Haaga-Helia
University of Applied Sciences

Improvement of the Requester Console (front-end) and the Reference Configuration (back-end) of a ticketing management system (BMC Remedy)

Minna Mäkelä - Raúl Martínez

Master's Thesis
Degree Programme in
Information Systems Management
2017

Haaga-Helia
University of Applied Sciences

Abstract

31/08/2017

Author(s)

Minna Mäkelä - Raúl Martínez

Degree programme

Information Systems Management, Master's Degree

Thesis title	**Number of pages and appendix pages**
Improvement of the Requester Console (front-end) and the Reference Configuration (back-end) of a ticketing management system (BMC Remedy)	96

The objectives of this project were to deliver and deploy an improved Requester Console and a standardised Reference Configuration to be used across all IT services in our target organisation.

The theoretical framework of this project included the exploration of methodologies that helped us to focus on the end users' needs to create a better interface (UX Research and UX Design), to simplify and remove waste (Lean) and to apply the best IT Service Management practices (ITIL).

The methods used to collect quantitative and qualitative data for our project were interviews and focus groups combined with brainstorming and extraction of data using a Business Objects reporting tool.

Once all the information needed was collected, we analysed it using four different methods: Affinity diagrams, Ishikawa diagrams, MoSCoW method and Pareto distribution.

The conclusions obtained from the analysis combined with the applied theory made possible the creation of several iterations for the Requester Console and the piloting of four IT services for the Reference Configuration. This helped us to collect feedback and fine tune them before the go live. A validation for both elements was done one month after the go live to verify that our initial objectives were achieved.

As a conclusion we were both satisfied with the project delivered since it was on time, with the quality and user satisfaction expected. From the personal point of view, this experience expanded our knowledge in the IT Service management area and improved the efficiency of our daily tasks.

Keywords: User experience, UX research, UX Design, ITIL, categorisation, Lean, Service Request, Incident, IT Service Management, ticketing system, Requester Console, Reference Configuration.

Table of contents

Table of figures

1. Introduction

The initiation of this project arises from the revamp of the IT Service Portfolio in our target organisation, which took place during 2016 and 2017. This revamp meant the big first change in years and it was born with very concrete demands.

IT services available in the IT Service Catalogue were originally created focusing on technical excellence instead of using a customer-centric approach. This decision had a negative impact on the main interface used by front-end users to interact with IT services and their dissatisfaction was corroborated with the feedback received in the internal IT Survey 2016.

On the other hand, the amount of IT services in the IT Service Catalogue became huge due to the model used to build them, “one business application one IT service” with a total of 68 IT Services. In addition, our target organization uses an Integrated Quality Management System (IQMS)[1] meaning that for each IT service it was necessary to create and maintain comprehensive documentation such as work instructions, written procedures and policies requiring lot of human resources to keep it up to date.

Other consequences of having a huge IT Service Catalogue were the complexity for its implementation and configuration. Such complexity was reflected in a confusing user interface with multiple choices in which users ended up making wrong selections and as a consequence data was wrongly classified. This fact had an impact on the quality of the data extracted, creating inaccurate reports, which at the end the management was not interested in reading them since it did not give any added value.

The lack of interest in reading reports supposed a poor monitoring of the IT services, not being able to see the full picture of what was really happening (bottlenecks, trends, etc.) and missing the opportunity to use that information to steer and improve the services.

In the following picture, it is shown the vicious circle created by a huge and complex IT Service Catalogue:

[1] *The IQMS is a repository where all the policies, processes, work instructions, etc. are stored and documented in order to ensure our organisation* works as a single unit with unified objectives.

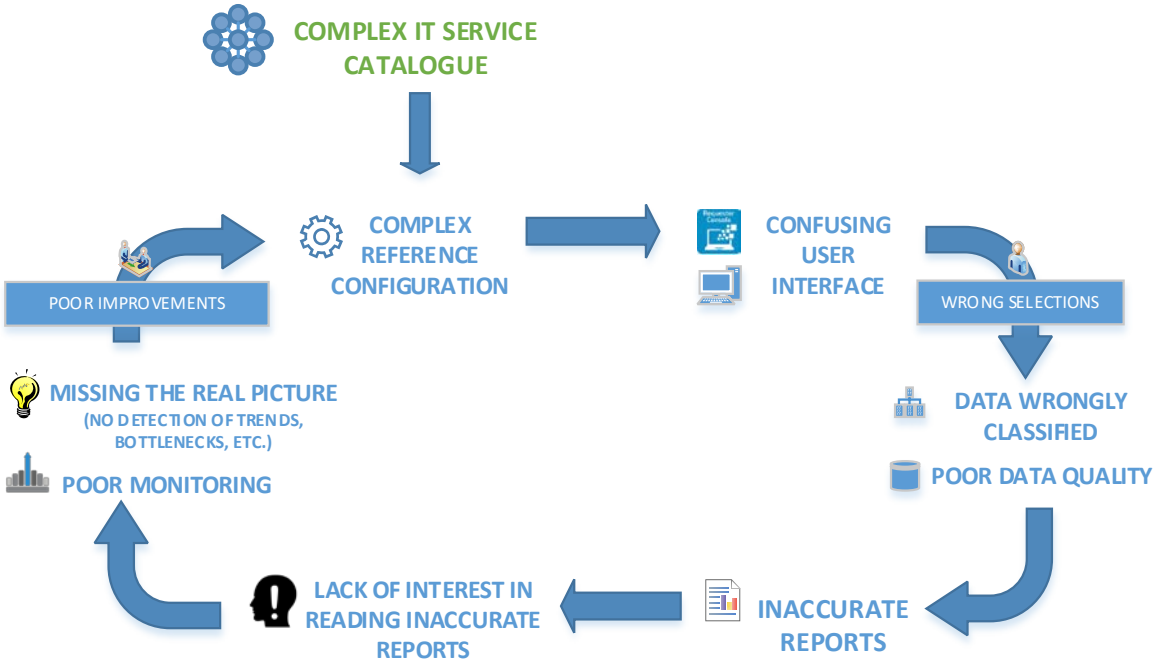


Figure 1. Consequences of a complex IT Service Catalogue

For the previous reasons, we took the initiative to start a research, collect evidences and understand how the current situation could be improved focusing on the following elements:

ELEMENT	DESCRIPTION
Front-end user	Staff that uses IT services on a daily basis and may need to report an incident or create a service request.
Back-end user	IT staff providing support to front-end users. For example: IT helpdesk officers, service managers, etc.
Requester Console	Interface used by front-end users to contact the IT services support (Service Desk).

Ticket Management Module	Interface used by back-end users to handle services requests and incidents submitted by front-end users.
Reference Configuration	Subset of the IT Service Catalogue used by the Ticket Management Module and the Requester Console. The Reference Configuration used within the scope of this project includes the following items: • Services • Products • Support groups • Operational categorisation

The interactions between the different elements presented above are illustrated in the following figure:

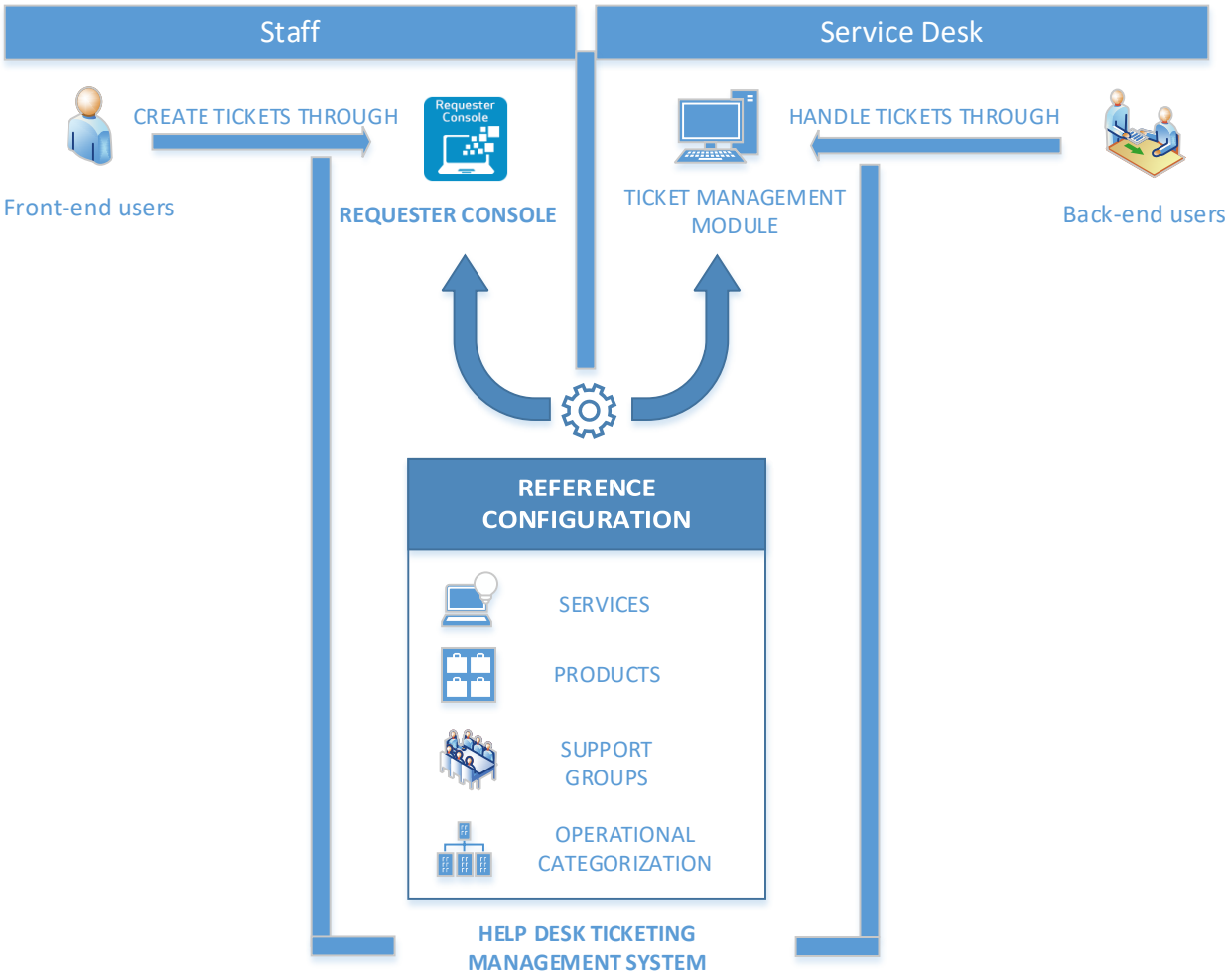


Figure 2. Interaction workflow

The previous described set up provides a service framework that can be seen as a bridge between the front-end and back-end users. The Reference Configuration is the central pillar that provides contents for both user interfaces and unifies two different perspectives of the same reality, which is the set of “Services”, “Products”, “Support Groups” and an “Operational categorization”.

Front-end users can submit service requests or report incidents by selecting any of the products and services available in the Requester Console. Once they have completed and sent their request, a ticket[2] is sent to the Ticket Management Module. They can also track the progress of their own tickets and provide additional information via this interface

Back-end users receive tickets based on established workflows and the initial selections made by front-end users. Once a solution is found or a service request is fulfilled, the front-user is informed and the status of the ticket set to resolved.

1.1. Aim

The aim of this project was to support and guide the IT Service Portfolio revamp ensuring that it moves toward a more customer-centric approach. This meant to provide tangible benefits to the users by enabling them to work more efficiently and maximising their satisfaction with the resources we had available.

1.2. Objectives and Research questions

The objectives of this project were to deliver and deploy an improved Requester Console and a standardised Reference Configuration to be used across all IT services.

For the front-end user interface (Requester Console) a validation was done through a user satisfaction survey and for the Reference Configuration via the comparison of the performance in the old and the new set-up.

Furthermore, it was required to create a short-term roadmap for planning further developments within the scope of the Requester Console and the Reference Configuration.

---

[2] *A ticket within the context of IT Service Management is a record with a unique identifier that usually contains information related to a service request or incident. It contains also other details such as the service and product selected, contact details of front-end and back-end user, impact, priority, etc.*

In order to deliver the aforementioned elements, we had to perform a proper research to investigate the main issues behind them. In order to do so, the following two research questions were defined:

1. ***What kind of front-end user interface should our target organisation have to increase user satisfaction?***
2. ***What kind of reference configuration should our target organisation have to improve data quality?***

1.3. Motivation

We acknowledge the need for improvements since we were affected in our daily tasks by the lack of efficiency in the processes related to the Requester Console and the Reference Configuration.

Minna as Service Manager for the Ticketing Management System (BMC Remedy), had the technical knowledge and a wide understanding about the IT Service Portfolio. Raúl as Product Manager and Reporting Officer understood the main needs from users and the poor data quality in the reports created.

In order to share and balance the workload, the theoretical framework was slightly divided between us. Minna focused on searching a concept of user experience (UX) and how it could be utilised within this project while Raúl focused on the investigation of the current state of art in Artificial Intelligence applied to the IT Service Management area.

Otherwise the rest of the work was divided and shared equally between both authors.

1.4. Scope

The scope of this project was narrowed to the revamp of the Requester Console and the Reference Configuration with limited resources during the period of December 2016 till July 2017.

It is relevant to mention that all improvements in this project were done on the existing BMC Remedy platform and we were obliged to use its licensed development tools. Therefore, we knew already since the beginning that this would bring some technical limitations, decreasing the amount of potential solutions that we could apply in the UX design phase.

Training to front-end users about how to use the Requester Console was not in the scope of this project, however, we organised four general presentations during the first month after the go-live.

Furthermore, the decisions and actions related to the IT Service Portfolio (e.g. simplifying the current structure) were out of the scope of this project.

Referring to the ITIL framework, the scope of our investigation focused in the Service Design phase, Service Transition and Continual Service Improvement explained shorty under “ITIL chapter” of the theoretical framework. The development work was done following the user experience research and design framework explained under chapter “User Experience” of the theoretical framework.

2. Theoretical framework

In order to perform our research, analyses, design and development work, we used three different frameworks for guiding us during the whole project.

Firstly, we focused on ITIL to enhance our understanding about IT service management practices and to define IT concepts that we identified as core entities in our project. Using this framework, we were able to identify the different phases in the lifecycle of IT Service Management. This overview helps us to locate our development case in this cycle and provide guidance to our work. The lifecycle of a service embeds the concept of the continuous service improvement (CSI) from where the initiation of our development case arises. Therefore, we needed to investigate how our project fitted into this concept and what ITIL's practices we were able to apply in this project.

Secondly, we focused on Lean and Lean IT which added new perspectives to the improvement possibilities in terms of reducing waste and tasks that do not add value to customers.

Thirdly, we focused on UX user experience and its methodology which guided us on how to build a good user interface to be followed in our design and development phases.

Lastly, despite it cannot be considered as a framework, we investigated how the field of artificial intelligence is disrupting the IT Service Management area and what benefits it would bring for our target organization to move into that direction beyond this project.

2.1. ITIL

2.1.1. ITIL Introduction

Information Technology Infrastructure Library (ITIL) is one of the frameworks developed for IT Service Management in the 1980s. It originates from the public sector in the UK with the aim to elaborate an approach to the efficient and cost-effective use of IT resources without being dependent on any supplier. ITIL defines how it can be achieved by organising IT operations as services and forming a process model view for managing them.

This ITIL's approach to services is business and customer-oriented since the demands and requirements come from the customer side. IT services are designed to provide value to customers with the aim of satisfying their expectations. If the business changes, it

means the service requirements also change having as a consequence a change in the service.

The content of ITIL can be seen as a collection of best practices observed in the IT service industry without being based on any proven theory. ITIL cannot be considered as a tool, however it gives a detailed description of a number of checklists, tasks, procedures and responsibilities that complements IT practices.

What is an IT Service?

ITIL defines the concept of the "Service" as follows: "*A means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks*" (Cabinet Office, 2011a, p.13).

Whereas it defines the concept of "IT Service" as follows": "*A service provided by an IT service provider. An IT service is made up of a combination of information technology, people and processes. A customer-facing IT service directly support the business processes of one or more customers and its service level targets should be defined in a service level agreement. Other IT services, called supporting services, are not directly used by the business but are required by the service provider to deliver customer-facing services*". (Cabinet Office, 2011a, p13).

Based on the abovementioned, the definition of the IT service ties it directly with information technology and groups IT services into different categories with an agreement with the customer on the outcome. The outcome provides value to customers by combining both utility and warranty. Utility means how the service fits for purpose (e.g. able to meet its required outcomes) and warranty how the service fits for use (e.g. availability, capacity, continuity and security). Since services are not tangible, the value comes from the outcome of the using service. (Cabinet Office, 2011a, pp.17-18).

What is a service desk?

ITIL defines the concept of the "Service Desk" as a function that supports interaction and communication between the users and the service provider. The purpose of this connection is to provide a single point of contact and to facilitate the communication in a day-to-day base. This service desk function manages issues related to disruptions in the service, formal service requests and requests for changes related to the operation side of the service. These actions belong to the incident management, request fulfilment and change management processes. Since these processes are separated, it allows faster restoration of the services back to normal operation. (Cabinet Office, 2011a, pp.22, 2011b, p.158).

Incident

ITIL defines the concept of the “Incident” as follows: “*an unplanned interruption in an IT service or reduction in the quality of an IT service*” which might be noticed and reported by users, raised by technical staff or alerted by monitoring tools. Therefore, incidents can be whatever that might occur in systems. Once the incident is logged, it is managed through the incident management process by the service desk that addresses the incident quickly in order to restore the service to normal operation as soon as possible. (Cabinet Office, 2011b, p.72).

Service Request

ITIL defines the concept of the “Service Request” with this generic description “*a formal request from a user for something to be provided*” which can be for example requests for information and guidance, to reset a password or to install a new device. Service requests are managed through the request fulfilment process and are considered as standard changes already pre-approved with low risk. The rest of requests for change should be managed through the change management process. (Cabinet Office, 2011b, pp.86- 87, p.343).

Request for Change (RFC)

ITIL defines the concept of “Request for Change” as “*a formal proposal for change to be made that includes details of the proposed change*”. Once a service is operational its support staff needs to ensure that it is running smoothly and stable. However, from time to time changes are needed to be implemented in the configuration items of the service (e.g. new components, upgrades, legislative changes, obsolescence of items and enhancements to processes). This kind requests needs to be managed in a controlled manner via the change management process so that changes are made with the minimum disruption to IT services. The way of dealing with each type of change is associated to a series of steps needed to handle it. (Cabinet Office, 2011b, p.227, p.337).

2.1.2. ITIL Phases

According to the latest version of ITIL version 3, there are 5 different stages that cover all the Service Lifecycle as illustrated in the following figure:

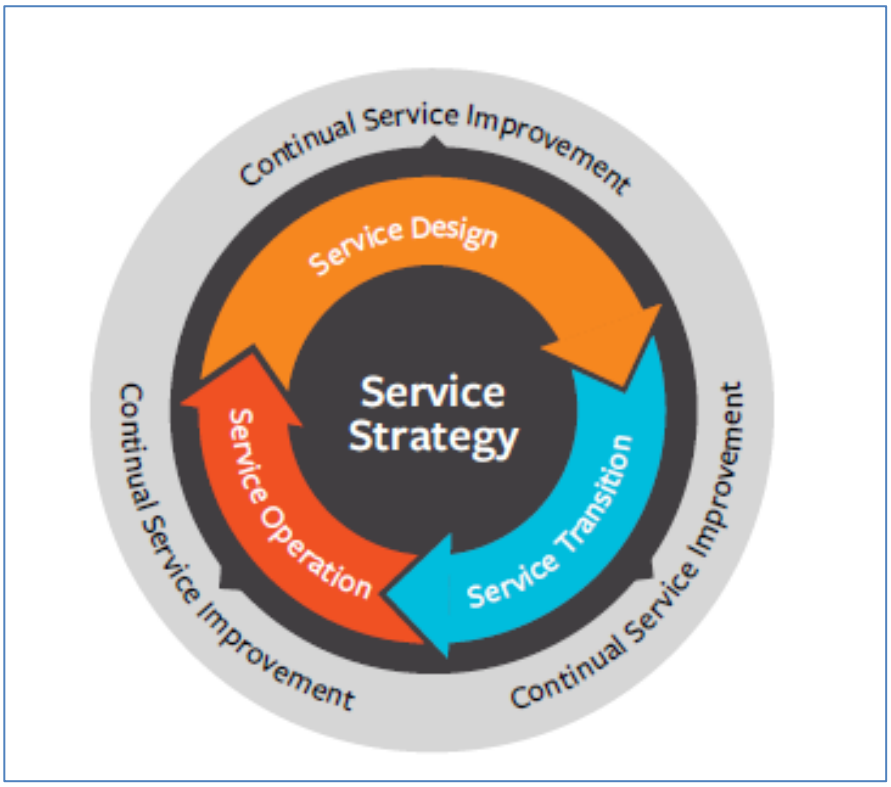


Figure 3. ITIL Service Lifecycle Stages, (BMC Software, 2016, p.3)

The core of each stage is summarised as:

1. *Service Strategy* – aligns IT and business by applying business strategic thinking to IT Service management. The service strategy can be seen as a plan of actions to achieve a certain goal that originates from a customer need and a desired business outcome. It covers the following functions: Service Portfolio Management, Demand Management and IT Financial Management.

2. *Service Design* – translates the strategic plan into a design and detailed specifications. The business requirements are defined more in detail having as a target the introduction of the service into the production environment. It focuses in finding a balance between the functionality of the service, a guaranteed service level, timetables and resources (e.g. available budged, staff and technology).

3. *Service Transition* – is the last stage before the service go-live. In this phase, the service is build, tested against the requirements, failures are fixed and a validation is performed before the service is ready to be released and operational. Previous steps can be seen also as change and risk management processes.

4. *Service Operation* – manages IT services and runs their day-to-day operations according the established service and operational level agreements. It continues the work that was accomplished in the service transition stage and offers services to customers starting to bring value to them.

5. *Continual Service Improvement* (CSI) - aims to improve services continuously through the whole service life cycle. It focuses on finding areas that are relevant to the business and new ways to improve them. An important element of this stage is the continuous measurement of the performance, proving if the improvements applied in the services have increased their efficiency and effectiveness or not.

The next figure shows how five phases fit together providing an added value to customers through service operations:

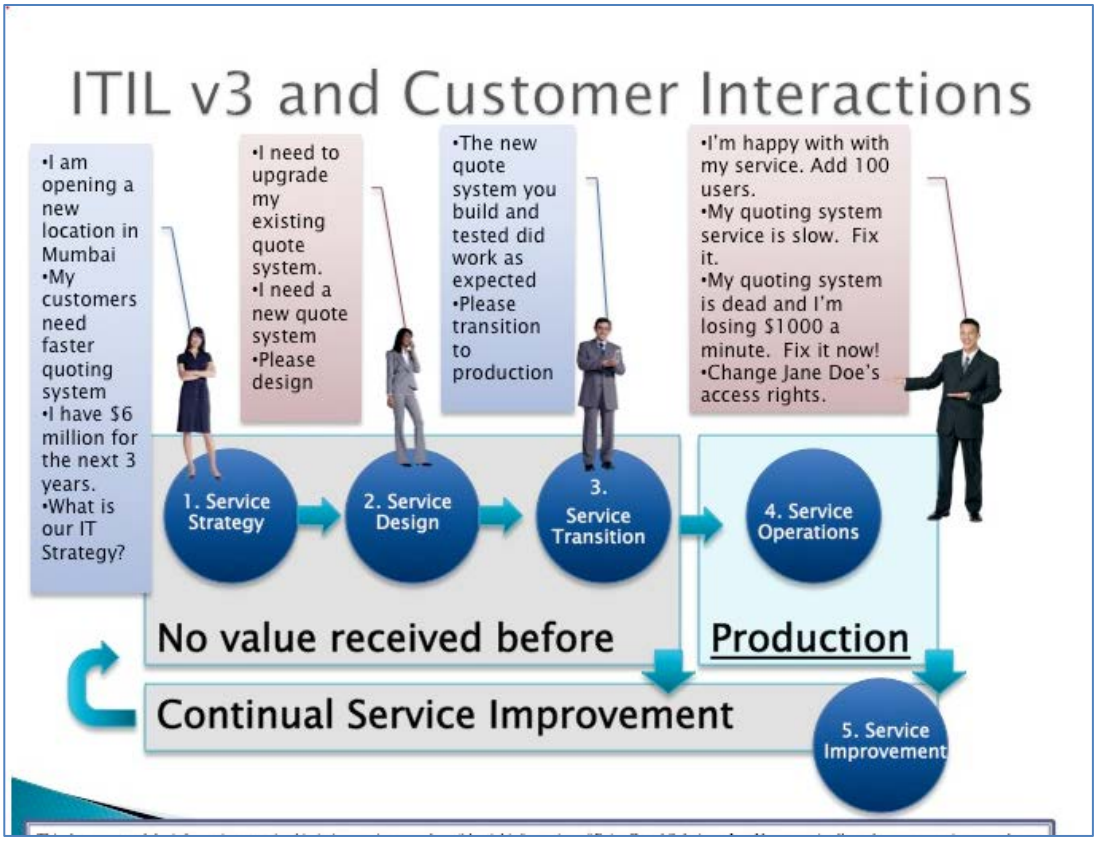


Figure 4. ITIL's phases with customer interactions, (Point Guard, n.d.)

2.1.3. Continual Service Improvement

ITIL CSI is the concept that provides a framework for service measurement. It uses the Deming Cycle Plan-Do-Check-Act (PDCA) as a base layer complemented with a seven-step improvement process that covers the entire lifecycle of a service. This process has

the shape of a continuous spiral in which each step is built based the previous one, meaning that the outputs produced becomes the input for the next one.

The following figure illustrates the PDCA base layer surrounded by the 7 step improvement process.

