in wide flat screen both in the system administrator’s office and the technician’s office. Another important to do work, is the integration with this incident management dashboard.

As stated in the Evaluating Phase section, this is the most important future work to do: get to effectually implement the change management application in the public organization and truly validate in the day to day operations of the service desk, what was only tested, prototyped and discussed before.
References

13. Cater-Steel, A.: ITIL adoption in Australia: 2 years of itSMFA surveys and case studies. itSMF Australia 10th National Conference and Expo, Melbourne (2007)
19. Morales, S.B.: Do you want to leave a trail or make a path? : Proceedings of the 36th annual
ACM SIGUCCS conference on User services conference. ACM, Portland, OR, USA (2008)
20. Wan, S.H.C., Chan, Y.-H.: IT Service Management for Campus Environment-Practical Concerns
in Implementation. Integrated Network Management, 2007. IM ’07. 10th IFIP/IEEE
Services Conference. Council of Australian University Directors of Information Technology, Gold
Coast, Queensland, Australia (2005)
22. OGC: Service Support (IT Infrastructure Library). The Stationary Office, United Kingdom
(2000)
24. OGC: Service Delivery (IT Infrastructure Library). The Stationary Office, United Kingdom
(2000)
Computer Society 48 (2006) 30
International Organization for Standardization, Geneva (2005)
International Organization for Standardization, Geneva (2005)
29. Cater-Steel, A.: Integration of service management with CMMI and SPICE. 5th Annual SEPG
Australia Conference. Griffith University, Gold Coast, Australia (2007)
Australia 2006 Conference. University of Southern Queensland, Toowoomba, Australia (2007)
responses. itSMF Australia 2007 Conference. University of Southern Queensland, Toowoomba,
Australia (2008)
33. Scott, D.: Making Smart Investments to Reduce Unplanned Downtime. Tactical Guidelines
35. Mattila, A.: Best Practices for Network Infrastructure Management - a Case Study of
Information Technology Infrastructure Library (ITIL) (Master Thesis). Helsinki University of
Technology, Helsinki (2008)
38. Outsystems: Transitioning to Agile in an Agile Way: Amplifying Traditional Approaches with
Agile Technology. (2008)