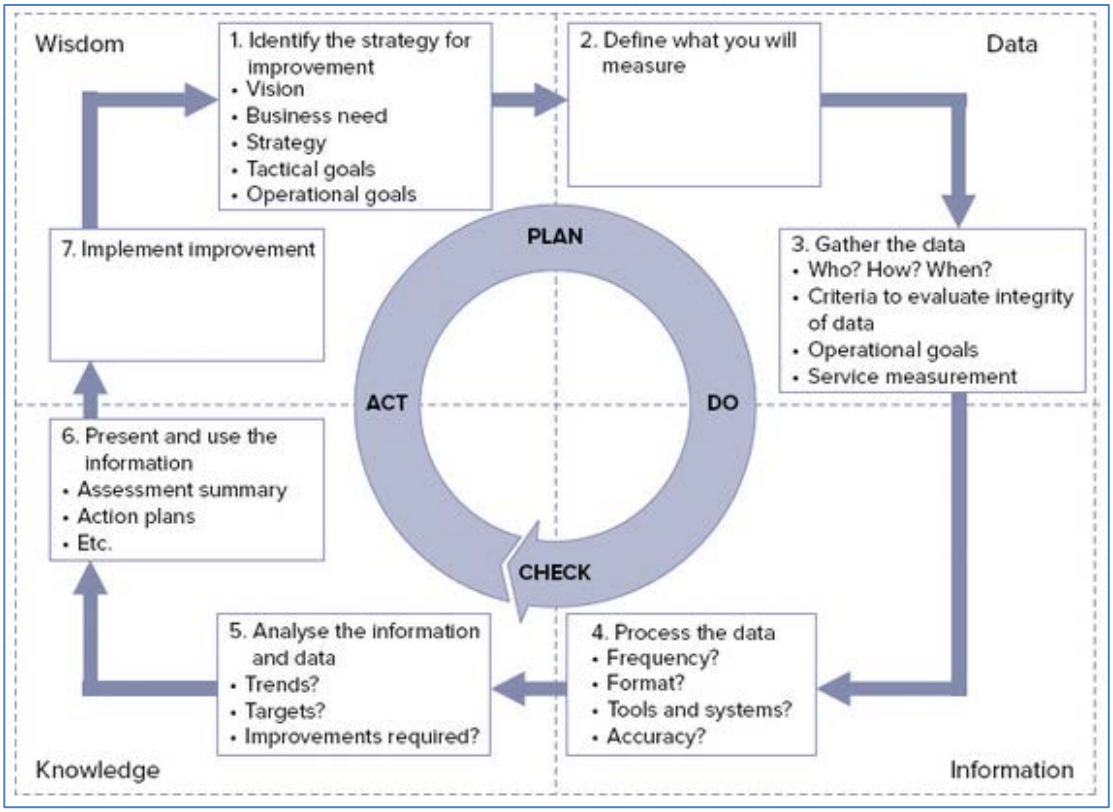


Figure 5. The seven-step improvement process, (Cabinet Office, 2011a, p.40)

These seven steps are summarised as follows:

1. *Identify the strategy for improvement* - The improvements start from the business point of view helping to understand where it wants to go. Therefore, the business vision, business needs, the strategy, the targets and the goals are the ones that identify what it should be measured. (Cabinet Office, 2011a, pp.49-50)

2. *Define what you will measure* - Once the strategy is clearly identified, then the measurement plan needs to be defined in the service strategy and service design phases. It begins with the review of the current state of the business and IT areas that will be compared with the target state. Then the differences in the measurements are analysed, caps identified and based on that, a new plan for measuring a new goal will be created. (Cabinet Office, 2011a, p.39, pp.49-50)

3. *Gather the data* – Once a new goal is defined, data from different sources are collected (service, process and technology metrics) in the service operation phase. This might require the creation of procedures (if not existing yet) for monitoring and collecting data. The raw data gathered will be passed to the next step for processing and analysing. (Cabinet Office, 2011a, pp.53-55)

4. *Process the data* – In this step raw data collected is converted into the required formats that can be understood and compared easily against the baseline data. The data is formed so that it provides an end-to-end perspective about the overall performance of the service and provides better means to analyse the information and data in the next step. (Cabinet Office, 2011a, pp.55-56)

5. *Analyse the information and data* - In this analyse phase, the data is in the format that gives the answers to the questions about "who, what, when, where, and how" showing the current trends and impacts. However, this requires a deeper analyse otherwise the use of the data may be just informal and no improvement opportunities can be noticed. (Cabinet Office, 2011a, p.40, pp.58-60)

6. *Present and use the information* - This phase focuses on presenting the analysed data in tailored means, formats and levels that suit best for different audiences. The idea is to provide an accurate picture of the improvements applied and verify if the goals were achieved. (Cabinet Office, 2011a, p.40, pp.60-62).

7. *Implement improvement* - The final step is about how to use the knowledge gained in the previous steps with the wisdom to take the actions and decisions to improve the service again. In the practice it means to start the whole improvement cycle once again. (Cabinet Office, 2011a, p.40, pp.62-64)

2.2. Categorisation of service requests and incidents

This chapter supports the design of the categorisation used in the Reference Configuration to manage tickets and extract data for reporting purposes.

A categorisation can be defined as: "*a group of people or things arranged by class or category*" (The FreeDictionary, n.d.).

In the context of ITIL, the categorisation is part of the request fulfilment and incident management processes and used to separate them. This separation ensures that the correct process is fast initiated helping to shorten the resolution time of a ticket. The original categorisation given at the creation of a ticket may be changed during its lifecycle. (Cabinet Office, 2011b, pp.76-78)

Depending on the tools and the level of the automation, the record of a service request or an incident can be done by front-end users via a service portal (in our project the Requester Console) or by the service desk staff on behalf of the users (in our project using the Ticket Management Module). ITIL assigns the task of categorisation to the first level support of a Service Desk who should classify recorded tickets (service requests and incidents) according to a category that is formed based on some common relations or attributes. (ITIL Service Management, 2007).

The next figure shows the flow in the incident management process since the moment an incident is identified until the incident is categorised (the steps for the service requests process are pretty much the same until the categorisation is done):

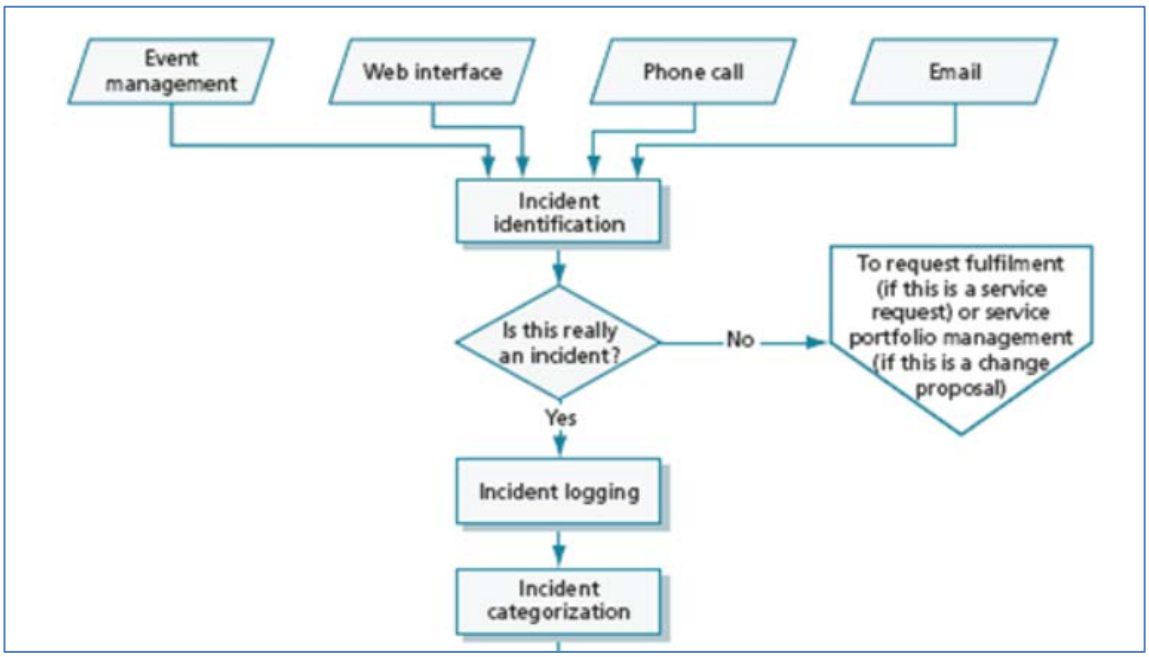


Figure 6. The Incident management process flow until categorisation (Cabinet Office, 2011b, p.77)

In the first place, the reason why a categorisation exists is to facilitate the initial support done by the service desk. This means proper analysis, evaluation, escalation and resolution to restore a service or fulfil a request. (Marquis, 2010).

In addition, there are other actions related to the incident and service request workflow that are facilitated by the use of the categorisation such as prioritization, assignation and escalation of tickets. The categorisation can be used also for problem management to analyse the root cause of incidents and for reporting purposes. (ITIL Service Management, 2007).

For the previous reasons a categorisation can considered as an important task however ITIL does not prescribe any specific format on how to implement a categorisation. Instead, it leaves freedom for organisations to choose and create the best categorisation that fits they own needs. This sounds reasonable but on the other hand, it can be seen as "l*ack of standardised term and terminology differences that causes problems for classifying tickets and communicating between different service provider organizations"*. (Jäntti, Cater-Steel, Shrestha, 2012, p.204).

Therefore, we also took a look to other sources than ITIL to get practical guidance on this matter. IT Service Management tools makers usually provide guidance for classification as a simple three-tiered relationship model which is called the Category/Type/Item (CTI) model. In this approach the "Type" is associated to a "Category" and the "Item" to a "Type" following a tree structure (Marquis, 2010).

This CTI approach suggests that words describing a "Category" or "Type" should be nouns and words describing an "Item" verbs as it is described in the following example:

- **Category (noun):** Database
    - **Type (noun):** Access
        - **Item (verb):** Upgrade. (Marquis, 2010).

If we think about an enquiry received that would need to be classified following the previous example, the first level of support (Service Desk) should identify first the kind of process it is, incident or service request. After that, they would need to know that "Access" is found under "Databases" which might be not so obvious and finally, a proper investigation with deep technical competence to complete the last level of the CTI classification. Unfortunately, this approach ruins the ITIL's purpose of simple initial support and it requires too much technical knowledge and details (Marquis, 2010).

Therefore, we turned back to ITIL and its concept of initial support determined by the type of work (process) that needs to be done. It can be an incident, a service request or a request for changes (RFCs) that belongs to a process called change management process.

This piece of information is the one that should go to the first level of a categorisation as a "Type". (Marquis, 2010).

The second level of categorization should be the "Category" that defines what action needs to be done based on the "Type".

The third level of categorization should be "Sub-category" that defines what specialist provides support. The next example shows how this approach would look in practice:

- **Type (process):** Service Request
  - **Category (action):** Help User
    - **Sub-category (special):** Desktop Application. (Marquis, 2010).

This ITIL based taxonomy clearly communicates and defines which process is required "Service Request", specifies the action to be performed "help user" and identifies what specialist "Desktop Application" is needed to fulfill the request. This approach does not mix different objectives but keeps the investigation and diagnose separated from initial support focusing only on classifying the issue reported. (Marquis, 2010).

2.3. Lean

The concept of Lean was originated with the Toyota Production System (TPS) developed after World War II to improve the quality and efficiency of their car factories (Bell, Orzen, 2011, p.14).

Its main principle was to reduce waste and tasks that do not add value to customers. To achieve this, it uses the so-called "pull production" to produce only what is needed. (Lee, Olson, Lee, Hwang, Shin, 2007, p.1).

The main principles of Lean are:

- Listen to the customer to understand what the customer values and focus on that.
- Identity the value stream and eliminate all tasks which do not give an added value.
- Build the value stream based on the customer needs (pull).
- Respect and positively acknowledge people in all levels of the organization and engage them.

- Continuous improvement every day.

Womack & Jones (Shrivastava, 2012) studied Lean in the context of consumption and introduced the concept of the Lean Consumption. It forms the idea of providing the full value of the service or good with the greatest efficiency and minimum disruption. According to them, there are six simple principles of Lean Consumption that can be linked to the previously listed principles of Lean which are:

1. Solve the customer's problem completely by ensuring that all the goods and services work, and work together.
    - This means that instead of fixing all time the same problem all over again, the aim is to identify the root cause of the problem and resolve it.
2. Don't waste the customer's time.
    - The aim is to minimize the time that the customer is forced to spend without any returned value.
3. Provide exactly what the customer wants.
    - It means to achieve the main Lean principle of "pull" delivering based on the customer order.
4. Provide what's wanted exactly where it's wanted.
    - Several delivery models are needed for the service since the circumstances of the customer changes.
5. Provide what is wanted, where it's wanted and exactly when it's wanted.
    - The aim is to give the customer a change to plan and customize her/his need together with the provider so that she/he gets benefits from early ordering.
6. Continually aggregate solutions to reduce the customer's time and hassle.
    - The aim is to collaborate with the customer in order reach common understanding, plan together and provide all needed services as a service package.

2.4. User Experience (UX)

Though ITIL gives loads of checklists, tasks and procedures, etc., it does not provide detailed practices on how to create an interface between front-end users and back-end users. It only mentions about the concept of having a Single Point of Contact (SPOC) but leaving the freedom to establish it in the way that it fits better for the service.

User Experience (UX) covers the most important areas that we wanted to improve in the Requester Console, therefore we decided to include it in our theoretical framework.

2.4.1. What is UX?

We start by defining what UX is according to the following definitions:

- "*UX is the experience that people have when they interact with the product, system or service*". (Newman, 2017).

- *"UX encompasses all aspects of the end-user's interaction with the company, its services and its products" (*Norman, Nielsen, n.d.).

- *"UX is a consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed system (e.g. complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g. organizational/social setting, meaningfulness of the activity, voluntariness of use, etc.)".* (Hassenzahl and Tractinsky 2006, p. 95).

Each of those definitions highlights that it is all about the experience that the users perceive when they interact with a company including all possible aspects that happens before, between and after the use of the services or products. Furthermore, even the user's internal state has influence on this experience.

The attributes that belong to a good UX are: useful, helpful, easy to learn, accessible, attractive, fun, connected, delightful and satisfying. Whereas a bad UX has the following characters: stressful, ugly, confusing, inefficient, distracting, tedious, frustrating and condescending. (Newman, 2017).

The attributes of UX can be simplified and break down into four major components based on Frank Guo's grouping (UXmatters 2012) as follows:

- **Value** – provides answers to the following questions: Is it useful? Does it accomplish what it should? Is it better than the alternatives?

- **Usability** – answers to the question: Can users do what they need to do?

- **Adoptability** - answers to the questions: Is it fun, attractive and pleasant to use?

- **Desirability** – answers to the question: Is it easy to find and to start use?

It is important to understand that the user interface is only one part of the total UX however it is very important. For example, a user interface might be excellent but if any information is missing in the underlying database it ruins the overall UX. (Nielsen , 2012).

2.4.2. History of UX

The evolution of UX is considered starting in the 1900's when Winslow Taylor pioneered the modern optimization of work by researching the interaction between workers and their tools. In 1940's, this was followed by Toyota by introducing their Production System which was a human-centred improvement process. It aimed to increase the efficiency, performance and quality by focusing on respecting people and improving processes instead of tools and technologies. *(*Marcin, 2014).

This was followed in 1940's with the beginning of the computer science when Alan Turing formed the first theoretical computer and became the father of theoretical computer science and artificial intelligence. This opened the gates for the development of the products and the future user experience design. *(*Marcin, 2014).

Soon after this in 1955, Henry Dreyfuss published the book "Designing for people" where he stressed the connection between people, their experience and successful design of a product. In his design, he applied common sense and a scientific approach. This resulted in significant contributions to human factor analysis and consumer research. *(*Marcin, 2014).

During the 70's, the personal computers were invented and introduced to public that created a new era with all the related devices and concepts. Design was seen as an im-

portant part of the development of those products from the beginning and the understanding about users and their needs became an approach to the new User-Centred Design. *(*Marcin, 2014).

In the 1980's , Don Norman who is specialized in the fields of design, usability engineering, and cognitive science wrote the *The Design of Everyday Things* book where he used the term "user-centered design". This design is "*based on the needs of the user, leaving aside what he deems secondary issues like aesthetics. User-centered design involves simplifying the structure of tasks, making things visible, getting the mapping right, exploiting the powers of constraint, designing for error, explaining affordances and seven stages of action*" (Norman, 2013). Later on he coined the term User Experience to describe the broad set of activities that his team was engaged in at Apple Computers.

According to Giardi (2016, p.230) the User-Centred Design approach was standardised by the ISO 13407:1999 "Human – centered design processes for interactive systems" (Revised by ISO 9241- 210:2010 "Ergonomics of human - system interaction - Part 210: Human - centered design for interactive systems"). This brought four main activities for the User-Centred Design process: specify the context of use and requirements; create design solutions and evaluate the design.

Nowadays, the term UX is very well known and it itself already explains what it is about.

2.4.3. Lean UX

The concept Lean UX combines the ideas of Lean with the UX Design process. The main principles taken from Lean are the follows:

- To remove waste from the UX design process by moving from detailed design documentation towards a light documentation concept with few handoffs and minimal deliverables,

- To engage people in all levels of the organization to collaborate in the iterative design process,

- To adopt a new way to design based on the rapid experimentation and measurement model where the designer's role changes to a facilitator.

The Lean UX process starts from a vision and goals for the project that can be broken down to assumptions, hypothesis and outcomes,. Assumptions are things that are believed to be true, hypotheses are the areas of the product that are targeted and outcomes are qualitative or quantitative criteria that will prove or disprove the hypotheses. Once the hypothesis are gathered and prioritised then the collaborative design phase starts. (Gothelf, Seiden, 2016, p.31).

Once the design phase is finalized, the building phase starts (also called Minimum Viable Product (MVP)). It means that the smallest outcome from the design phase is enough to be built for testing the assumptions. (Gothelf, Seiden, 2016, p.71).

The process ends in the research & learning phase once the hypotheses have been validated based on feedback received. (Gothelf, Seiden, 2016, p.89).

The following figure illustrated the Lean UX process described above.

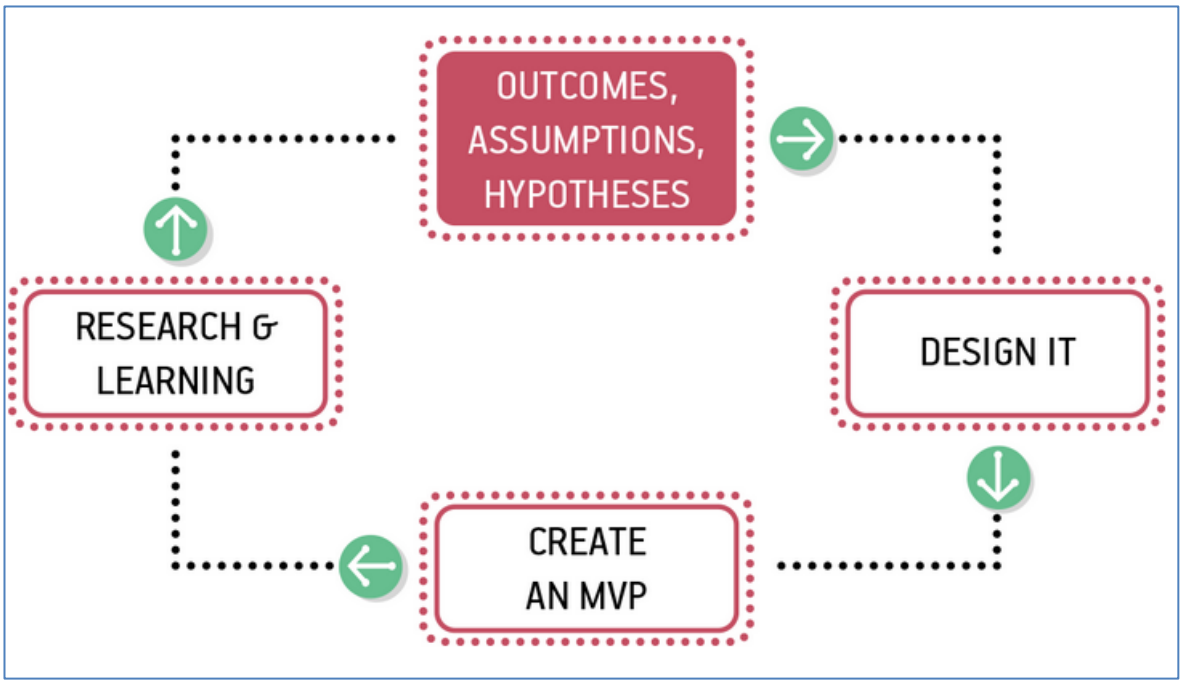


Figure 7. The Lean UX process (Gothelf, Seiden, 2016, p.18).

2.5. UX Research and Design Methodology

In this chapter, we examine a methodology to create a product or a system that provides good UX. This methodology was presented by Newman (2017) in the UX501x Introduction to User Experience course of EdX.

The key of this methodology is based on a user-centric research and design which forms an iterative prototyping process containing three sequential phases: Asses, Design and Build.

The next figure shows the iterative process and how it integrates the three phases with the UX Research and UX Design methodology:

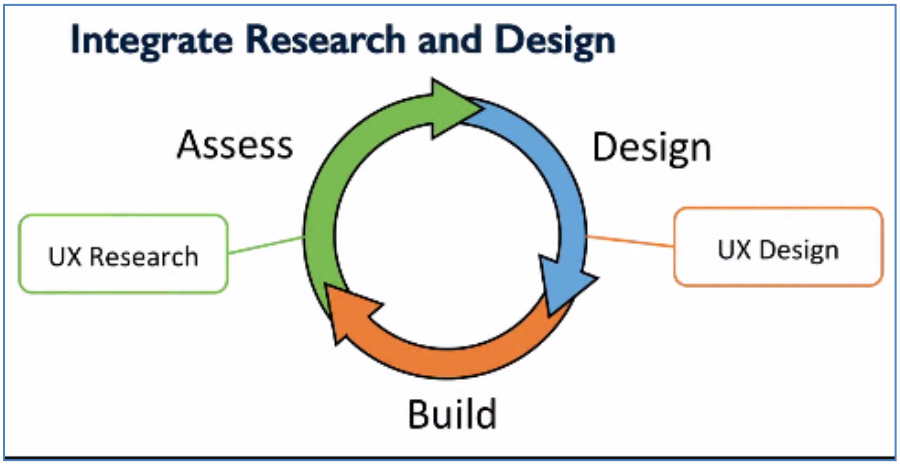


Figure 8. Process of iteration, (Newman,2017).

The iterative process starts from the assess phase where designers try to understand what users are doing and what their needs and possible problems are by applying the methods that UX research provides. The output of this phase is given to the design phase for quickly generating several solutions and ideas to meet the needs. The best ones are selected and handed over to the build phase where prototypes or mockups are created. These outputs are presented to the users to test the ideas, to find possible new issues and needs and to ensure that the development is heading to the right direction. This interaction directs and steers the output towards a good UX.

After this, the cycle starts again assessing the feedback from the previous cycle, designing and redefining new ideas, building new prototypes and assessing them. This cycle happens many times until the users are satisfied with the output. This process has the form of a spiral that ends when the final user satisfaction has been reached. The spiral model is shown in the following figure with the starting point located at the beginning of the spiral and the end point in the middle of the picture where the process ends.

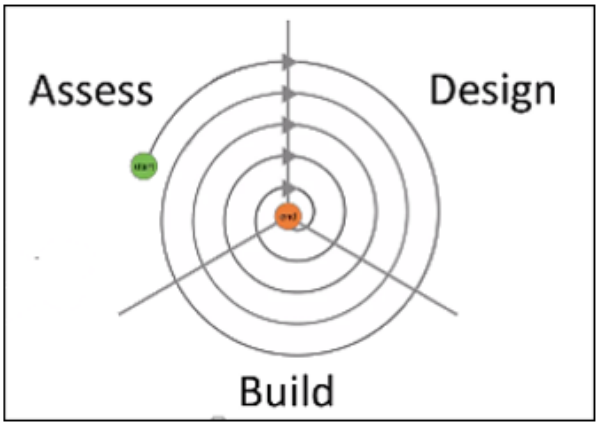


Figure 9. The spiral model, (Newman, 2017).

2.5.1. UX Research

The purpose of the UX Research is to steer a product or a system towards a great UX by understanding what users need and to evaluate the prototypes and mockups as they are being developed. Furthermore, it aims to address all attributes (value, usability, adaptability and desirability) of UX.

The following methods that can be used to facilitate the UX research:

- *Ask* - users can be asked in the format of interviews and surveys to understand the issues in more detailed level. The interviews are private and targeted to a reduced number of people whereas the surveys are used to reach a very large audience

- *Observe* – in this format, the interaction of users with the system is observed to ensure that all relevant elements that they might not be able to tell are captured. This method can be used also in user testing and to find usage patterns.

- *Inspect* – used for studying guidelines and best practices to compare what works and what does not in the prototypes and then to determine whether the development is on track for delivering a good user experience or not.

- *User or usability testing* – a test where users try to accomplish tasks using the product. This can be observed to confirm if the product works as expected or not, where the issues are located and what elements are missing.

2.5.2. UX Design

In general, design is considered to be mostly about how a product or a system works and about its functionality. The look of a product belongs to the scope of the design but it is not that important aspect as the first mentioned. Therefore, design can be seen as finding and solving problems of users. Furthermore, design can be seen "*as a plan for arranging different elements in such a way as to best accomplish a particular purpose and for producing an outcome*". This definition comes from Charles Eames who was a well-known mid-century furniture designer. (Eames Official Site, 2017).

To achieve a good design, UX designers need to use common sense and have a basic understanding of how humans work. This includes the notion of how people perceive things through their senses, how people act and process information and how emotions play a role in decision making.

The following specific methods can be used in UX design:

- *Personas, scenarios and user stories* are the descriptions from the perspective of users and what they need to achieve with the product.

- *Sketching and ideation* is meant for producing ideas.

- *Storyboards* are used to understand the interactions needed with the product.

- *Mapping and navigation* are used to map out all the different things that can be done with the product and ensure that they all make sense together.

- *Comparative research* can be used to study similar products to find the best practices available or things to be avoided.

- *Prototyping* is used to build preliminary versions of the final output.

3. Artificial Intelligence in IT Service Management

3.1. Introduction

Artificial Intelligence (A.I.) is becoming one of the most disruptive technologies ever created since it is changing the way we do currently business by increasing the productivity and efficiency to levels never seen before.

Some of the main reasons why A.I. is expanding now faster than in previous years and in different areas are because of the:

- Increase of computer power capacity.
- Creation of advanced algorithms.
- Explosion of information.

If we take a look to the A.I. within the IT Service Management field, we can see it is still in an early stage of development and adoption, however major industry leaders such as "Service Now" is moving quickly toward that direction after the acquisition of different A.I. companies.

As we are seeing in other fields such as the car industry the adoption of A.I. is expected to happen progressively, in which automation and manual processes will need to co-exist for several years until fully automation is adopted.

3.2. Why A.I. will be a key player in IT Service Management?

One of the main reasons why A.I. will become a key element in IT service management is due to the digitization of the real world thanks to the internet of the things (IoT). For this reason most of the items we know in the real world (such as houses, cars, cities, etc.) will become smart and connected 24 hours 7 days a week.

According to Gartner (2015), by 2018 things as customers will drive six billion support requests, requiring businesses to develop new customer support capabilities.

In order to deal with such explosion of support requests, the acceleration and integration of A.I. within the IT service management area will become crucial, and for those early adopters, it will mean a competitive advantage over those companies not using it.

As an immediate consequence, there will be a reduction in many of the tasks currently performed by Service Desk officers, especially those repetitive ones without any added value (such as replying simple questions by email, phone, etc.).

According to Gartner (2017): "*By 2019, IT service desks utilizing machine-learning enhanced technologies will free up to 30% of support capacity*".

As the technology gains in maturity, the integration of A.I. will expand progressively across all the IT service management areas, changing the traditional reactive approach in which end users need to contact the Service Desk to report an incident or service request toward a proactive approach in which failures will be mitigated before they happen or end user needs will be anticipated after analysing customer behaviour and records of previous requests.

This proactive approach will require less and less human intervention from the operational point of view, however until things get automated the collaboration and alignment between the people involved in the different ITSM processes will be needed, from Data Scientist, Agile methodology Scrum Master, Software engineer to Infrastructure experts, application specialists, IT operations, etc.

Below it can be seen a graph showing the correlation between the data analytics collected and the human effort needed as the integration of A.I. in ITSM increases.

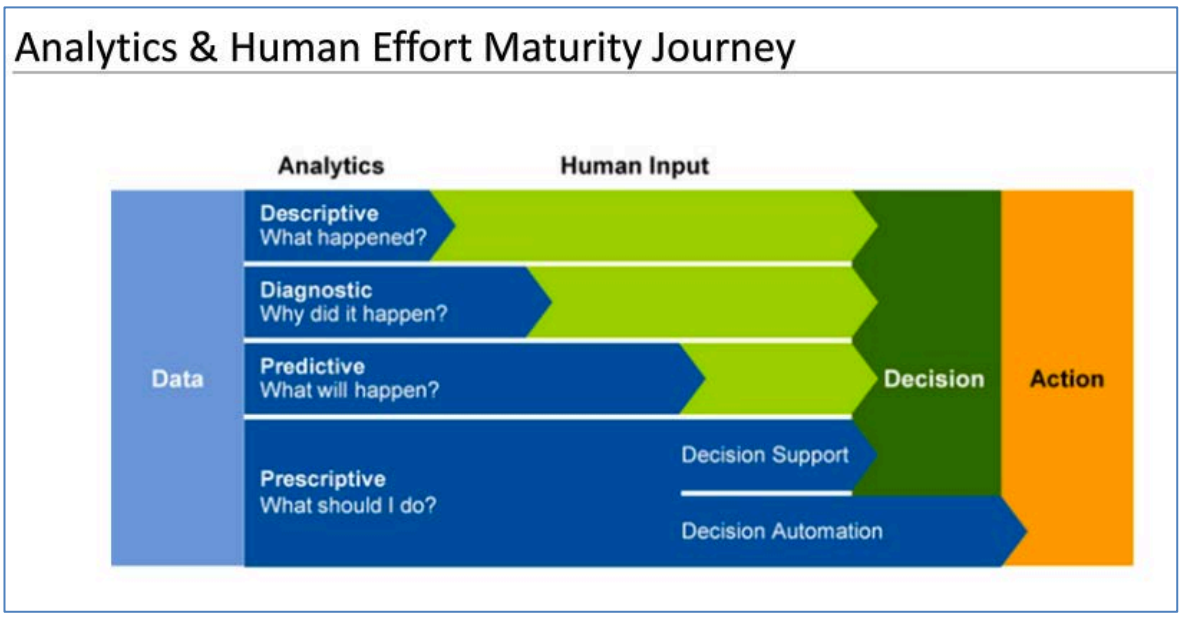


Figure 10. Effort maturity journey in ITSM (ServiceNow, 2017)

3.3. Virtual Agents

A virtual agent is a "*computer generated, animated, artificial intelligence virtual character (usually with anthropomorphic appearance) that serves as an online customer service representative. It leads an intelligent conversation with users, responds to their questions and performs adequate non-verbal behaviour*" (Chatbots, n.d.). Some of the most popular virtual agents nowadays are for example Siri from Apple, Google now, IBM Watson, Cortana from Windows or Alexa from Amazon.

Within the ITSM market, the virtual agents most known are MyIT from Remedy BMC and Sofi.ai from ServiceNow. They are able to understand different languages, which simplifies and speed up the interaction with customers using natural processing language in an audio or text format.

Virtual agents use machine learning algorithms that are continuously improved and need to be trained with historical data from tickets, user behaviours, etc. in order for example to select appropriate answers from a knowledge management database or to propose suggestions to the end user.

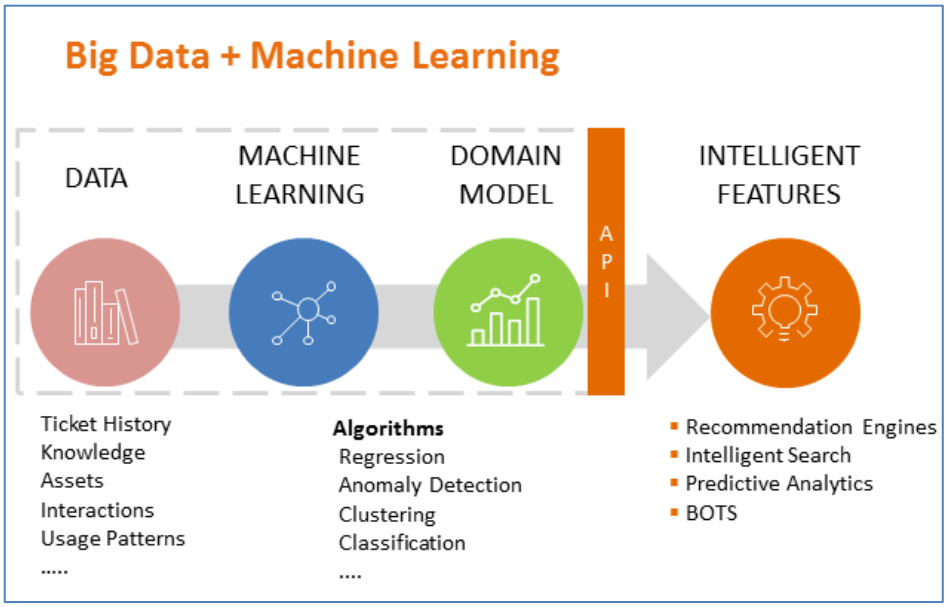


Figure 11. Combination of Big Data & Machine learning in ITSM (Hall, n.d.)

Machine learning algorithms can also incorporate sentiment analysis, which can help to detect the frustration of a customer and escalate selective requests that need human intervention having as a final consequence an increase in the user satisfaction. Another positive aspect of using machine learning is the automatic classification and rooting of tickets to the correct helpdesk support groups. However, the real potential for maximizing

the use of virtual agents and machine learning is not only in the capacity of interacting with customers and analyse data but to take decisions and act on other systems to resolve incidents or attend service requests. Such capacity is limited to the level of integration that can be achieved between the different applications and systems involved in an ITSM process.

In many cases the integration can very complex or costly due to the inheritance of existing legacy systems. Major ITSM vendors as Service Now includes third party integration capabilities with other applications such as SCCM, Splunk, Appdynamics, Jenkins and Active Directory)

Another limitation can be the adaptation time from human beings to changes, many people may want traditional ways of requesting information with a human touch instead of using a virtual agent. Also, the interaction that can be achieved nowadays with end customers using Natural Language processing or visual elements works at a very low bandwidth. In the future interaction will be faster thanks to Brain Computer Interfaces or other kind of faster devices.

Below there is a list of the most relevant benefits of using A.I. in IT Service Management:

- Reduction of support costs and human resources in the front line helpdesk.
- Reduction of human errors and time employed to train service desk officers.
- Increase in support hours for helpdesk and improvement in the dispatching process of tickets.
- No customer impact due to IT maintenance by having a more predictable environment.
- Capacity to take decisions and act on other systems (for example: instant fulfilment of an IT customer demand when people need infrastructure for projects).
- Enhancing Strategic Decision-Making

4. Research design

In this chapter we describe the research methods and analysis techniques used in our project.

4.1. Research methods

The next figure illustrates what methodological choices are available to create a research strategy (the ones located in the inner sections have a broader focus than the ones located in the outer sections):

Figure 12. Available research strategies – methods, (University of Jyväskylä, 2010)

In our project we applied the following methods:

1. Empirical research

   Penn State University Libraries (2017) describes the empirical research as follows: *"An empirical research is based on observed and measured phenomena and derives knowledge from actual experience rather than from theory or belief".*

   Reflecting this in our project, we used the empirical method by observing and measuring real data collected and analyzed from the Requester Console and the Reference Configuration.

2. Qualitative research

QRCA (n.d.) defines the qualitative research as follows: *"It uses in-depth studies of small groups of people to guide and support the construction of hypotheses. The results of qualitative research are descriptive rather than predictive.*"

This research method was used to collect data in the format of interviews for the Requester Console and through focus groups and brainstorming for the Reference Configuration. We used it because we wanted to dig into the reasons why front-end users were dissatisfied with the existing Requester Console and to understand why the Reference Configuration became so complex.

3. Quantitative research

USC Libraries (2017) describes the Quantitative research as follows: "to *emphasize objective measurements and the statistical, mathematical, or numerical analysis of data collected through polls, questionnaires, and surveys, or by manipulating pre-existing statistical data using computational techniques. Quantitative research focuses on gathering numerical data and generalizing it across groups of people or to explain a particular phenomenon.*"

This research method was used in the interviews done to collect data for the Requester Console with the purpose of measuring which was the user satisfaction before and after the revamp.

It was also applied in the data collection of the Reference Configuration extracting quantitative data from the database related to the Ticket Management Module.

4. Case study

Writing@CSU (2017) defines a case study as: "*The collection and presentation of detailed information about a particular participant or small group, frequently including the accounts of subjects themselves. A form of qualitative descriptive research, the case study looks intensely at an individual or small participant pool, drawing conclusions only about that participant or group and only in that*

specific context. Researchers do not focus on the discovery of a universal, generalizable truth, nor do they typically look for cause-effect relationships; instead, emphasis is placed on exploration and description".

Referring to the aforementioned definition, the content of this project was developed within the specific context of our target organization focusing and studying only a small group of research objects in an isolated environment. The outcomes and conclusions of this project are not meant to be extrapolated as a generalizable truth.

4.2. Analysis techniques

To analyse the data collected during the project we used the following analysis techniques:

1. Affinity diagram

   It is a method that allows ideas to be sorted into groups based on their natural relationships. The term Affinity diagram was devised by Jiro Kawakita in 1960. (Wikipedia, n.d)

   We used the Affinity diagram to group and organise all the ideas and raw data collected from the interviews and focus groups using a bottom-top approach for both, the Requester Console and the Reference Configuration.

2. MoSCoW method

   It is a method widely used in iterative developments and originated from the Dynamic Software Development Method methodology. (Waters, 2009).

   The initials of MoSCoW stand for the prioritization of features into four categories:

   - **Must** have features are critical deliverables of a project and without them the success cannot be achieved. They can be referred as a minimum usable subset.
   - **Should** have are the features that are considered to be valuable but not critical.
   - **Could** have are the features that are nice to have but not necessary. In case, there will be a risk with the time these will be firstly removed from scope.

- **Won't** have are the features that are requested but cannot be delivered in the scope of the project during the planned duration however maybe implemented in future developments. (Waters, 2009)

The ideal balance of the total effort of a project should be divided into 60 % to "Must have" features, 20 % to both "Should have" and "Could have" features (Leclercq, 2016, p.72).

We used the MoSCoW method in both the Requester Console and the Reference Configuration. It helped us to prioritise and select those requirements that we consider relevant to be implemented during the development phase.

3. Pareto principle

It is a very simple principle and theory discovered by an Italian economist Vilfredo Pareto in 1906. He noticed that things in life are not distributed evenly and that some of them have a higher contribution than others. For example, 20 percent of Italians owned 80 percent of the properties in Italy. Since them the rule became known as the 80/20 rule since it states that "*80 percent of the output from a given situation or system is determined by 20 percent of the input*". (Rouse, n.d.a).

We used the Pareto principle in the analysis of the data collected for the Reference Configuration. It helped us to discover the distribution of those labels that were used more frequently in the tickets classified.

4. Ishikawa diagram

It is named according to its Japanese creator Kaoru Ishikawa who aimed to create a visual tool for identifying root causes in a specific problem.

The diagram is also called the fishbone or cause-and-effect diagram. It looks like a skeleton of a fish having its head facing to the right and the spine starting from the head to left. The spine is connected with several main fish bones and the head contains the description of the main problem to be researched. The fish bones contain the main causes of the problem. Each fishbone is divided into smaller bones with more details. (Rouse, n.d.b.).

.

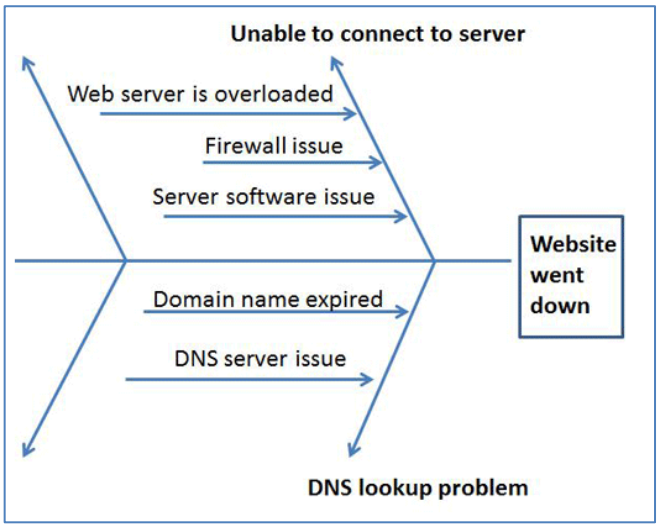


Figure 13. An example of the Ishikawa diagram, (Rouse, n.d.b.).

We used the Ishikawa diagram to represent in a visual way the conclusions obtained in the data analysis of both the Requester Console and the Reference Configuration

4.3. Research strategy

In the planning phase of this research, we created a strategy to help us to conduct the project to reach its objectives and to find the answers to the research questions. The research strategy can be seen as a collection of guiding principles and methods to be used.

The overview of our research strategy is illustrated in the following figure indicating the methods used in the different phases of the project.

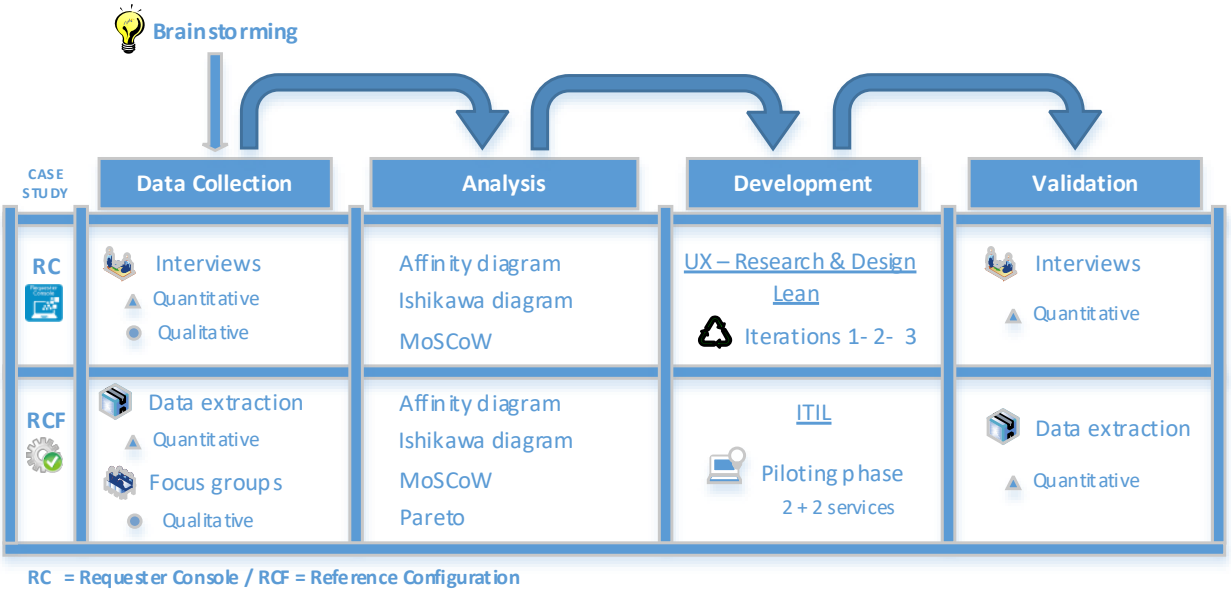


Figure 14. Research strategy workflow

5. Data collection

5.1. Requester Console

The methods used to collect data from the front-end users were face-to-face and phone interviews.

The interviews were done by back-end users (IT support staff) when front-end users called the IT Help Desk asking for support or when they met face-to-face during service situations. We selected this approach since front-end users are usually very busy and it would have been very time consuming to allocate exclusive time slot for this purpose. Therefore, the time needed for going through the questions was planned to be short and the answers were immediately noted down by the support staff.

The interviews took place in November and December 2016 approximately during one month without having a fix schedule but with a just in time approach. For each of the face-to-face and phone interview there was not time restriction however most of the interviews did not last most than 15 minutes.

For the creation of the questionnaire it was decided to divide the questions formulated into four qualitative questions (with sub questions depending on the answer) and one quantitative question.

5.1.1. Questions

The following questions were used in the interviews:

1) The last time you created a ticket, were you able to find the information/service easily?

1. *If Yes - Could you please explain why? Could you please try to find the service Teleworking*
2. *If Not - Could you please give us an example of a service you had problems to find?*

We created this question to know if front-end users were able to find the information or not through the interface.

For those who answered “Yes” the purpose of the sub-question was to dig into the reasons why they were able to find the service. To verify the validity of their answer we asked

them to search for a particular service called Teleworking and check if they gave an honest answer.

For those who answered “Not” the purpose of the sub-question was to collect the reasons why they were not able to find the service.

2) How do you think you would be able to find the information/service in an easier way?

This question pretended to collect improvements, ideas and suggestions about which elements will make easier to find the different services or products.

3) Do you check or update the status of your tickets in the requester console interface?

1. *If Not - Could you please explain the reason/s why?*
2. *If Yes - Do you ever look at their status?*

This question pretended to collect if the user really uses the Requester Console for something else than just creating new tickets and if they know that apart of the basic functionality there are other available.

4) Do you ever read the notifications sent by the requester console when a ticket is created or resolved?

1. *If Not - Why?*
2. *If Yes - Which ones do you read? Is the relevant information?*

The purpose of this question was to know if the notifications sent by the Requester Console are read and if they are really useful for the front-end users.

5) In a scale of 1 to 5 (1 really difficult 5 - really easy) How easy or difficult is to create a ticket using the Requester Console?

This is a closed question with the intention to collect quantitative data about the Requester Console as a tool (not about the quality of the IT services provided). The reason why this question was created was to collect quantitative data that could be compared with the results obtained in a future survey launched after the go-live of the revamped Requester Console. Asking the same question in both surveys allowed us to measure if there was an increase or decrease in the user satisfaction.

5.2. Reference configuration

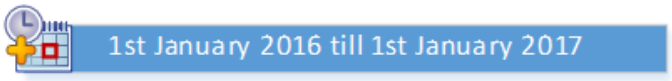
The methods used to collect data for the Reference Configuration were data extracted from a database (quantitative) and brainstorming in focus groups (qualitative).

Data related to the Reference Configuration was extracted using a SAP Business Objects reporting tool. This tool is connected to the Ticket Management Module's database that records all service requests and incidents received as tickets. All tickets contain a unique identifier of 8 digits and multiple fields such as the contact details of front-end users, urgency, impact, etc. Some of the fields are filled by front-end end users when the ticket is created and others need to be filled manually by back-end us when resolving the ticket.

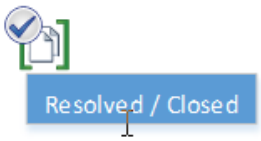
Focus groups sessions used the data extracted from Ticket Management Module's database as an input for brainstorming and generate ideas about how to improve the current Reference Configuration and understand the reasons why it became so complex.

The criteria used for the data extraction within the context of the project were the following:

- Time period



- Status



- Fields extracted

The fields "Service" and "Support group" are composed by a single level of granularity. On the other hand fields "Product" and "Operational Categorization" are composed by 4 and 3 levels of granularity. Each level of granularity has the structure of a tree in which next levels act as children of the previous ones.

Below an example of how a resolved/closed ticket looks like:

Field name	Description	Example
Reference number	Unique identifier	INC00009832
Received Date	Date the ticket was created	21/06/2017 10:00:21
Resolved Date	Date the status of the incident was set to resolved	22/06/2017 15:17:54
Service	Name of the service	Email
Support Group	Name of a group of people dealing with a ticket	Email Level1
Product Cat. Level 1	First level of detail	Software
Product Cat. Level 2	Second level of detail	Application
Product Cat. Level 3	Third level of detail	Standard Workplace
Product name	Name of the product	Microsoft Outlook
Operational Cat. Level 1	It is compose by the name of a service or a product.	Email
Operational Cat. Level 2	Details about the incident or service request.	Functional Mailbox
Operational Cat. Level 3	A more detailed description of the action.	Increase quota

There are five possible statuses for each ticket, which can be:

Status	Description
Assigned	It means the ticket has been created by the front-end user and assigned to a support group.
In progress	An IT officer is currently working on the ticket.
Pending	The IT officer dealing with the ticket is expecting an action from a third party or a front-end user.
Resolved	The ticket has been resolved and it will remain in this status during 15 days. During this time the end user may come back.
Closed	The resolved ticket has been automatically changed to this status after 15 days. It is not possible to reopen the ticket any more.
Cancelled	The ticket will not be counted.

The steps done to collect quantitative data are represented in the workflow below:

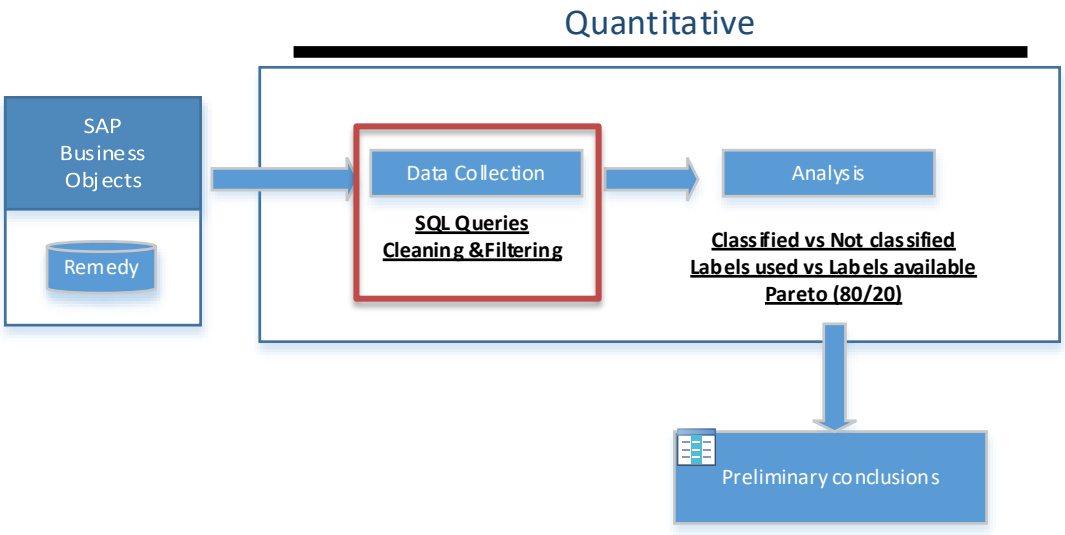


Figure 15. Data collection phase (Quantitative) of the Reference Configuration

Focus groups

A total of 12 different focus group sessions were run during a period of three months.

The focus groups were composed by a diverse group of active back-end users (Service Managers, Product Managers, etc.) and their sizes were usually around four to six people.

Different topics were addressed during the focus groups, however, we will only mention the ones that were relevant to the scope of this project.

The focus groups helped to analyse deeper the quantitative data obtained and to combine with a qualitative analysis trying to understand reasons why it happened and what it is possible to do to improve the current situation.

During all the sessions, the same structure was followed for each of the focus groups as indicated below:

- Presentation of all the preliminary conclusions (data extracted from all services and overall conclusions).
- Presentation of specific data related to the service or services the participants of the focus group were affected by.
- Brainstorming / discussion (two qualitative questions).

The sessions were conducted in a collaborative approach having discussions and brain-storming trying to find an answer to the following questions:

1. ***Why did it happen?***
2. ***What can we do to improve the current situation?***

The steps done to collect qualitative data are represented in the following workflow.

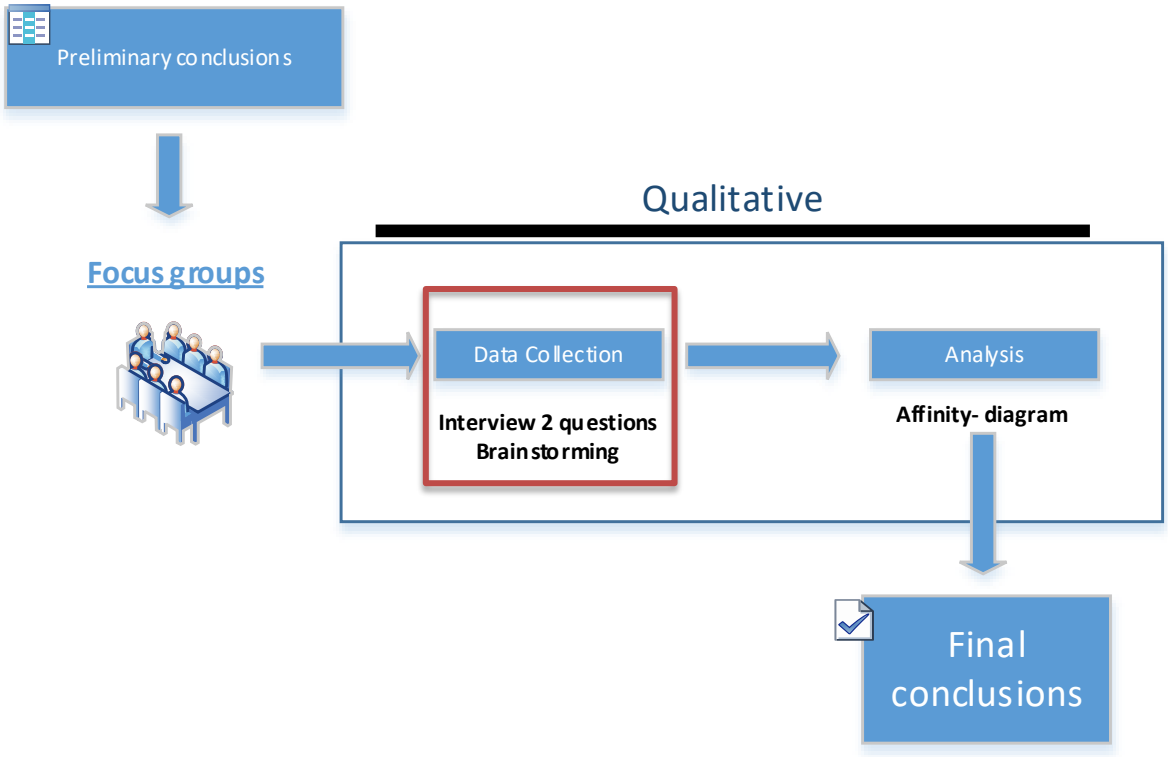


Figure 16. Data collection phase (qualitative) of the Reference Configuration

6. Analysis

6.1. Requester Console

In total 40 people were interviewed with a gender balance of 29 women and 11 men during a two months period. The analysis for the qualitative questions and the unique quantitative question was done separately as explained in this chapter.

Qualitative questions

In order to group the data obtained from the qualitative answers, the affinity diagram was used to cluster and organise the information using a bottom-top approach. As an outcome of this process, we ended up having five main topic groups based on the natural relations and connections of the replies obtained as indicated below:

CONTENT

- The content of the Requester Console should be reviewed and rewritten in a user friendly language without using technical abbreviations and incorporating keywords that are understandable by non IT users.

- A meaningful service card should be created indicating in a simple way what the service can do for the customer. (This is out of the scope of this project).

- A unified service catalogue should be created in order to avoid having duplicated information in the intranet of the target organisation and the Requester Console. (This is out of the scope of this project).

FUNCTIONALITIES

- There are many functionalities missing in the Requester Console that could be added progressively following the ITIL approach of continuous service improvement and in agile iterations.

LAYOUT

- The current layout is very poor visually. For example, it does not contain icons or any other visual element that would help front-end users to quickly identify and locate services.

- The information is not sorted per relevance and many clicks are needed to create a ticket.

- Due to a messy layout, the search functionality was unnoticed often by front-end users.

- The layout of the email notifications should be improved by eliminating and simplifying the number of notifications and the amount of information sent.

STRUCTURE

- The current structure is confusing and definitely created from an IT perspective. It is difficult for non-IT people to use it.

- The structure should be aligned with what front-end users really understand and simplify by offering less choices and menus that expands as a user selects them.

WORKFLOW

- Requests that require the creation of several tickets to different services could be merged into a single request. This would save time, increase efficiency and remove waste.

Using the Ishikawa diagram, the fish bones (in red squares) represent the main topic groups which at the same time are decomposed into sub/groups:

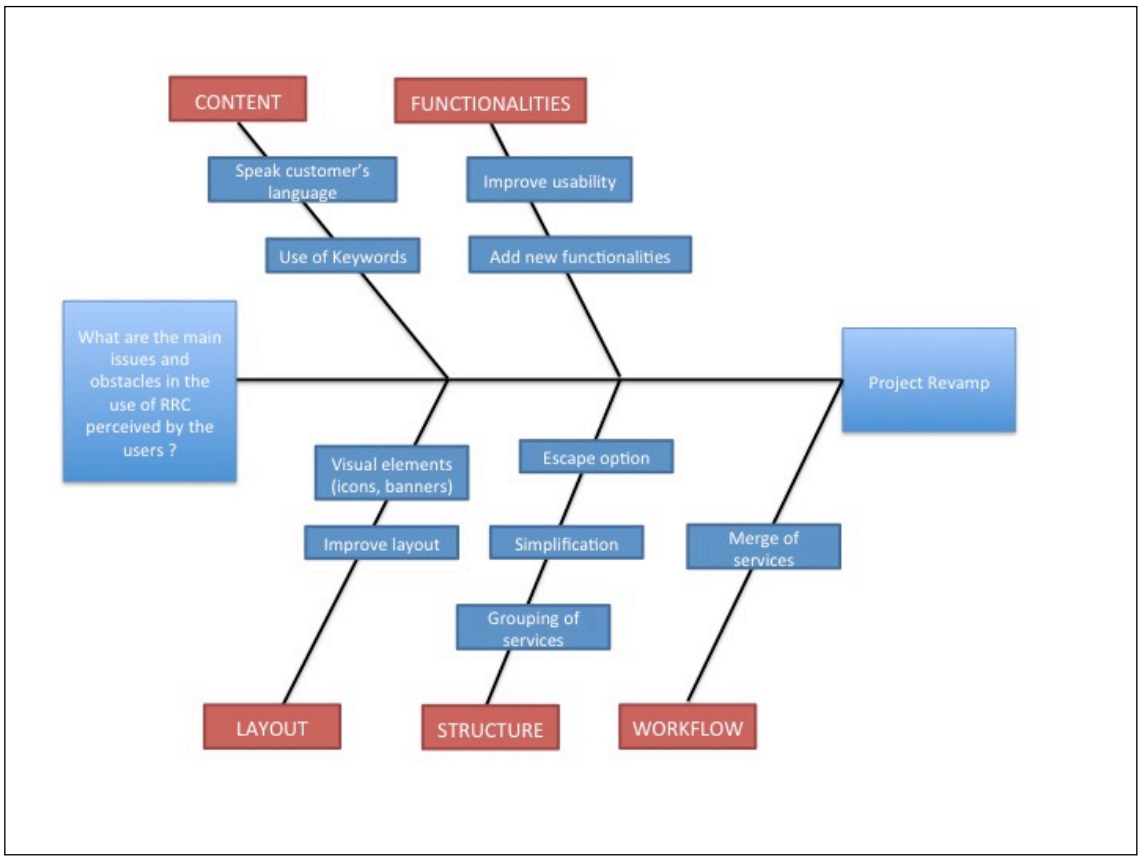


Figure 17. Ishikawa representation of the Requester Console analysis

After the analysis, we used the MoSCoW method to prioritise and select the features to be developed as it is shown in the table below:

MoSCoW	Features
MUST	• **Content** - Speak customer's language • **Functionalities** – Improve usability • **Layout** – Improve layout • **Structure** – Grouping of services • **Structure** – Simplification
SHOULD	• **Layout** – Visual elements (icons, banners)
COULD	• **Content** - Use of keywords • **Functionalities** – Add new functionalities • **Structure –** Escape option • **Workflow** - Merge of services
WON'T	• Not applicable

Quantitative question

Turning to the analysis of the quantitative question, the average result of all people interviewed was 3.24 points of a maximum score of 5 points.

One of the most common replies was: "*I gave 3 because the tool is working and I have been so many years in the company, always requesting the same things. However, if I had to select a different service or if I would be a newcomer it would be difficult to find the information*".

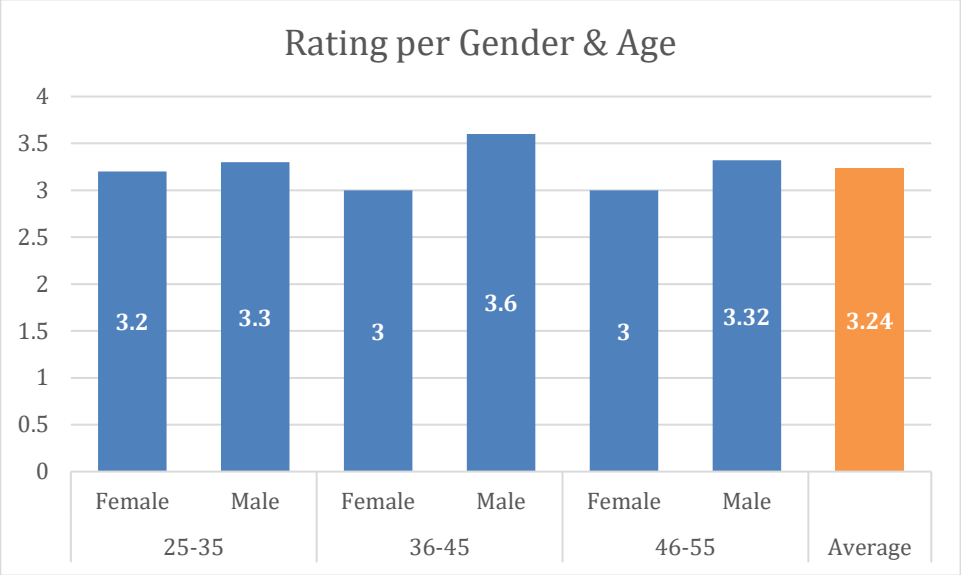


Figure 18. Survey quantitative results per gender / age

If we apply further granularity per gender, men in general gave a slightly higher positive score compared to women as shown in the picture below:

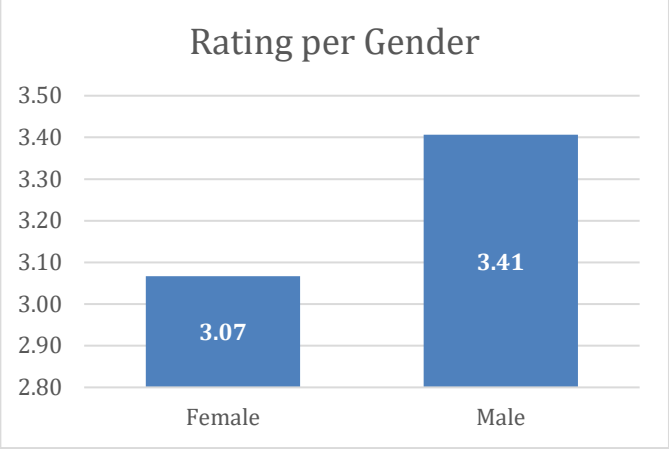


Figure 19. Survey quantitative results per gender only

Taking a look to the rating per age those in the age band of 36 to 45 gave the highest score.

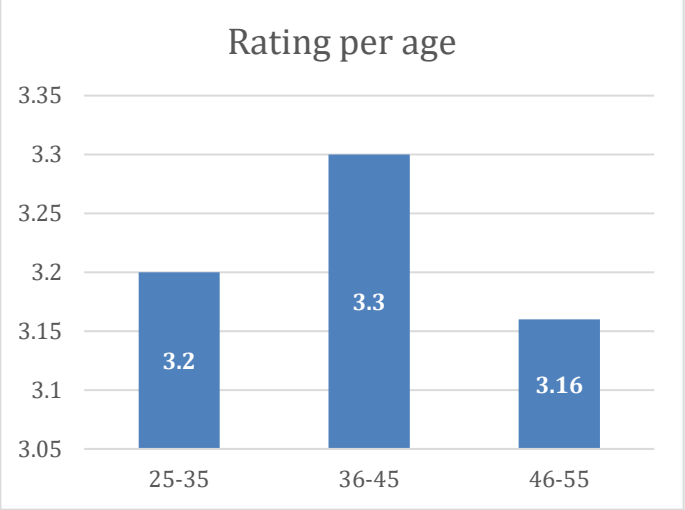


Figure 20. Survey quantitative results per age only

6.2. Reference Configuration

The results obtained from the data extracted using the SAP Business Objects reporting tool showed a total of 17878 tickets Resolved/Closed for a 12 months period.

The distribution among the different fields was the following:

Field name	Tickets Classified	Tickets not classified	Labels Available	Labels Used
Service	**15478**	**2400**	68	68
Support Group	**17878**	**0**	110	90
Product Cat. Level 1	**16345**	**1533**	6	6
Product Cat. Level 2	**16443**	**1435**	110	33
Product Cat. Level 3	**14798**	**3080**	49	21
Product name	**10790**	**7088**	130	72
Operational Cat. Level 1	**17878**	**0**	63	47
Operational Cat. Level 2	**10601**	**7277**	101	56
Operational Cat. Level 3	**5620**	**12258**	84	36

Where:

- Tickets classified are those tickets in status Resolved/Closed that contained a value for a specific field.

- Tickets not classified are those tickets in status Resolved/Closed contained an empty value (NULL) for a specific field.

- Labels available means the number of items/choices available for a specified field.

- Labels used means the number of items/choices that were really used from the list of available items for a specified field.

The data analysis was separated into tickets that were “Classified versus Not Classified”, “Labels used versus Labels Available” and the “Pareto” distribution as indicated in the workflow below:

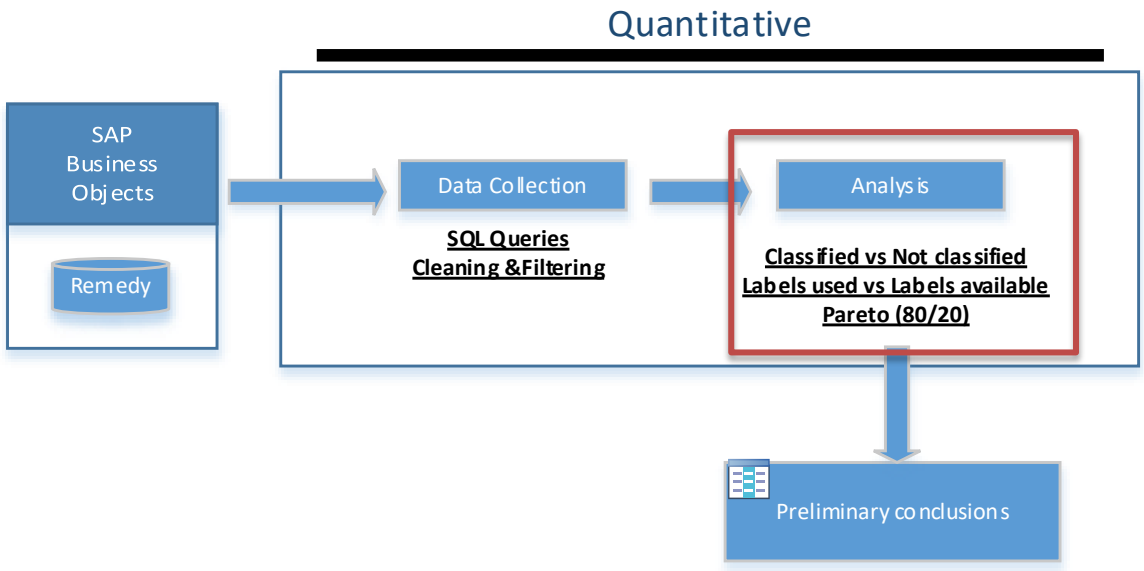


Figure 21. Analysis phase (quantitative data) of the Reference Configuration

Classified versus Not Classified

In the graphic below it is represented the percentage of tickets per individual field that contained a value ("Classified") versus those that contained a null value ("Not classified"):

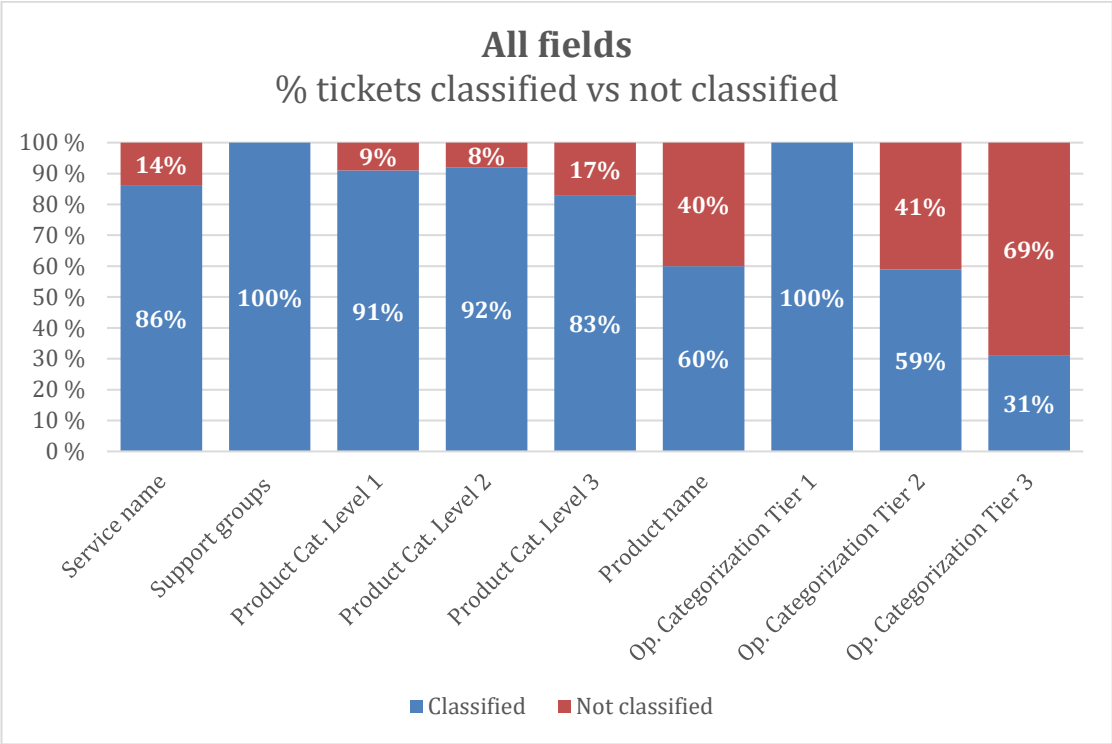


Figure 22. Bar chart representation of tickets classified versus not classified

The main conclusions obtained after the data quality analysis were:

- Only two fields "Support groups" and "Op. Categorization Tier1" were 100% classified.
- The fields with the highest percentage of tickets "Not classified" were "Product name" = 40%, "Op. Categorization Tier 2" = 41% and "Op. Categorization Tier 3" = 69% indicating that as the level of a detail or granularity gets deeper the probabilities of having tickets "Not classified" increases.

Labels Used versus Labels Available and Pareto Distribution

The conclusions obtained for the Labels Used versus Labels Available and the Pareto distribution are presented grouped into three blocks of fields:

- Service & Support group
- Product fields
- Operational Categorization fields

Service & Support group fields

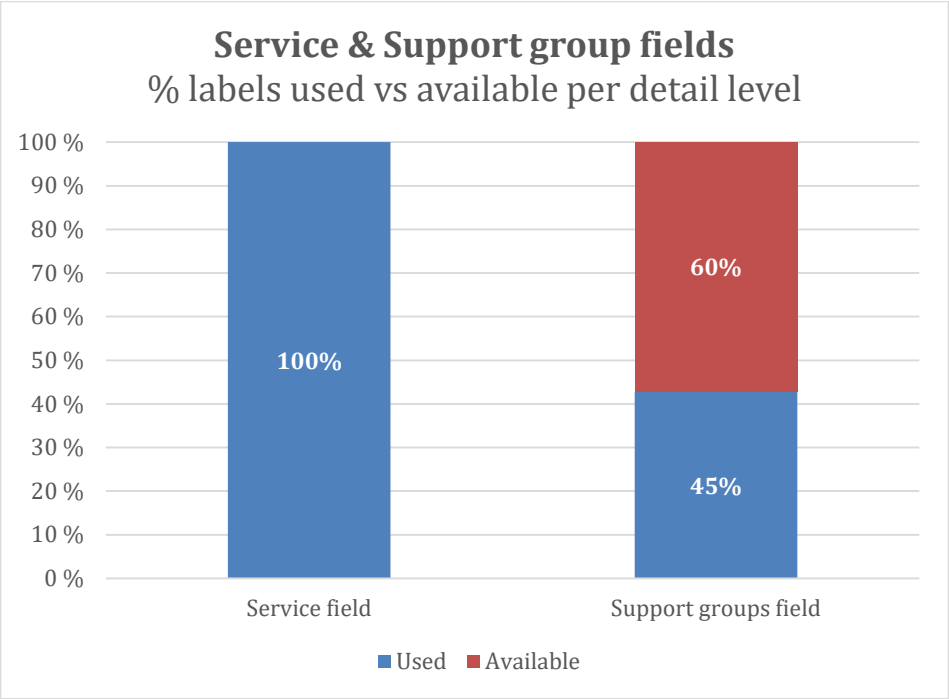


Figure 23. Bar chart representation of labels used versus labels available for Service & Support group fields

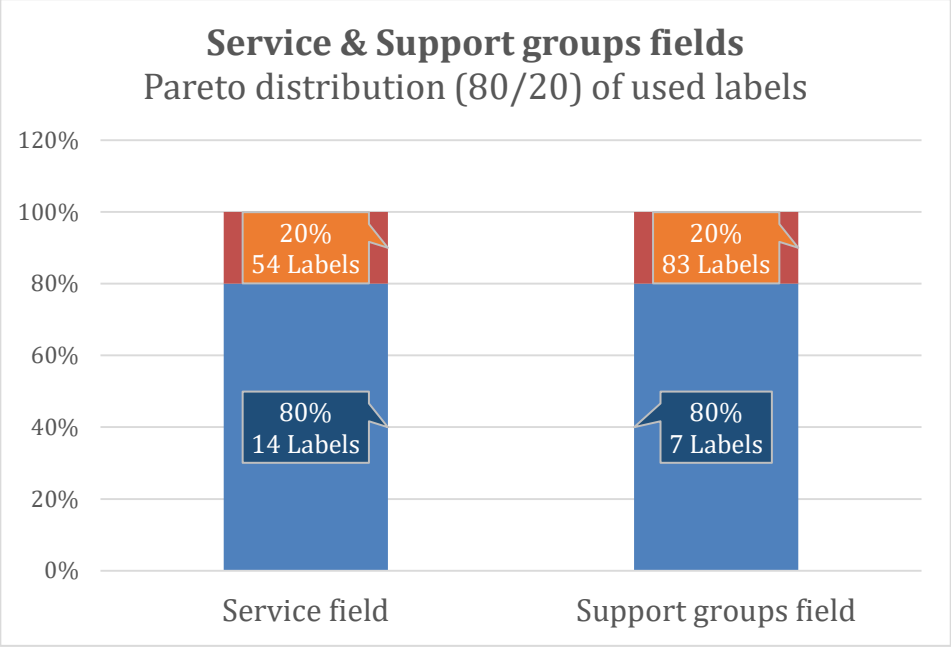


Figure 24. Pareto distribution of the labels used for Service & Support group fields

The main conclusions obtained were:

- For the “Service” field all labels available were used and 14 services concentrated 80% of the tickets resolved.

- For the “Support group” field less than half of the support groups were used and only 7 concentrated 80% of the tickets resolved.

Product fields

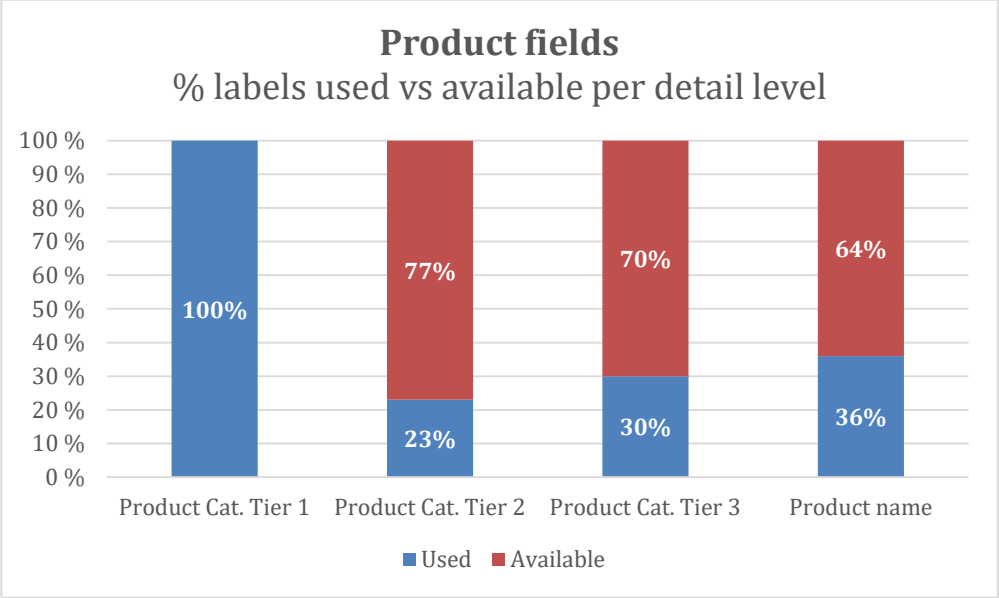


Figure 25. Bar chart representation of labels used versus available for Product fields

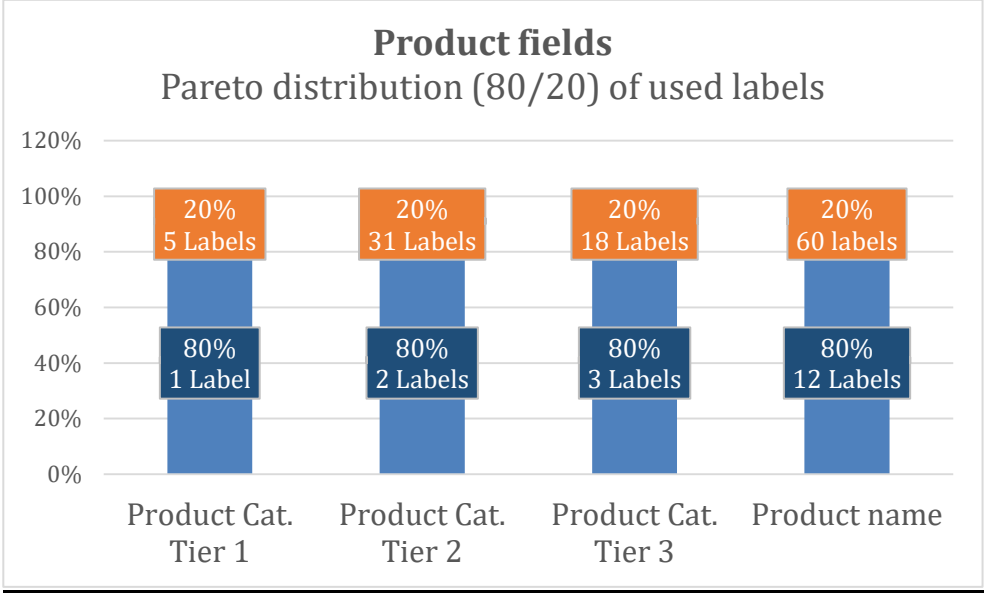


Figure 26. Pareto distribution of the Products fields

The main conclusions obtained were:

- The field “Product Cat. Tier 1” was the only one that used all the labels available.
- The rest “Product Cat. Tier 2”, “Product Cat. Tier 3” and “Product name” had a very low usage of the labels available with a combined average of approximately 30% used.

- The Pareto distribution for all the “Product” tiers reflected that a small number of labels concentrated the majority of tickets resolved and as the level of granularity increases the number of labels used decreases.

Operational categorization fields

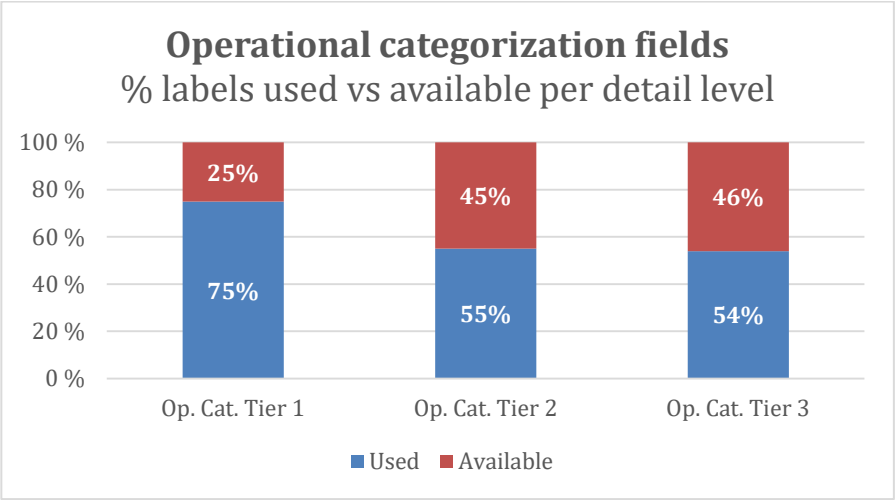


Figure 27. Bar chart representation of labels used versus available of the Operational Categorization field

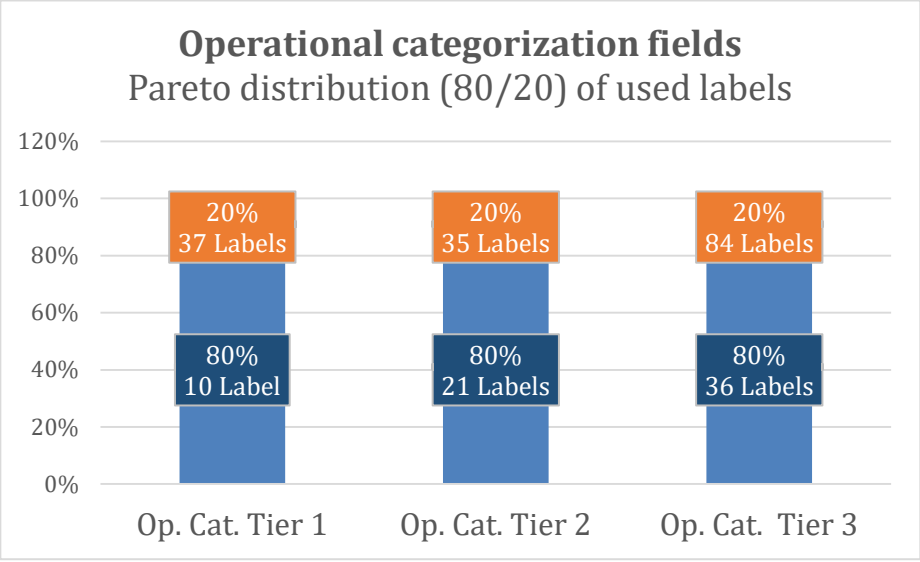


Figure 28. Pareto distribution of the Operational Categorization field

The main conclusions obtained were:

- For the “Op. Categorization Tier 1” most of the labels were used however for the “Op. Categorization Tier 2” and “Op. Categorization Tier 3” a bit more than half of them were used.

- The Pareto distribution for all the “Op. Categorization” tiers is similar to the “Product” categorization tiers and behaves in a similar way, meaning that a small number of labels concentrated the majority of tickets resolved and as the level of granularity increases the number of labels used do as well.

The overall conclusions from the analysis of all the tiers combined were the following:

- When the number of labels available for a specific field increases the chances for that field to be classified decreases.

- When the level of detail/granularity gets deeper the chances for that field to be classified decreases.

- For all the fields a small number of labels concentrated the majority of tickets “Resolved/Closed”.

Focus groups conclusions

All previous conclusions were used as input for the brainstorming in the focus groups as indicated in the graphic below.

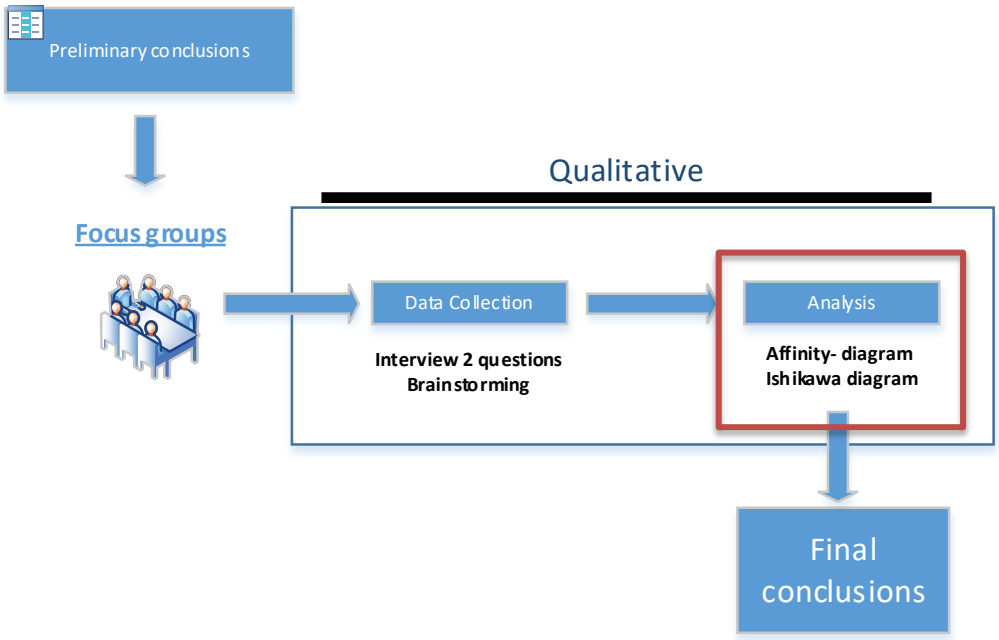


Figure 29. Analysis phase (qualitative data) of the Reference Configuration

The transcriptions collected from the focus groups' sessions were clustered using an affinity diagram and an Ishikawa graphic representation to visualize the most relevant conclusions related to the question: **Why did it happen?**

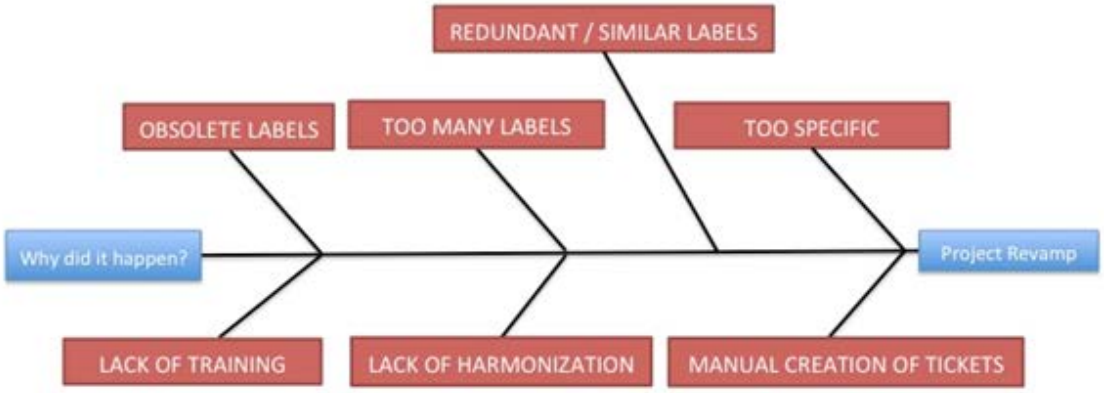


Figure 30. Ishikawa representation of the Reference Configuration

Obsolete labels

Some of the existing labels were obsoletes and not applicable any longer however they were never requested to be deleted. The main justification why those specific labels were never requested to be deleted was "just in case", meaning that potentially a ticket could be created for that specific product, service, etc.

Too many labels

Too many labels create confusion on the front-end and back-end users.

Redundant / Similar labels

Some labels used across different services have a very similar wording or meaning creating redundancy and difficulties to maintain them up to date.

Labels too specific / too detailed

In some services, the fact of having too detailed labels made them to choose in occasions random labels (just to have a value) since none of the available ones fit the context of

their tickets. Also it did not exist an “Escape” alternative such as “Other” to classify those tickets that did not fit using any of the existing labels.

Lack of training

For some back-end users the use of Remedy was still confusing and resistance was found in the willing to use the tool. Some of them directly did not like the tool, others did not know how to use it, which fields were supposed to be filled (too many tabs, buttons, etc.).

Lack of harmonization

Many labels used across different services did not follow a specific format or clear structure, in some occasions the name of a product contained acronyms and for others it contained the full description, sometimes the acronym was appearing at the end of a sentence for others it was at beginning combined with the full description. This created confusion when listing them in alphabetical order.

No added value

In some services, the real reason why they did not fill the classification was the effort (amount of clicks that was necessary to do to classify a ticket) versus the reward (nobody is using/reading the reports generated).

Manual creation of tickets

Most of the tickets are created by front-end user using the Remedy requester console, however, back-end users can also create tickets manually. It was noticed that most of tickets created manually were not fully classified due to the number of clicks that was needed to do while creating a ticket.

What would you do to improve the current situation?

There were multiple solutions proposed during the focus groups sessions and the main conclusions are clustered below in the MoSCoW approach for its future development:

MoSCoW	Features
MUST	• Maximize the mapping/alignment with the Requester Console. • Creation of labels with more abstract concepts / simplification. • Harmonization of all fields across services. • Creation of templates for back-end users.
SHOULD	• Evangelization of the tool / change of mind-sets / training sessions. • To create quality reports to control which tickets are not classified.
COULD	• To run one to one training sessions.
WON'T	• To make mandatory all the fields at the moment a ticket is resolved.

Maximize the mapping/alignment with the Requester Console

To maximize the mapping of fields (services, products, support groups, classification) between two perspectives (front and the back-end) so that as much information as possible is provided directly by the front-end users.

The responsibility to properly classify a ticket will always rely on the IT officer but by having most of the tiers completed by the end-user, it will require less effort and time spent on classifying the tickets manually and increase the ratio of tickets not classified.

Creation of labels with more abstract concepts / simplification

To replace all specific labels by more abstract concepts so that they can fit a wide range of tickets and the same labels can be applied across all services.

To create an escape option "Other" for the operational categorization and product fields for those tickets that cannot be classified using any of the existing labels.

Harmonization of all fields across services.

To create a harmonized structure and equal formatting rules across all services in terms of naming conventions, acronyms, positioning (pre-fix, su-fix), use of brackets, etc.

Evangelization of the tool / change of mind-sets / training sessions

To evangelize and convince back-end users about the benefits of using a proper classification and having good data quality. To promote the use of reports showing that “data matters” and they should be used regularly to detect bottlenecks, products that creates more incidents, etc. To run several training sessions across all the services to teach them how to use the Remedy Configuration.

Creation of templates for back-end users

To create templates with all fields pre-filled so that when tickets need to be created manually back-end users do not need to do all the classification manually. This will allow to reduce the number of clicks that back end users need to do to create and resolve a ticket.

To create quality reports to control which tickets are not classified

To create an automated report that detects which tickets were not classified and inform the owners of the tickets to classify them.

To run one to one training sessions.

To train individually those front-end users that selected wrong products in the Requester Console and back-end users that did not classify the tickets at all or that classify them wrongly.

To make mandatory all the fields at the moment a ticket is resolved

After consultations with all service managers this idea was finally discarded since some of them did not agree to make all the fields mandatory for the tickets in their services.
In addition, there is a technical limitation in the Remedy Configuration module that is not possible to apply the mandatory conditions for specific services but it needs to be applied to all.

7. Development of Requester Console

7.1. As-is state

This paragraph documents and describes the layout and the functionality of the Requester Console from where the research and improvement started.

As already explained in general in chapter 1.2 Requester Console, this tool offers an interface to the front-end end users to request services and report incidents for the services offered in the Reference Configuration. In the Requester Console there are two tabs available on the main page for accessing two separate service domains: “IT & Corporate Services” and “Quality Management”. Furthermore, there is a functionality available through the My Request tab to view all tickets submitted by front-end users and to access to the manual of the tool via the Help tab. The following figure shows the layout of the main page.

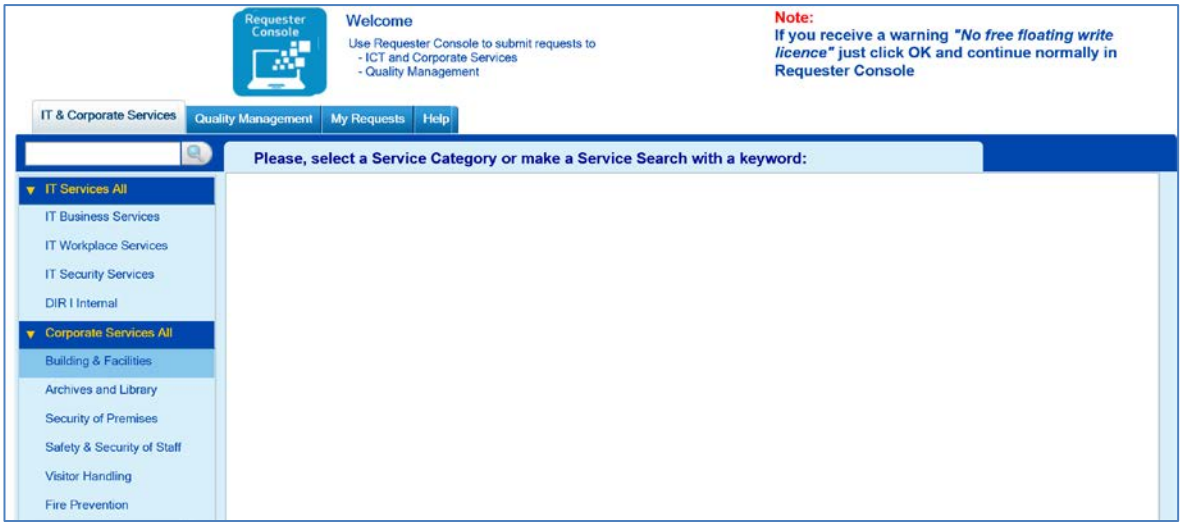


Figure 31. Main page of the Requester Console.

The domain “Quality Management” is out of the scope of this research. Instead, we focus on the “IT & Corporate Services” tab which offers altogether 68 IT services sorted alphabetically and grouped into two main categories:

- IT Services All
- Corporate Services All

The abovementioned main categories are divided into four and five subcategories. When clicking further on these subcategories, they are expanded respectively and the services belonging to the selected subcategory are shown on the right hand panel. The next figure illustrated this for the “IT Business Services” subcategory under “IT Service All”.

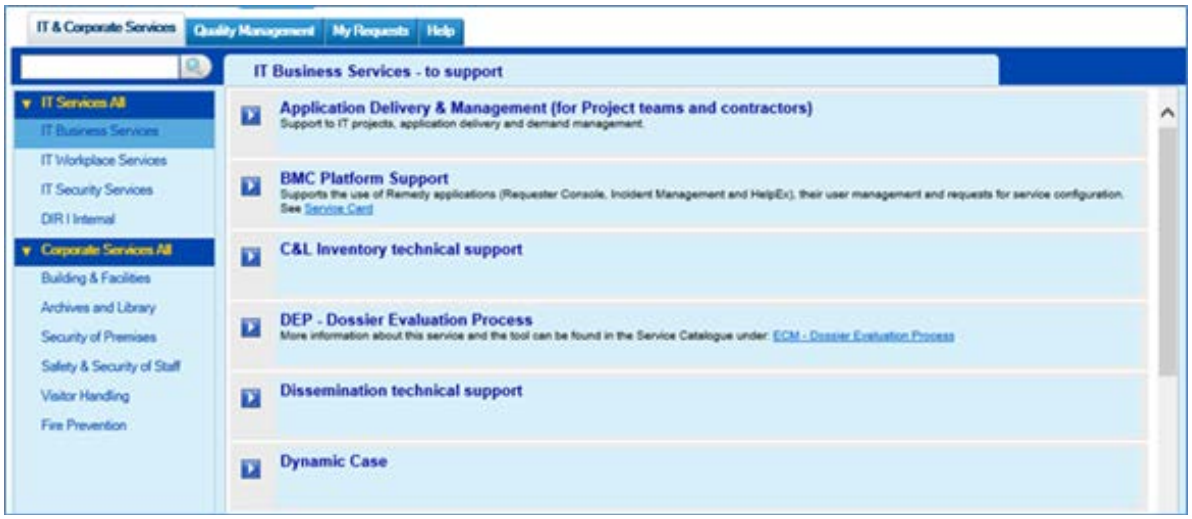


Figure 32. Panel for the IT Business Services subcategory

Furthermore, there is a search functionality available to facilitate the service search based on keywords.

From the services available on the panel, the required service is selected by double clicking the arrow icon. This selection opens either a dropdown menu window or directly a window to add the detailed description of the request. The dropdown menu is shown in the figure below.

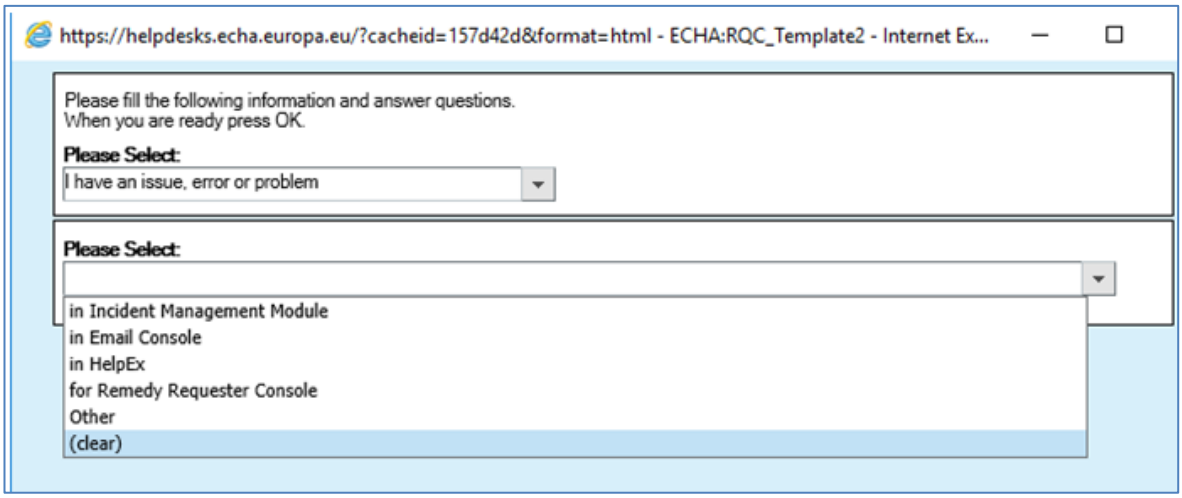


Figure 33. Dropdown menu for the Remedy service

Once the values are selected, the window where the description of the request is given opens. At this point, the submission of the request can be complemented with the following information:

- the name of the person to be requested behalf of (optional)
- the summary information (to update the default value - optional)
- the request details (mandatory)
- the date required (optional)
- the urgency (optional)
- the name of the person to be notified on the submission (optional)
- add the supporting documents (e.g. approvals, screen shots)

The request is completed by clicking the “Create” button or cancelled by clicking the “Back button”. The next figure shows the layout of this window.

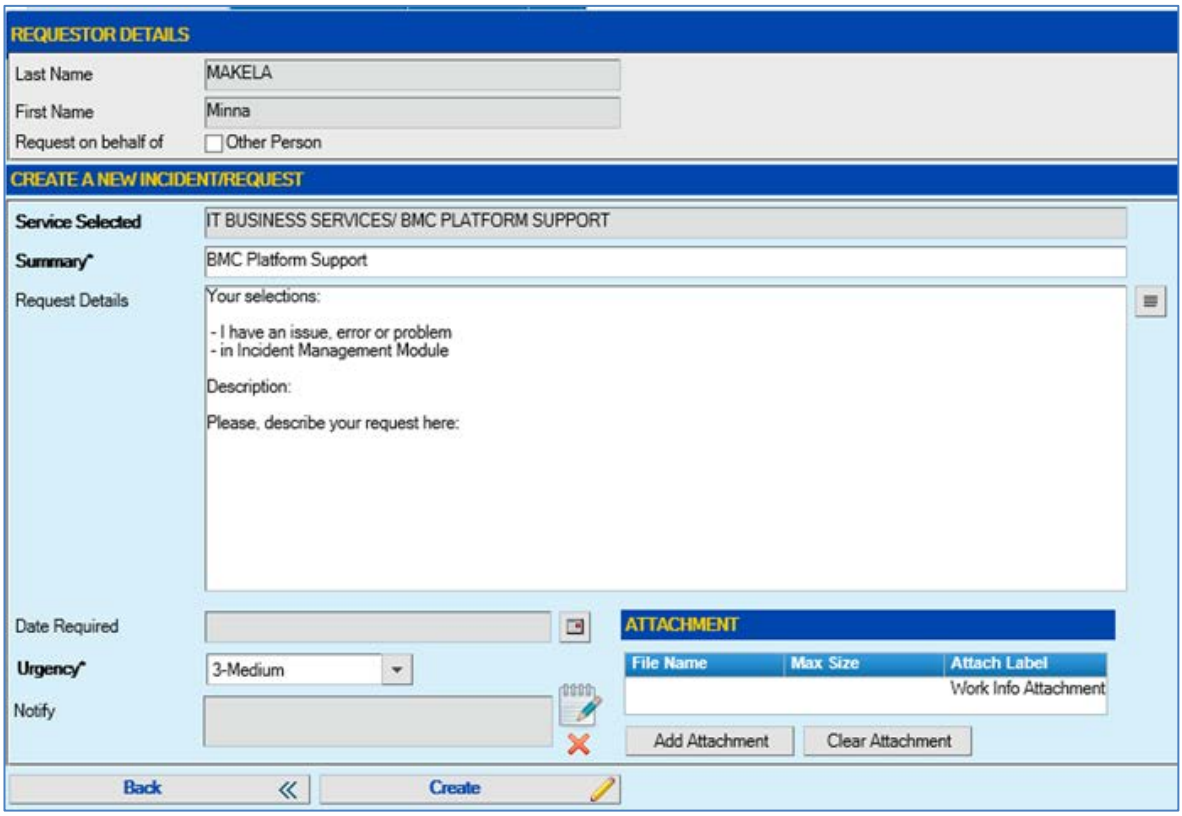


Figure 34. View of the menu to complete the submission of a request

Once the request is submitted, a new ticket is created out of it. The front-end user is able to see this created ticket and its detailed information immediately under the tab “My Request“. This is show in the next figure.

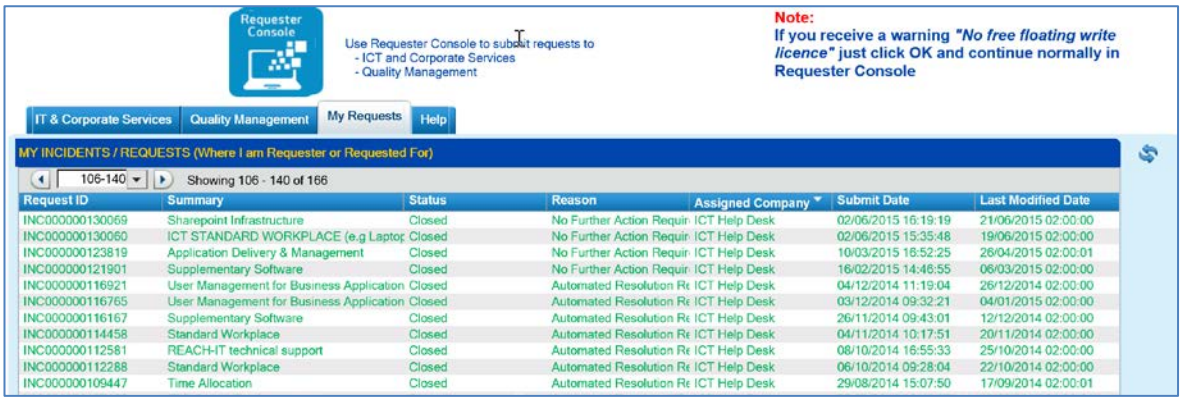


Figure 35. View of the My Request tab

The tab “My Request” shows the overview of created tickets. A front-end user is only able to view the tickets, which he/she created by him/herself and those which were created on behalf of him/her. In order to view the detailed information of the ticket, a front-end end user needs to select and double click the row of a ticket to open a detailed view. In this detailed view, it is possible to submit additional information and to provide a satisfaction rating for the resolved tickets as illustrated in the next figure.

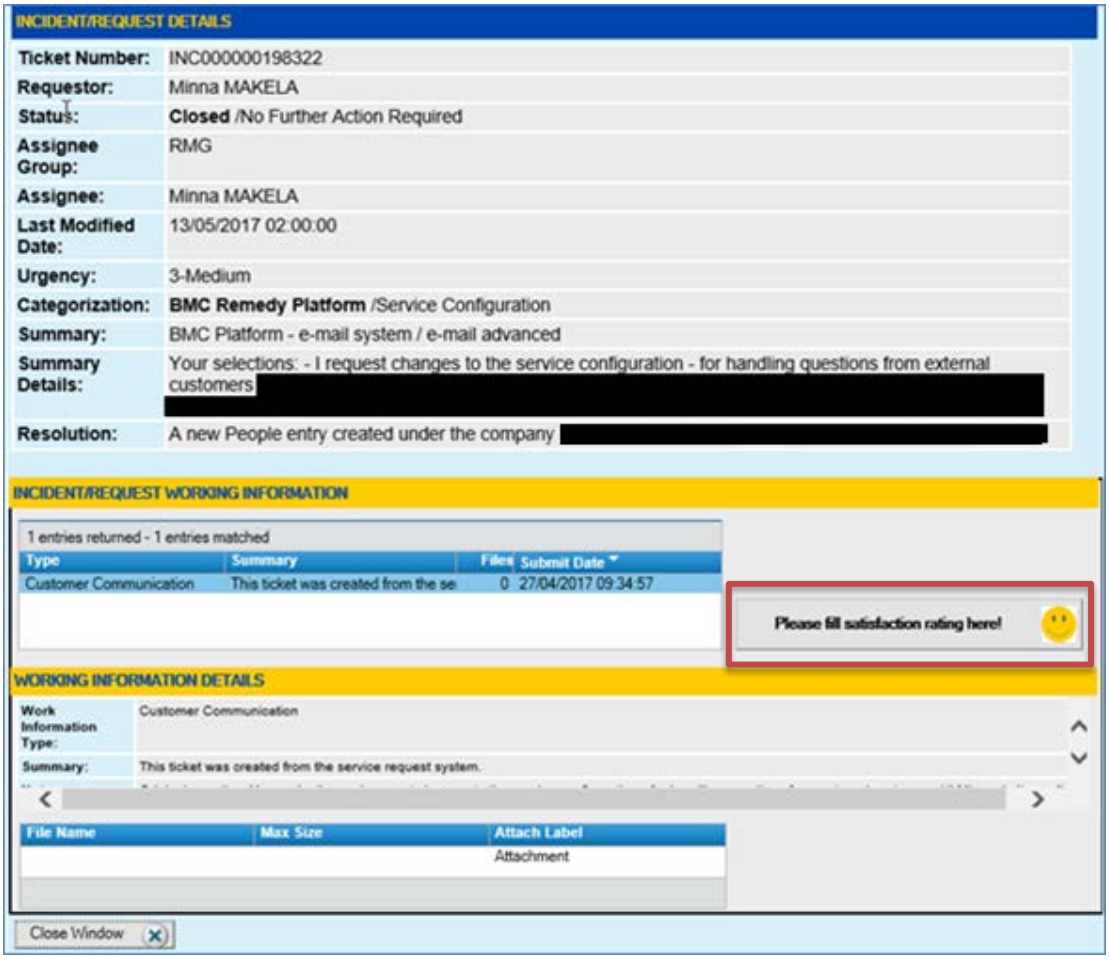


Figure 36. View of the “My Request” tab

7.2. Applying UX Research and Design method

In the previous chapter 3.7 under the theoretical framework, we explained the UX Research and Design method provided by Newman (2017). In this project, we applied its main concepts and principles to improve the Requester Console.

The key principle of the UX Research and Design method is the user centered research which we implemented in our work. At the beginning of the research, we focused on the front-end users of the Requester Console by collecting their feedback about the as-is state. This data was the starting point and the first step in our UX research in order to understand what users were doing, which were their needs and which issues were needed to be solved.

Other important principle of the UX Research and Design method is the iterative way of working in which a process is repeated via several iterations. Each iteration provides an output which is taken as a starting point for the next iteration. The iteration cycle is divided into three phases starting from the assess phase which is followed by the design phase and everything ends in the building phase.

We applied this iterative way of working and included all three phases in our iterations in the following ways:

- In the assess phase, we understood the existing problems and what was necessary to fix.
- In the design phase, we generated possible solutions to solve the problems identified and selected the best solution.
- In the build phase, we built the selected solution.

After doing that, we started the iteration cycle once again from the assess phase by letting front-end users to test the on boarded solution and collected feedback. This way we identified potential problems and if the development was going in the right direction. The design and build phases followed and then the whole cycle started again. Since our project had a limited amount of time and resources we were able to create only three iterations.

Furthermore, during the improvement of the Requester Console, we applied the concepts of Lean and Lean UX by focusing on and listening to users, removing waste in the processes that did not add value to them and followed the light documentation concept with minimal deliverables.

7.3. Development iterations

First iteration (24th of April till 8th till May 2017)

We started the iteration with the assess phase, using the data collected during the interviews and analysed using the methods described previously in order to understand the problems that the front-end-end users had with the use of the Requester Console. Then we prioritized them by using the MoSCoW method. It is important to mention that the whole assessment was always done in combination with the data collected and analysed from the Reference Configuration, since at the end both components needed to be aligned.

The design phase started with a brainstorming having as a target to produce ideas on how to tackle and overcome the problems. The outcome of those actions produced several sketches and after the evaluation we selected the best one for handing over to the build phase.

The best idea introduced a new approach for the interface which can be called a product based navigation which creates relationships between categories, products and services. This way a front-end user would always need to navigate until she/he finds/selects a concrete product which was the reason why she/he wanted to contact the Service Desk.

In practice, this concept offers three different ways to navigate through the interface:

- Via the categorisation - All available products are grouped in common categories that make sense to the front-end end users like for example: account management, business intelligence and reporting, email and calendar or business applications. When a front-end user selects a category, it limits the number of products available in the list of the products. Choosing this navigation option, the front-end user needs first to click on a category and then select the product and the service or vice versa.

- Via the list of products – All the products available (310) are based on the structure created in the product categorization (this is explained in details explained in the development section of the Reference Configuration). This option might be time consuming if the front-end user is not familiarized with the product categorization's

structure. In some cases a product might be mapped to several services and she/he will need to select the service manually. Otherwise if the product is mapped to one service, the service will be automatically filled. In this option, the user first selects the product and then the service if it is not automatically filled in.

- Via the list of the services – All available services are shown in a list od the services. If a front-end user selects one in the list, it will show only the products that are linked to that particular service. In this option, the front-end user first selects the service and then the product.

Once a front-end user has selected a product and a service, the next step is to choose the type of request. The possible options are either an incident (“I have a problem”) or a service request (“I need something”). Once this selection has been done, the end user should give a detailed description of the reason to contact the Service Desk.

Another aspect that was selected was the idea of building a minimise interface in which everything can be done in one page (removing pop-up windows that were used in the as-is state and considered as waste in the view the front-end users). Also, it was targeted to minimise the number of clicks needed to submit a request.

Following the aforementioned principles, all tickets submitted by a front-end user are directly visible in the same window. Only if the front-end user clicks on a specific ticket, it opens a new window to view its full details or to provide additional information or satisfaction rating for the resolved tickets.

The next figure shows the mockup that was produced as an outcome of the building phase. The mockup was formed using the Balsamiq software.

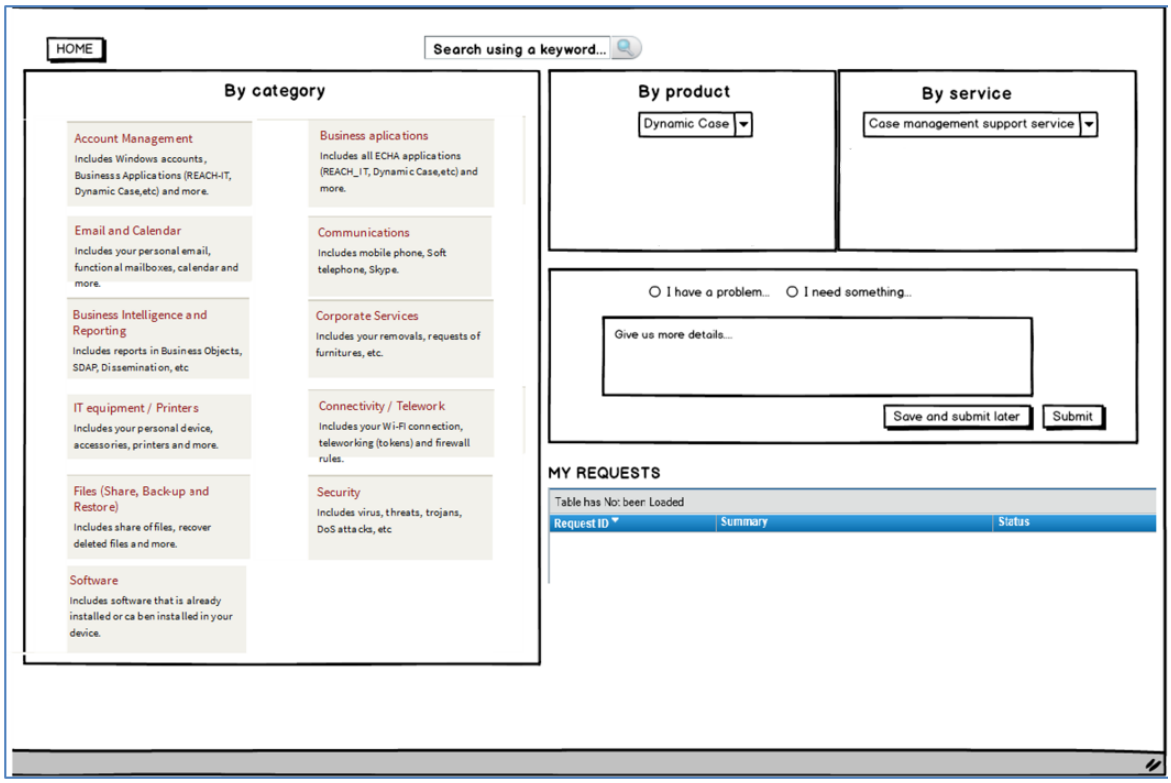


Figure 37. Mockup produced by the first iteration

Second iteration (from 9th till 19th of May 2017)

Once the first build phase ended, it leaded to the next iteration that started with the assessment phase where the produced mock-up was tested by a group of front-end users. We received different feedbacks about the colours, structure, layout, etc. and with the improvements included in the updated mock-up it we moved to the build phase to produce a real prototype in the development environment of the Requester Console.

The main improvements that took place in the second iteration were:

- Creation of a working prototype to have real data and functionalities.
- The name and descriptions of the categories were redefined to be more front-end user friendly
- Icons in the categories were added to improve its visibility and quick identification.
- A counter of the number of products available for a selected category was included.

The prototype developed in the second iteration is shown in the next figure:

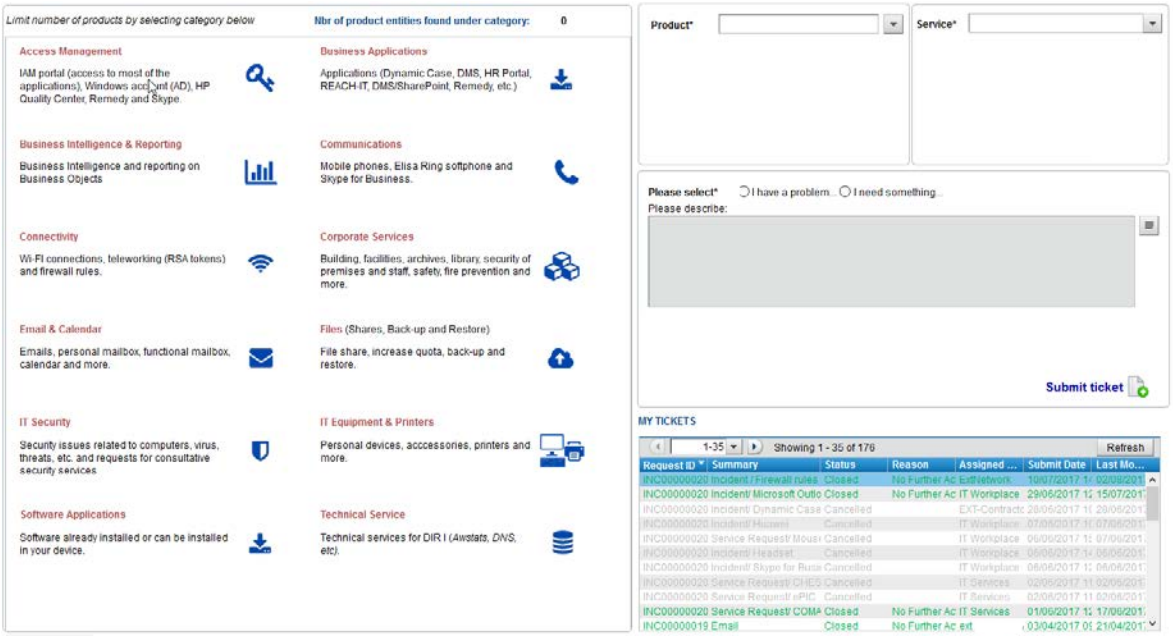


Figure 38. Prototype produced by the second iteration

A prerequisite for the working prototype of the Requester Console was to have the new Reference Configuration in place. Therefore, this second iteration was synchronised with the piloting of the reference configuration.

Third iteration (from 22nd till 30th of May 2017)

The assessment of the third iteration started with several testing sessions of the produced prototype by the second iteration. Already at this stage, it was known that this iteration would be the last one before the go live due to the strict timeline of the project. For this reason, we extended the group of front-end users to test the Requester Console during the last iteration.

The main improvements that took place based on the assessment were:

- The functionality to add attachments was included.
- Two new fields Module and Environment were added.
- The functionality to clear the selections was introduced.
- Guiding messages were included to the selections to guide front-end users in the navigation and selection of a product or service.
- Predefined templates for the incidents (I have a problem) and service requests (I need something) were defined and included.
- Descriptions, pictures and graphic for products and services were included.

The final outcome of the third iteration it is shown in the picture below:

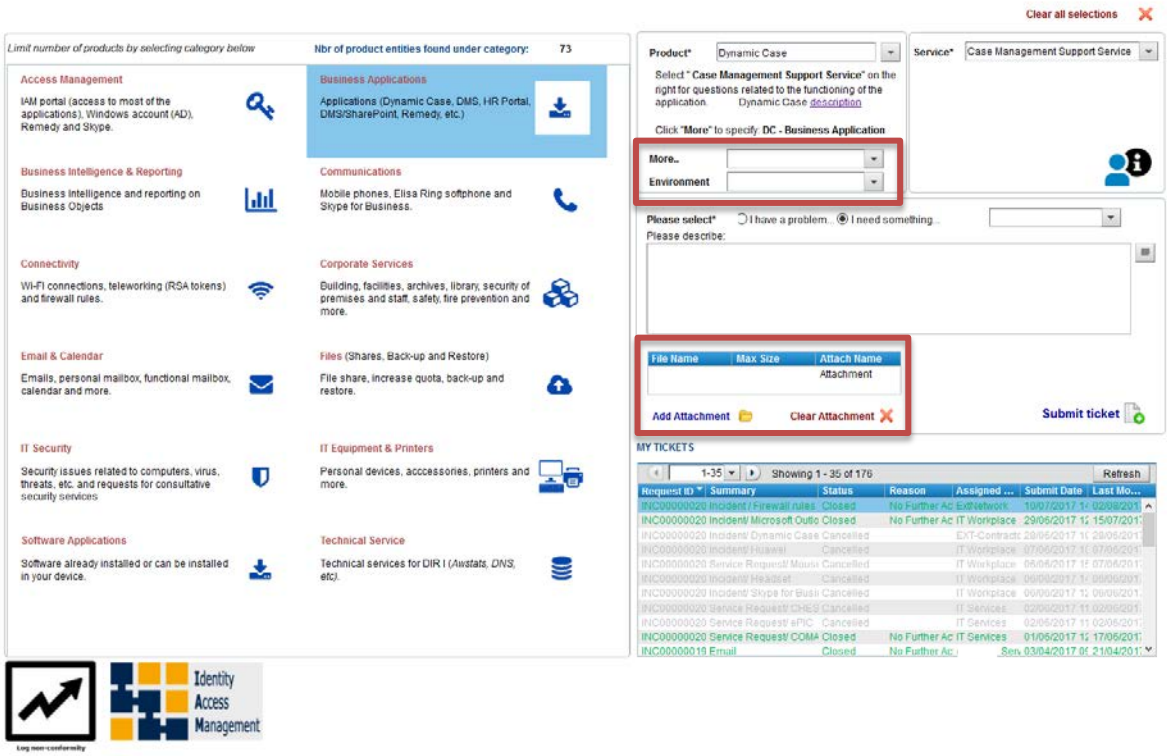


Figure 39. The final prototype produced by the third iteration

The following figure shows an example of the detailed description and graphic added to a "Product" and a "detailed description added to a Service".

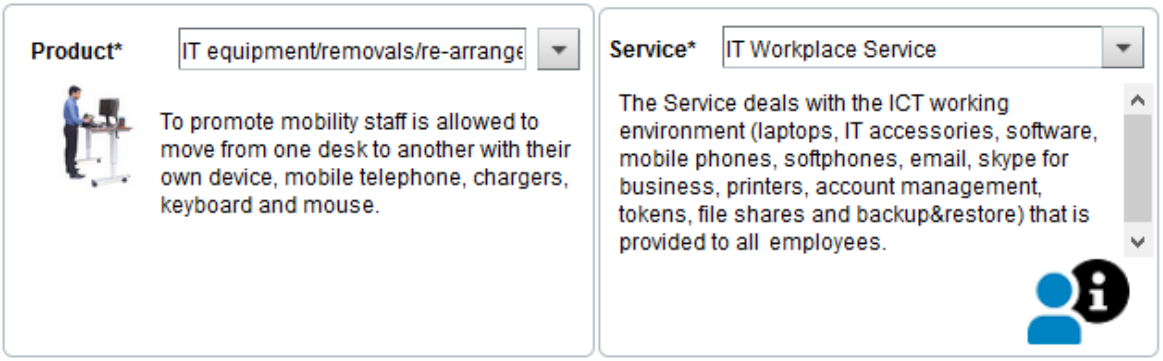


Figure 40. Details of a selected "Product" and "Service"

User Acceptance Test (UAT) and final configuration (from 30th of May till 5th of June 2017)

After the third iteration, we scheduled about one week to User Acceptance Testing, in which the final correction of bugs identified was done and the rest of services and products were added to the configuration including the Reference Configuration.

The results of the UAT were successful and our delivery was on time for the go live deadline.

Go live on the 6th of June 2017

The go live of the new the Requester Console took place on the 6th of June 2017 and went smoothly without any major issues on the technical side. The only issue that rose was related to some specific products that were mapped to several services (most of the services are mapped one to one). This fact created confusion and some front-end users ended up selecting the wrong service, having as a consequence tickets assigned to a wrong supports group. A solution was found immediately to rename this service that created confusion and adding a more detailed description indicating that the service can be used only by Administrators and not front-end users.

8. Development of Reference Configuration

8.1. As-is state

This paragraph documents and describes the main interface of the Ticket Management Module used by the back-end users to handle tickets received from front-end users.

Back-end users are members of one or several “Support groups” and for each “Support group” there is a queue where tickets created (by front-end users) are waiting to be dis-patched.

Once dispatched, they are presented in the Ticket Management Module interface as a list and depending on different criteria (impact, urgency, topic, etc.) attended and resolved.

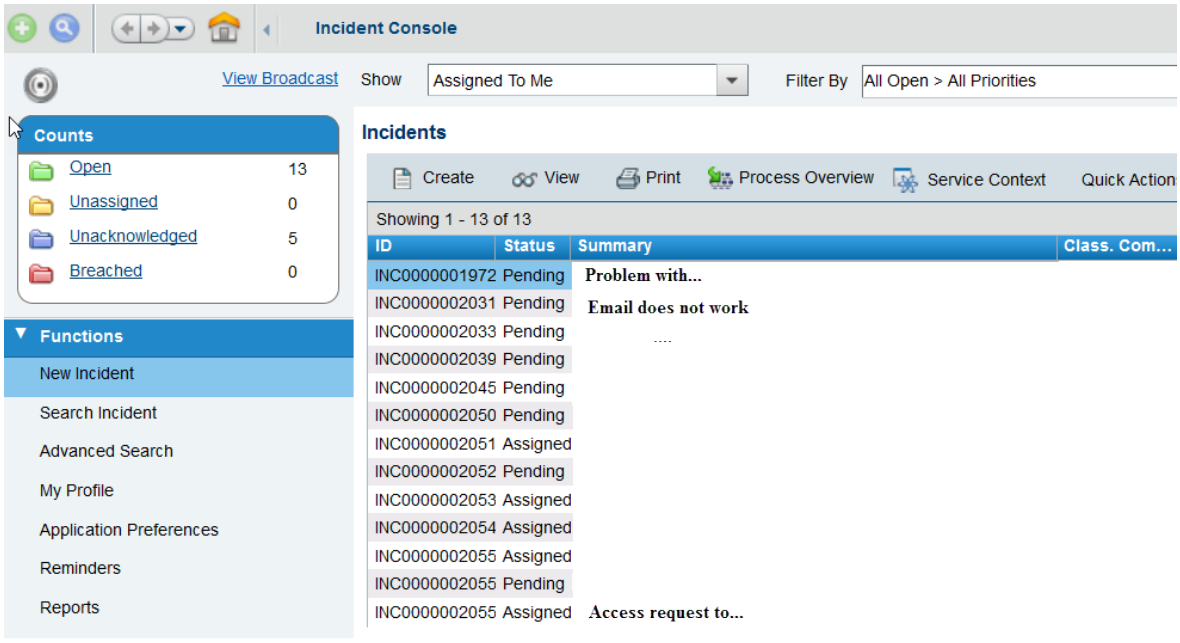


Figure 41. Interface of the Ticket Management Module

When opening a ticket more detailed information is shown as indicated in the figure below:

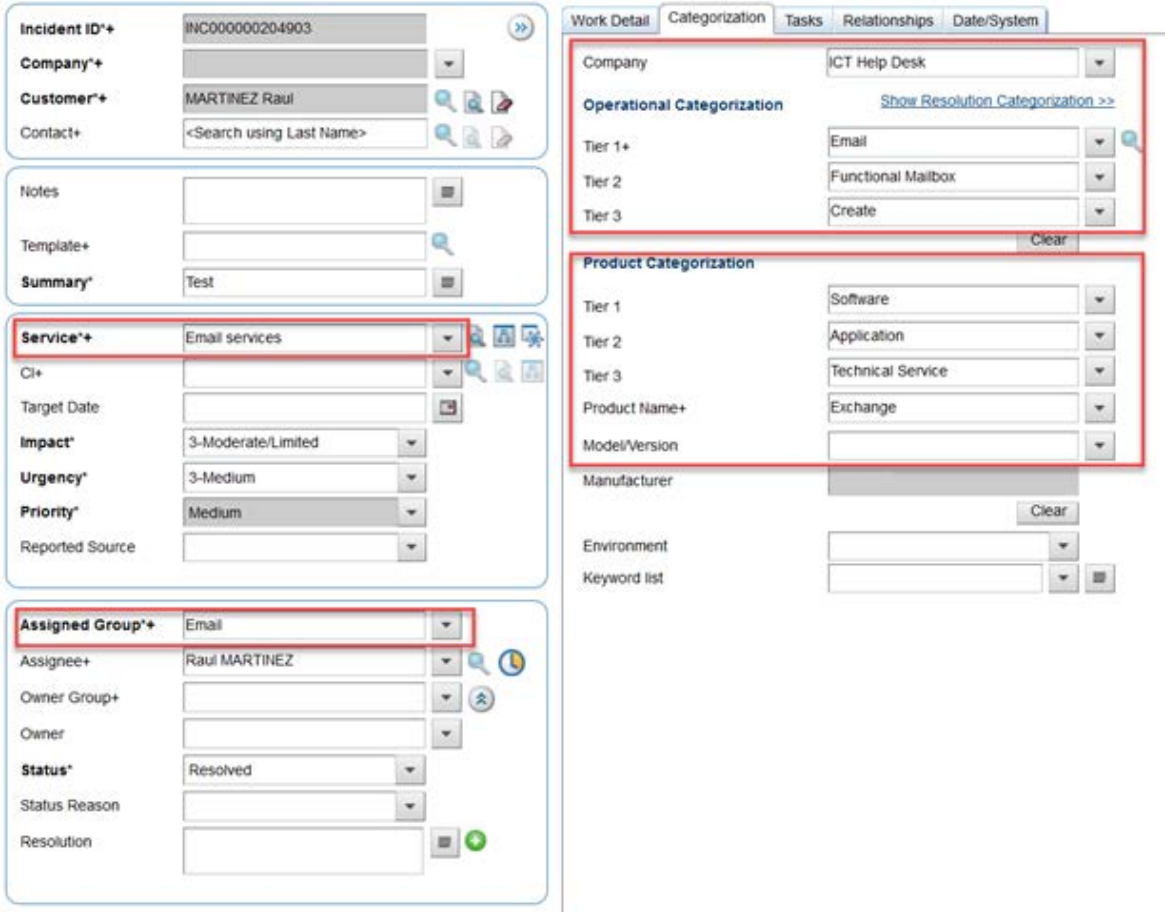


Figure 42. Detailed view of a ticket from the back-end user perspective

The elements that are relevant for the scope of the research are the ones in red squares “Service”, “Assigned Group”, “Operational categorization” and “Product Categorization” and used as the main pillar to build the Reference Configuration. The Reference Configuration is the kernel that provides values to both the Ticket Management Module and the Remedy Requester Console.

8.2. Applying ITIL and categorisation of service request and incidents

In the previous chapter 3.2 under the theoretical framework, we explained the main concepts of ITIL and a method to categorize tickets for service requests and incidents.

Our organization was already using a customized version of ITIL in which the main concepts of service strategy, design, transition and operations were familiar and properly documented in an Integrated Quality Management System (IQMS).

During the whole re-structuration of the IT services other frameworks related to IT architecture (TOGAF), governance (COBIT) and project management (PM2) were used, however, they are out of the scope of this research.

Concerning the classification of tickets into “Incidents” and “Service requests” we followed the approach suggested in the theoretical section (since ITIL does not have a clear guidance on how to classify tickets).

The conceptual structure for used for the classification and applied to the “Operational Categorization Tier 1, 2 and 3” was the following:

- TYPE (process)
- CATEGORY (action)
- SUB-CATEGORY (special)

For the other fields “Service”, “Support Groups”, “Product Cat. Tier 1, 2, 3” and “Product name” we used a bottom top approach and affinity diagram to create the structure.

8.3. Development stages

The new Reference Configuration was built in different stages as indicated below and explained more into detail in the next sections:

- Restructuration of existing “Services” and creation of new “Services”.

Figure 43. Restructuration of Service’s diagram

- Inventory of all existing “Products” and creation of a new structure.

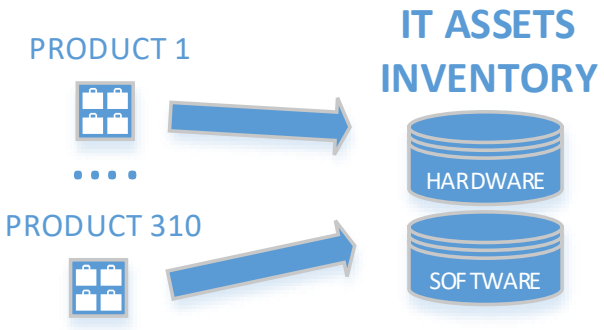


Figure 44. Inventory of Product’s diagram

- Linking of “Products” to the new “Services”.

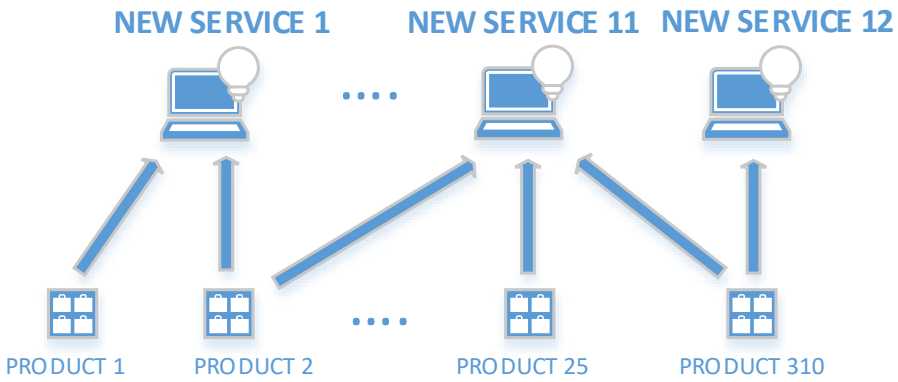


Figure 45. Liking of Products and Services diagram

- Creation of new “Support groups”.

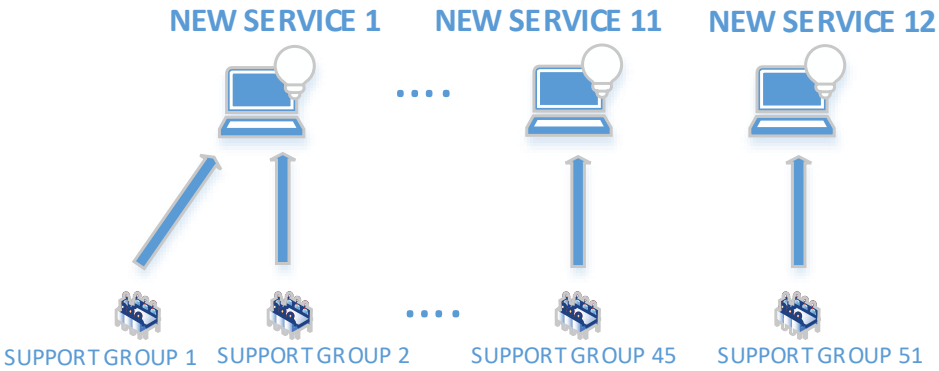


Figure 46.Restructuration of Support group’s diagram

- Creation of a unique “Operation categorization” for all services.

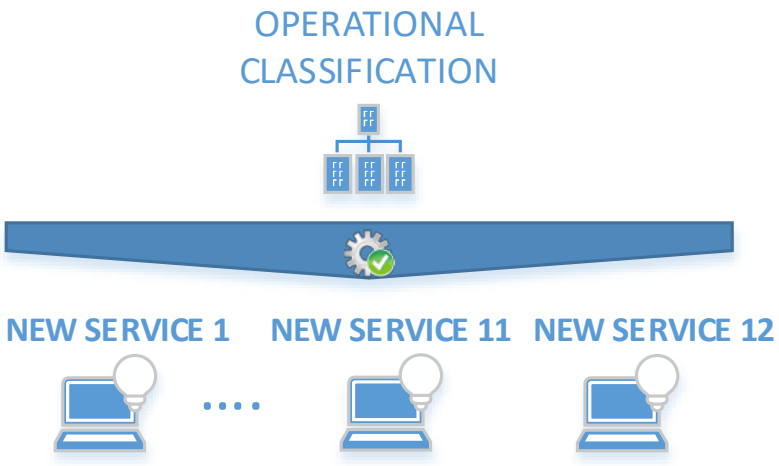


Figure 47. Creation of unique operational categorization’s diagram

Services

The restructuration of the services was the starting point to build the Reference Configuration and the reduction in the number of services was one of the main objectives.

Most of the discussions happened at the IT Service Catalogue level (which is out of the scope of this research) however we participated actively providing input for the discussions with the preliminary conclusions obtained from our research.

During the focus groups sessions our mission was to provide guidance and harmonization across the services making sure that the focus of the re-structuration was on the front-end user satisfaction and not just another internal re-structuration thinking from an IT perspective only.

A total of 68 services were grouped and merged into 12 new services (11 IT services + 1 corporate service).

Products

In the revamp of the product list it was necessary to take into account certain technical limitations in the Reference Configuration and the fact that we do not have a Configuration Management Database (CMDB) module installed in our Ticketing Management System.

The limitation was related to the fact that for each “Product name” added, it is mandatory to have a value in “Product Cat. Level 1, 2 and 3” meaning that all products needed to belong to three different groups.

Considering this limitation an indication that helped us to create the structure was the Pareto analysis which gave us an estimation in the proportion and the number of labels most used per product tier as indicated in the graphic below:

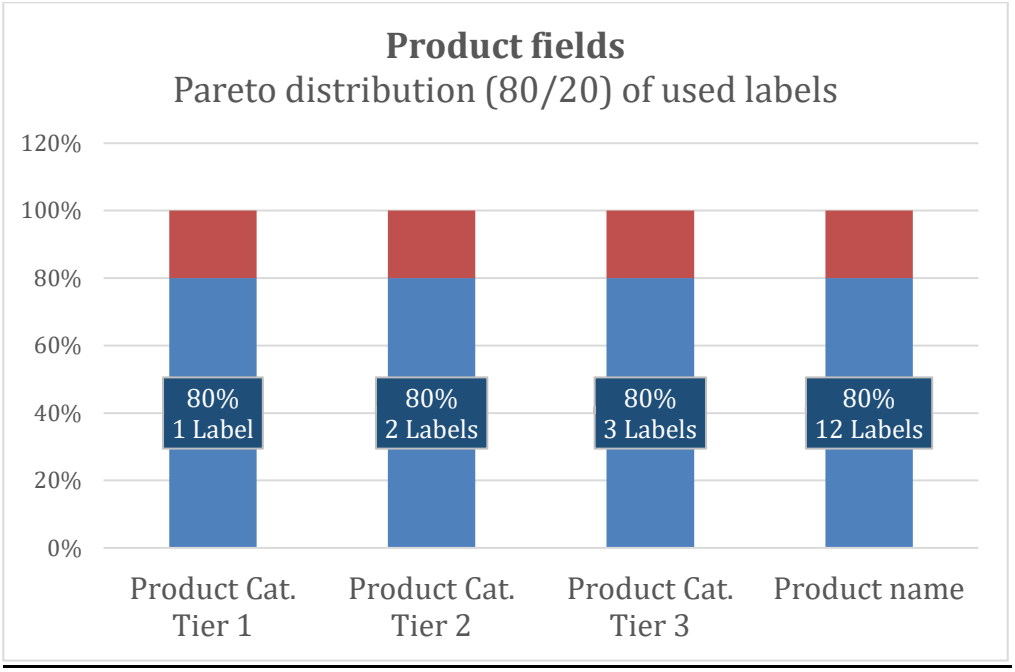


Figure 48. Pareto distribution of labels used for the Product fields

Keeping in mind the previous aspects the next step was to identify how many “Products“ were still valid, remove the ones that become obsolete and to include new ones.

To create the full inventory we scheduled individual meetings with all Service Managers in order to create a centralized “IT assets” inventory accessible to everyone. Other stakeholders were also involved in the inventory process such as business units, finance units, etc.

A total of 310 products were tracked and using a bottom top approach and an affinity diagram the following concept layers were defined to create the structure in the Reference Configuration:

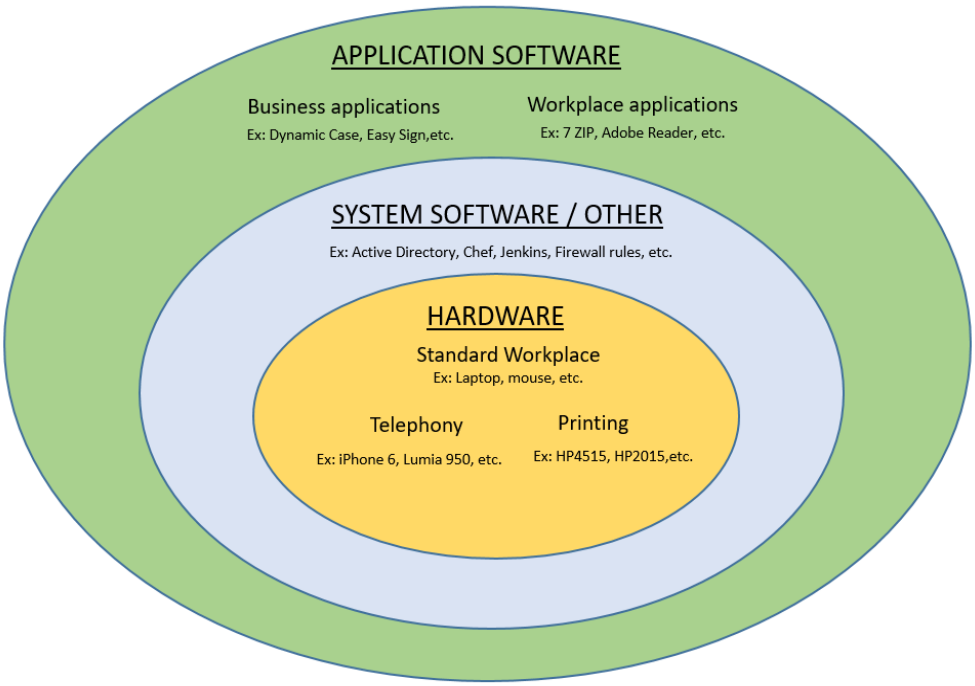


Figure 49. Conceptual structure of the IT inventory

There is a kernel layer in which all “Hardware” components were grouped and decomposed into three main categories:

- Standard Workplace
- Telephony
- Printing

In top of the “Hardware” layer there is “Software” decomposed also into three main categories:

- Business applications
- Workplace applications
- System software

On top of both layers there is a super root level covering both, all “Hardware” and “Software” components named “IT assets”.

The final implementation into the real configuration looked as indicated in the table below divided per “Product Categorization Tier 1, 2, 3” and “Product Name”:

Prod. Cat. Tier 1	Prod. Cat. Tier 2	Prod. Cat. Tier 3	Product Name (Random Examples)
IT ASSETS	Hardware	Standard Workplace	Laptop External Screen…
		Telephony	iPhone Samsung…
		Printing	MP4343 MP3234
	Software	Business Applications	CHESAR REACH-IT…
		Workplace Applications	Microsoft Outlook Chrome…
		Systems / Other	Active Directory Jenkins…

A harmonization in the naming convention was applied across all products using always the full description of the product at the beginning of the label in capital letters and if acronyms needed placed at the end of the label in brackets (for example “Document Management System (DMS)”.

Once we had the full inventory of “Products” completed, we initiated the mapping or link between “Products” and “Services”. For most of the “Products” the relation was one to one meaning that one “Product” was supported by one “Service” however there were several “Products” that it was necessary to be mapped to several “Services”.

Support Groups

To create the new “Support groups” it was necessary to understand all the processes and workflows for each of the new “Services”. To do so we created a mapping of how tickets were flowing between different escalation levels. In our organization there are three escalation levels in which the first one provides general support and acts as the main contact point for front-end users. The second level provides more specialized support and resolve those tickets that cannot be resolved at first level. The third level provides even more specialized support resolving more complex issues than the second level.

It the picture below there is a simplified example of the “Support Groups” and workflows of the “IT Workplace” service:

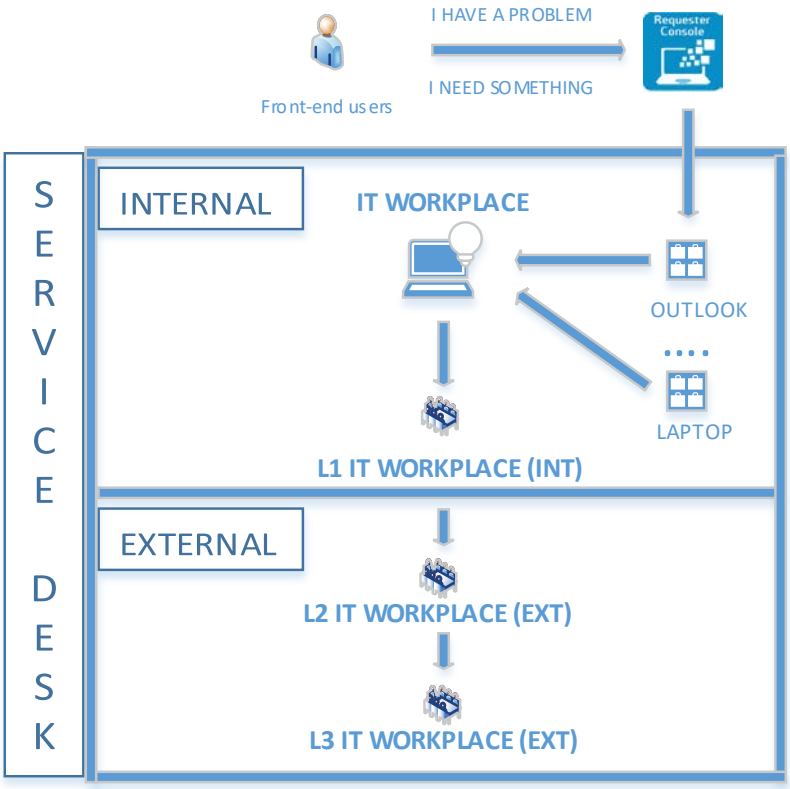


Figure 50. Workflow of a ticket through different support levels

Once we had a clear understanding of the different workflows we started to build up the support groups in coordination with the Service Managers. Each of the support groups created was classified into three different levels according to ITIL best practices. The format we chose to name the support groups was the following. At the beginning of each support group name, the escalation level should be indicated using an abbreviation of the word “Level” in capital letters and level of escalation 1, 2 or 3 indicated (abbreviations did not cause any confusion since the escalation concept was already mature in the organization). The name of the support group should be in the middle separated by a dash - and at the end it should indicated if the support group was internal (staff belonging to organisation) or external (staff not belonging to the organisation).

The reason to split “Support groups” into escalation levels and internal and external is due to the need of extracting the resolution times employed by each of the support groups and to verify that SLAs (Service Level Agreements) and OLAs (Operational Level Agreements) are fulfilled.

An example of the “Support groups finally implemented in the Reference Configuration can be seen fin the table below:

Service	Support group name	Members (Random example)
IT WORKPLACE	L1 – IT Workplace (Int)	Raul Kimmo Eerika Marjo
	L2 – IT Workplace (Ext)	Daniel Gabor Ivan Minna
	L3 – IT Workplace (Ext)	Jyrki Nadia

The same structure was applied uniformly for each of the 12 “Services” until creating a total of 51 “Support groups”.

Operational Categorization

The operational categorisation was perhaps the most challenging to revamp since we found resistance and it took many hours of negotiations to change the mind-sets of certain Services Managers, trying to convince them to give up the specific and move into more general abstract concepts.

The strategy used was small wins meaning that the more Service Managers were convinced to accept the new classification the easier it would be to convince the next ones. We started with those that did not offer resistance, learnt from that experience and improve the initial draft classification with their feedback. The main arguments used to convince the ones that offered resistance was showing the results obtained in our research, explaining that a change in the classification would be a “win-win” situation obtaining mutual benefits for both sides. It would improve the quality of the reports created, more tickets would be classified, more labels used, etc.

At the end we achieved a compromise having the first two fields “Op. Categorization Tier 1 and 2” fixed, meaning they cannot be changed without a general consensus and the last tier “Op. Categorization Tier 3” would be more flexible, possible to customize it as long as the change is relevant, not redundant and has a real added value.
The final classification implemented in the reference configuration and divided per “Op. Categorization Tier 1, 2 and 3” is indicated in the table below:

Op. Cat. Tier 1	Op. Cat. Tier 2	Op. Cat. Tier 3
INCIDENT	Configuration	
	Failure	• Hardware Issue • OS-Level Issue • Software Issue • Unavailability (service/components) • Other
	Performance	• Capacity Issues • Interruptions • Slowness • Other
	Security	• Attack • Breach • Compromised Credentials • Compromised Data • Other
	Other	
SERVICE REQUEST	Account Management	• Legal Entity Update • Modify • New User • Password / Pint Reset • Remove • Role / Permissions • Other
	Change Standard (Pre-approved)	• New Item / Installation • Removal / Uninstallation • Quota Increase • Configuration Change • Loan Item • Lost / Stolen • Other
	Training / Guidance / Information	
	Other	
CHANGE	Change Emergency (eCAB)	• Account Management • Code Improvement • Configuration Change • Data Management • Resource Management • Other
	Change Normal (CAB)	• Account Management • Code Improvement • Configuration Change • Data Management • Migration Change • Resource Management • Other

In the table below it can be seen final results of the Reference Configuration. It shows the number of labels used in the old and in the new reference configuration as the percentage of simplification that was achieved for all the fields relevant:

Field name	Number of Labels Old system	Number of Labels New system	% Reduction
Service	68	12	**-82 %**
Support Group	110	51	**-54 %**
Product Cat. Level 1	6	1	**-83 %**
Product Cat. Level 2	110	2	**-98 %**
Product Cat. Level 3	49	6	**-88 %**
Product name	130	310	**+ 138 %**
Operational Cat. Level 1	63	3	**-95%**
Operational Cat. Level 2	101	11	**-89%**
Operational Cat. Level 3	84	35	**-58%**

For all the fields there was a reduction in the number of labels with the exception of the "Product name". This is due to the fact that new products were added and most of them moved into the "Product name" field.

Piloting phase

For the reference configuration we did not use iterations per se, but a "piloting services" in which we tested with a small group of back-end users that helped us to collect feedback and fine tune the configuration until the scheduled go live.

Pilot of 2 Services (from 1st till 15th of May)

We selected two services ("IT Workplace" and "Management Information Systems") to pilot the new Reference Configuration for two weeks. The reason why we chose those services first was because they were the ones that received more tickets in the in the year 2016 and had the widest scope. In this way they would be able to test the new Reference Configuration at its most, being able to use almost all the new labels created.

The experience during the whole piloting exercise was positive, mainly because during the focus groups many back-end users participated in the brainstorming sessions and they felt that the new Reference Configuration was logical and applicable.

Pilot of 2 + 2 Services (from 15th till 29th of May)

Two more services “Application Delivery Service Management” and “Business & Reporting” were added to the pilot of the new Reference Configuration.

During all the piloting phase, we received feedback that helped us to fine tune the Reference Configuration, re-phrasing some labels that created confusion and adding some additional ones to cover new scenarios. In addition, we ran several reports to monitor the usage of the classification, tickets classified versus not classified, etc. to really check if the new reference was being really used.

Approximately one week (29th May till 6th June) was kept as a buffer between the piloting phase and the go live in case major changes, correction of bugs, etc. needed to be done. This week was used specially to remind all the back-end users about the coming release and to train them on how to use the new Reference Configuration.

Go live 6th of June

The go live of the new reference configuration took place on the 6th of June 2017 at the same time as the Requester Console and it went smoothly due to the fact that the services with the highest number of products associated and with the higher volume of tickets expected were included in the pilot phase.

9. Validation of the developments

During the development phase of the Requester Console, we collected continuous feed-back from front-end users through iterations and one month after the go live of the re-vamped Requester Console and Reference Configuration we performed the validation of our project. This step was needed to check the outcomes met the requirements originally defined.

Requester Console

The validation test for the Requester Console was done on the 6th of July (one month after the go live). In total 40 people were interviewed during the validation phase, 20 women and 20 men.

For the validation data, we created only quantitative closed questions for the interviews since we wanted to measure the level user satisfaction.

The following questions were asked in comparison with the previous “Requester Console”:

1. ***Do you like more the current layout (look and feel)? Yes / No***
2. ***Is the new structure more clear and intuitive to use? Yes / No***
3. ***In a scale of 1 to 5 (1 really difficult 5 - really easy) How easy or difficult is to create a ticket using the new Requester Console?***

The results obtained from the three questions are analysed as follows:

Do you like more the current layout (look and feel)?

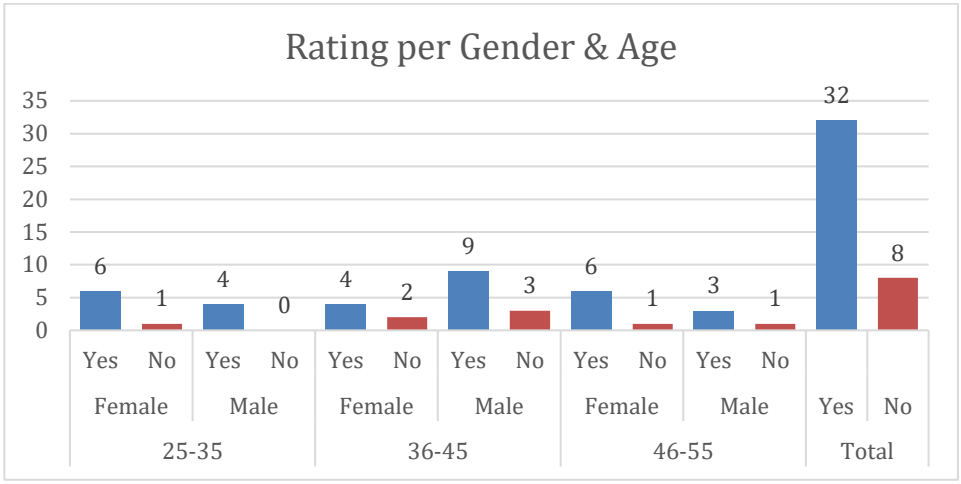


Figure 51. Bar chart representing the validation results obtained for question 1

Most of the people interviewed (32 out of 40) were satisfied with the new layout of the new Requester Console. The satisfaction among both genders was equal.

Is the new structure more clear and intuitive to use?

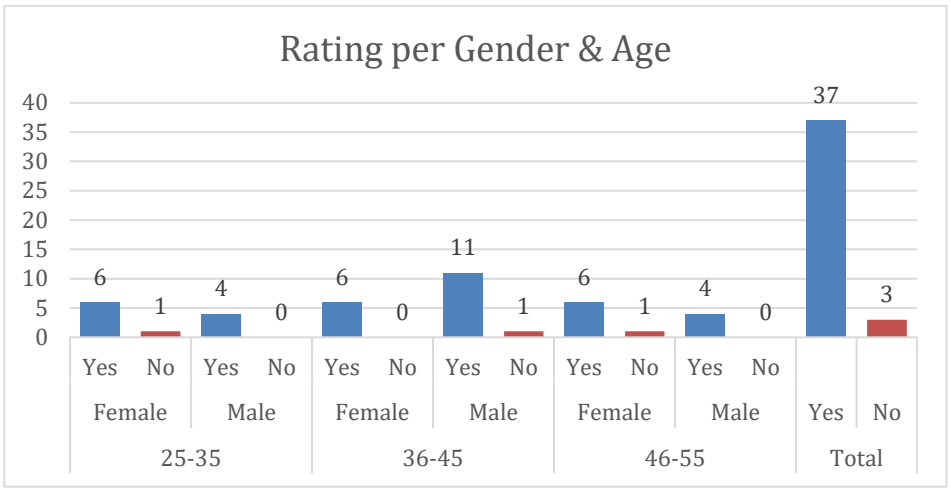


Figure 52. Bar chart representing the validation results obtained for question 2

Most of the people interviewed (37 out of 40) were satisfied with the structure of the new Requester Console. The satisfaction per gender was slightly higher for males.

In a scale of 1 to 5 (1 really difficult 5 - really easy) How easy or difficult is to create a ticket using the new Requester Console?

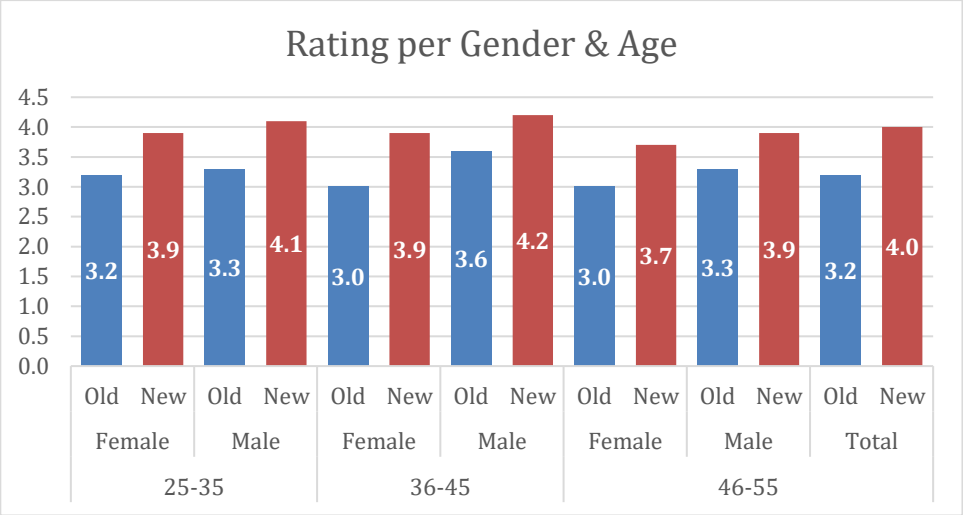


Figure 53. Bar chart representing the validation results obtained for question 3

Comparing the results obtained in the survey done for the old Requester Console against the new one, we could conclude that there was an increase in the user satisfaction of 0.76 points having a score of 4 points out of a total of 5. In general, men gave a higher score than women.

Reference Configuration

The validation test for the new Reference Configuration was done for the month of May 2017 in which a total of 478 tickets “Resolved / Closed”.

In the next table it is shown the number of tickets classified versus not classified and the percentage that were classified:

Field name	Number of tickets classified	Number of tickets not classified	% classified
Service	478	0	**100 %**
Support Group	478	0	**100 %**
Product Cat. Level 1	472	6	**99%**
Product Cat. Level 2	472	6	**99%**
Product Cat. Level 3	471	7	**98%**
Product name	470	8	**98%**
Operational Cat. Level 1	478	0	**100%**
Operational Cat. Level 2	357	121	**75%**
Operational Cat. Level 3	278	200	**58%**

For the whole validation exercise it is important to take into account the limitations in the comparison, since the total amount of tickets “Resolved/Closed” for the old Reference Configuration was 17878 covering a period of 12 months and for the new Reference Configuration it was 478 covering a period of less than one month (25 days).

Since we could not wait one year to obtain the same amount of tickets “Resolved /Closed” for the new Reference Configuration, from this point and on, we will consider the results obtained in May 2017 as extrapolated to a period of 12 months and having the same amount of tickets “Resolved/Closed” as the old reference configuration 17878.

If we compare the percentage of tickets classified in the old Reference Configuration versus the new one, the results obtained were as indicated in the next graphic:

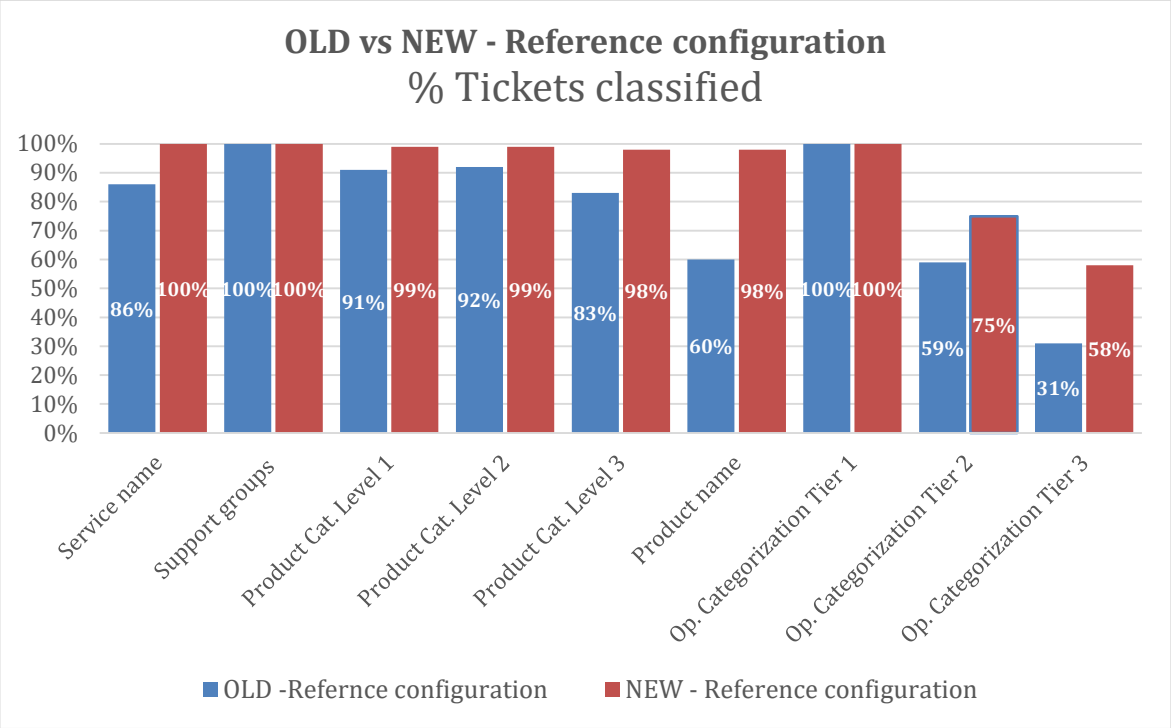


Figure 54. Bar chart comparing the percentages of tickets classified per field in the old and the new Reference Configuration

There were two fields “Op. Categorization Tier 1” and “Support groups” that did not vary since they were both already at the maximum level 100%.

On the other hand, for the rest of fields there was a clear improvement having all of them a positive increase when comparing the old and the new Reference Configuration. The

field “Product name” would be the one with the highest increase + 38% of tickets classified, followed by “Op. Categorization Tier 3” with an increase of + 27% and both “Op. Categorization Tier 2”, “Product Cat. Level 3” with an increase of + 16%.

In the graphic below it is represented the difference between the percentages of tickets classified in old and the new Reference Configuration:

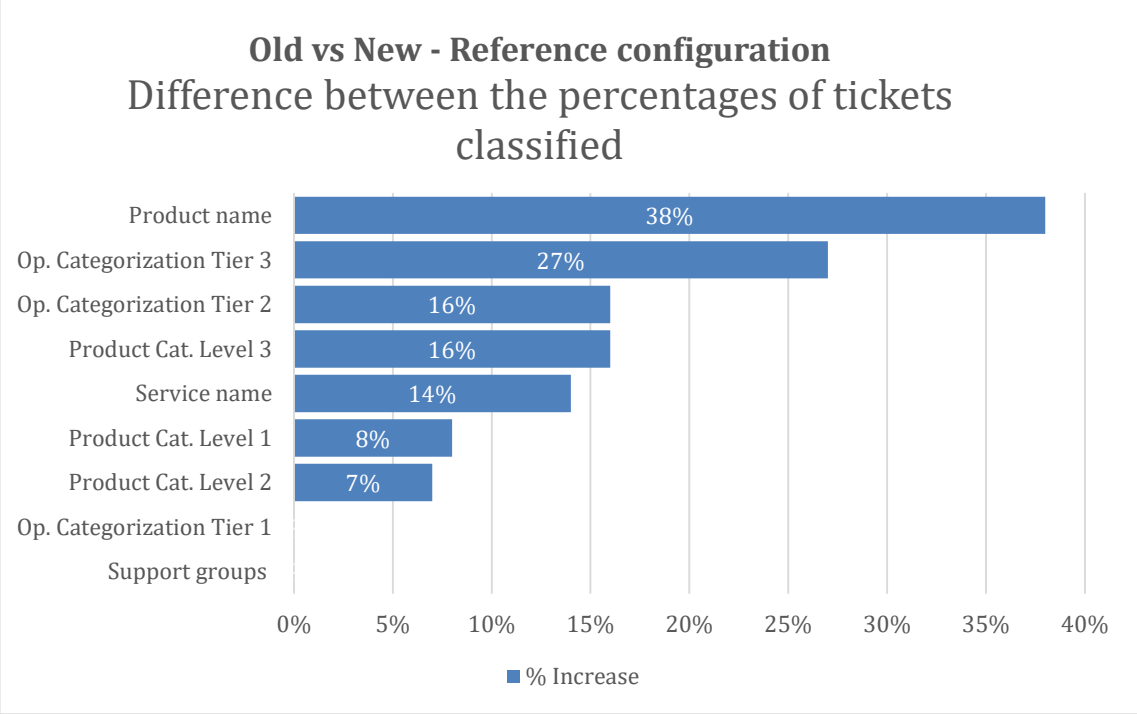


Figure 55. Bar chart representing the difference between the percentages of tickets classified per field in the old and the new Reference Configuration

Concerning the number of labels available versus labels used the results obtained from the validation were the following:

Field name	Number of labels available	Number of labels used	% of used labels
Service	12	12	**100 %**
Support Group	51	37	**100 %**
Product Cat. Level 1	1	1	**100%**
Product Cat. Level 2	2	2	**100%**
Product Cat. Level 3	6	6	**100%**
Product name	310	45	**15%**
Operational Cat. Level 1	3	3	**100%**
Operational Cat. Level 2	11	10	**91%**
Operational Cat. Level 3	31	20	**65%**

If we compare the percentage of used labels in the old Reference Configuration versus the new one the results obtained were as indicated in the next graphic:

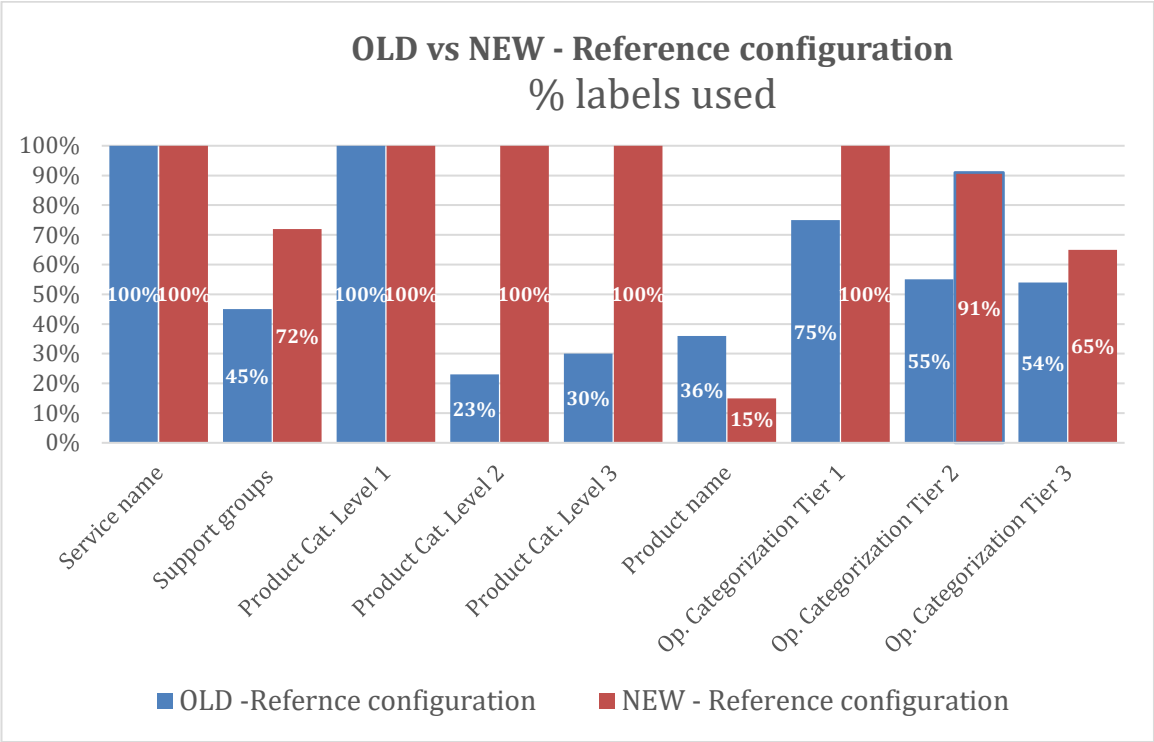


Figure 56. Bar chart comparing the percentages of labels used in the old and the new Reference Configuration

There were two fields “Service name” and “Product Cat. Level 1” that did not vary since they were both already at the maximum level 100%.

There was one field “Product name” that decreased -11%. This is due to the fact that in the validation results only one month was extracted and it was not time enough to receive tickets for the full scope of products available.

For the rest of fields there was a high improvement having all of them a positive increase when comparing the old and the new Reference Configuration. The field “Product Cat. Level 2” would be the one with the highest increase + 77% of labels used, followed by “Product Cat. Level 3” with an increase of + 70%. This increase that puts both labels to the maximum level 100% it is due to the simplification in the tree structure of the products available, a reduction in the number of labels available and a better mapping to the requester console. For the same reason “Op. Categorization Tier 1” increased a + 25% reaching to the maximum level of 100%.

The field “Support groups” had an increase of + 27% due to the reduction in the number of support groups available.

The fields “Op. Categorization Tier 2” had an increase of + 36% and the field “Op. Categorization Tier 3” an increase of + 11% also due to the reduction in the number of labels and the creation of more abstract concepts.

In the graphic below it is represented the difference between the percentages of labels used in old and the new reference configuration:

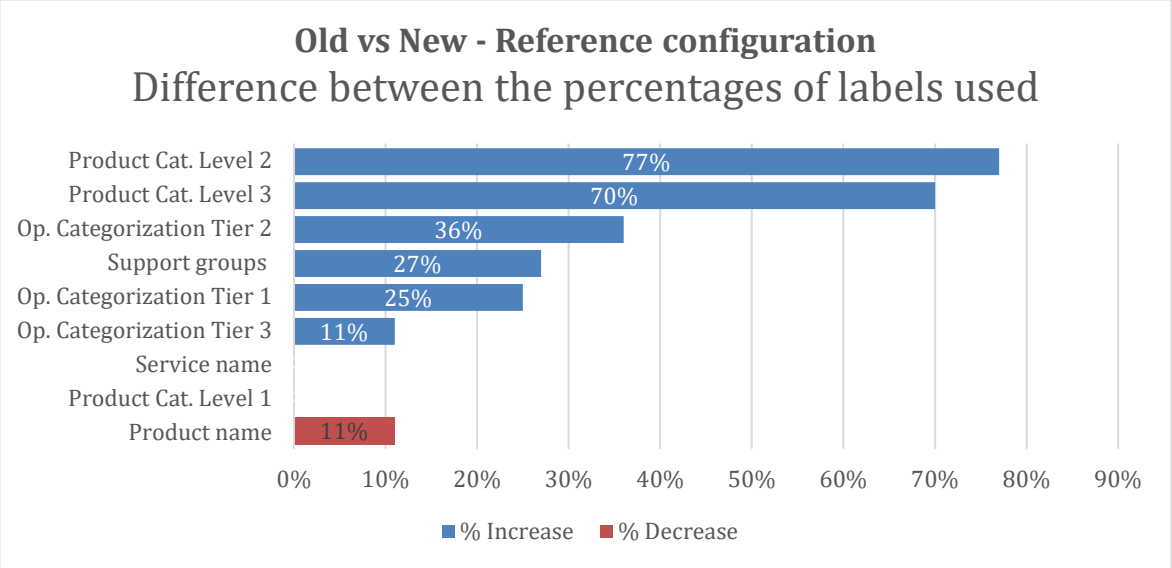


Figure 57. Bar chart representing the difference between the percentages of labels used per field in the old and the new Reference Configuration

10. Proposal for the roadmap

In this paragraph, we present our proposals for further developments of the Requester Console and the Reference Configuration after this project following the continuous improvement approach.

10.1. Requester Console

The suggestions for the Requester Console are based on the analysis of the data collected during this project which we could not fitted in the scope of this project due to the prioritisation (they belong to the “Could” group as indicated in the analysis by the MoSCoW method) and some feedback received after the go live. These influence positively on user experiences and therefore should be considered for implementing.

Most of the improvements of the Requester Console can be implemented in the short term already in 2017. We propose to divided them to two different cycles and follow the same iterative approach as we used in this project rather than waiting for a big bang deployment. In this way, it is possible to improve the satisfaction of the front-end users in the continuous way.

For the next year, we propose a study to examine different alternatives for machine learning in the IT Service management area especially related to the tools we are using. This proposal rose from the literature review of the artificial intelligence in IT service management done for this project. Based on this review, artificial intelligence and related machine learning are becoming more important and interesting since they provide huge opportunities in this area. Therefore, we firstly propose to perform a study project to examine possible options suitable for our target organisation and select the best solution out of them. As a next step, a pilot project should be organised to validate the selected solution for the final decision.

10.2. Reference Configuration

For the Reference Configuration, we propose data quality reviews to be performed at the end of each quarter. The results of these reviews either provide evidences for further improvements or validates the current values. If improvements are proposed, they are col-

lected and implemented at the beginning of 2018. This is due to the reporting requirements that the main updates of the Reference Configuration should be always synchronised with the reporting periods which is usually a calendar year in our organisation.

The aforementioned proposals are shown in the roadmap table below.

	2017		2018	
	Q3	**Q4**	**Q1**	**Q2**
Requester console	• Workflow to merge products via one ticket submission • Add new fields: Summary and Urgency • Use of key words for search	• Favourite services list • Templates per service • Add new functionality: news feeds on service statuses • Automatic refresh of my tickets	• Study to examine alternatives for Machine learning	• Pilot of Machine learning
Reference Configuration	• Review of data quality and propose improvements	• Review of data quality and propose improvements	• Update according to the reviews if improvement proposals done • Review of data quality and propose improvements	• Review of data quality and propose improvements

11. Conclusions

The purpose of this chapter is to conclude the outcome of this project by assessing how well the objectives were met and providing a summary of the answers to the research questions.

Objectives

We can state all the objectives initially defined were achieved successfully.

The Requester Console and Reference Configuration were improved and deployed meeting the most important requirements that were gathered during data collection.

The improved Requester Console was validated through a user satisfaction survey in which the overall score was increased.

The Reference Configuration was standardised and simplified. According to the validation results, it is now used more consistently than the previous one having better data quality and facilitating better reporting.

A feasible roadmap was created to facilitate change and release management of the Requester Console and Reference Configuration.

Research questions

Two research questions were initially defined and the following answers were found during this project.

1. ***What kind of Requester Console should our target organisation have to increase user satisfaction?***

   Our target organization should have a Requester Console with the following characteristics in terms of:

   - Content – It should be written in a user friendly language instead of IT jargon with concise descriptions.
   - Structure – All contents should be grouped into categories that make sense for the front-end users and detached from the structure used at the IT Service

Portfolio. Alternative ways of finding information should be offered (searching using categories, products or services directly, with a search box, etc.)

- Functionalities - It should maximise the use of the main page in which a whole request can be sent using the same window, minimising the number of clicks and increasing the simplicity to create a ticket.
- Workflow - It should automatically generate several tickets in specific order for the services where this is needed.
- Layout - It should include visual elements that help the navigation with a limited amount of colour combination promoting accessibility.

2. What kind of Reference Configuration should our target organisation have to improve data quality?

Our target organization should have a Reference Configuration with the following characteristics:

- A harmonised configuration created with the consensus and collaboration of back-end users so that they are confident and engaged to properly classify tickets.
- A simplified configuration that maximizes the use of the same general concepts across all IT services.
- An aligned configuration with the Requester Console so that most of the tickets are correctly classified in a transparent way by front-end users.
- A change in the mind-sets of back-end users and evangelization of ITIL concepts, having as a consequence a harmonised use of the ticket classification.

12. Discussion

In the discussion chapter we would like to expose some of our thoughts and reflections after the development of the project grouped into the following topics:

Legacy systems

Unfortunately we did not have the possibility to investigate or use any other alternative ticketing management system than BMC Remedy. This was due to the context and nature of our target organisation (around 700 employees) in which the migration to a new system would be very costly in terms of budget, human resources and time due to the complexity of its current structure. BMC Remedy acts as a hub for other third parties application (web forms, SAP BO reporting, etc.) which creates many dependencies and any simple change can have an impact in multiple applications making it very inflexible.

Communication

We discovered that it existed a lack of communication between the different IT services, making us realize that the problems were not only related to an IT tool but to a human factor. In addition back-end users were overloaded with multiple tasks showing resistance to use the Ticking Management Module since it requires a lot of clicks (meaning time) to record or classify a ticket. It required a lot of evangelization and still (since this is only the beginning) promoting the benefits of classifying properly tickets and how the reports created could help them to improve the services.

Though our organisation follows ITIL for IT Services Management and applies the concepts of request fulfilment and incident management, not all the IT staff shared the same understanding. Several times during our project, we had to re-explain basic definitions of those ITIL processes in order to have a proper discussion. This just confirms how difficult ITIL is at the end and it still requires lot of evangelizations in our organisation.

Furthermore, it was challenging to identify the right target audience from the IT services and engage them during the different project phases. We started the engagement from the top level and went through to the lowest level which taught us how important it is to speak in the same level as your target audience. A top management does not understand the details but just need the high level information to be engaged whereas with specialists is needed to go through each detail level information.

While our project progressed, also the restructuring of the IT services went on and new Service Managers were appointed. This brought a new challenge for our project since suddenly we had new key players who started to manage IT services without any previous contribution or detailed knowledge of our project. We had to engage these new persons just before the go live which was time-consuming.

Governance

It should exist a clear governance applied from the top management layers till the bottom operational layers.

For example, the fact of having now a clear policy and a centralized model for the acceptance in the creation of new labels in the Reference Configuration it will avoid the creation of redundancies and to repeat the same mistakes that leaded to a huge and inefficient Reference Configuration.

Another aspect is the ownership of the service data that should be nominated to Service Managers. This means to check the content of the tickets regularly to see if the Reference Configuration is use consistently, and also, to gain the understanding what is ongoing in their services. In this way, they can ensure the quality of their service reports and take the continuous improvement actions based on the reposts

Automation

With the integration of machine learning algorithms it will be possible to create a dynamic classification of tickets by learning on the existing data that is already correctly classified.

13. Reference

Bell, S.C., Orzen M.A., 2011. Lean IT Enabling and Sustaining Your Lean Transformation. Portland, United States: CRC Press.

BMC Software, 2016, Best Practice Insights – Focus On: ITIL Service Strategy For ITIL 2011. Available at: <http://www.bmc.com/guides/itil-service-strategy.html> [Accessed 03 April 2017].

Cabinet Office, 2011a. ITIL Continual Service Improvement. Edition 11. TSO.

Cabinet Office, 2011b. ITIL Service Operation. Edition 11. TSO.

Chatbots, n.d.Virtual agent. Available at: <https://www.chatbots.org/virtual_agent/> [Accessed 02 April 2017].

Eames Official Site, 2017. Design Q & A Text, Available at: <http://www.eamesoffice.com/the-work/design-q-a-text/> [Accessed 09 April 2017].

Gartner, 2015. Gartner Reveals Top Predictions for IT Organizations and Users for 2016 and Beyond. Available at: <http://www.gartner.com/newsroom/id/3143718> [Accessed 05 August 2017].

Gartner, 2017. Gartner IT Infrastructure, Operations & Data Center Summit 2017. Available at: <http://www.gartner.com/binaries/content/assets/events/keywords/data-center/dc12a/gartner_iodc_summit_overview.pdf> [Accessed 05 August 2017].

Giardi. A., 2016. Importance of user-centered design: “iTunesU Siena” experience. International Journal of Engineering Technology and Scientific Innovation Volume:01, Issue:02. Available at: <http://www.ijetsi.org/uploads/ijetsi_01__19.pdf> [Accessed 09 April 2017].

Gothelf J., Seiden J., 2016. Lean UX: Applying Lean Principles to Improve User Experienc,e Lean UX 2nd edition. Available at: <http://pdf.th7.cn/down/files/1312/lean_ux.pd f> [Accessed 1 March 2017].

Hall L.n.d. BrightTalk Channel – Available at: <https://business.brighttalk.com/products/channel/ > [Accessed 07 May 2017].

Hassenzahl, M., & Tractinsky, N. (2006). User Experience - a research agenda [Editorial]. Behavior & Information Technology,

ITIL Service Management, 2007. All About Incident Classification. 29.05.2007. Available at: <http://itservicemngmt.blogspot.fi/2007/05/all-about-incident-classification.html> [Accessed 19 April 2017].

Jäntti, M., Cater-Steel, A., Shrestha, A., 2012. Towards an Improved IT Service Desk System and Processes: A Case Study. International Journal on Advances in Systems and Measurements. Available at: <https://eprints.usq.edu.au/22735/3/Jantti_Cater-Steel_Shrestha_PV.pdf > [Accessed 17 April 2017].

Leclercq P., 2016. PM2, PM C4, Managing Requirements & Stakeholders. Edition 2.5. European Commission.

Lee S.M., Olson D.L., Lee S-H., Hwang T., Shin M.S, 2007. 'Entrepreneurial applications of the lean approach to service industries'. The Service Industries Journal. Available at: <http://www.researchgate.net/profile/David_Olson3/publication/231174854_Entrepreneurial_applications_of_the_lean_approach_to_service_industries/links/0fcfd50632fe24100b000000.pdf> [Accessed 17 April 2017].

Marcin T., 2014. The History of User Experience Design. Medium. Available at: <https://medium.com/@marcintreder/the-history-of-user-experience-design-5d87d1f81f5a> [Accessed 09 April 2017].

Marquis H., 2010. How to Classify Incidents. DITY TM DO-IT-Yourself Guides, itSM Solutions. Available at: <http://www.itsmsolutions.com/newsletters/DITYvol6iss27.htm> [Accessed 23 April 2017].

Newman M., W., 2017. UX501x Introduction to User Experience [online via internal VLE] EdX. Available at: <https://courses.edx.org/> [Accessed 03 February 2017].

Nielsen J., 2012 Usability 101: Introduction to Usability. Nielsen Norman Group. Available at:<https://www.nngroup.com/articles/usability-101-introduction-to-usability/> [Accessed 18 March 2017].

Norman, D. A. 2013. Design of Everyday Things: Revised and Expanded. New Yor, Basic Books. London: MIT Press (UK edition).

Norman D., Nielsen J., n.d. The definition of User Experience (UX). Norman Group. Available at: <https://www.nngroup.com/articles/definition-user-experience> [Accessed 18 March 2017].

QRCA, n.d. Leading the Conversation in Qualitative. Qualitative Research Consultants Association. Available at: <http://www.qrca.org/?page=whatisqualresearch> [Accessed 1 July 2017].

Penn State University Libraries, 2017. Empirical Research in Education and the Behavioral/Social Sciences. Available at: <http://guides.libraries.psu.edu/emp> [Accessed 1 July 2017].

Point Guard, n.d. Archive for the 'ITIL Metrics for Business Process Optimization' Category, Available at: <http://www.pointguardsolutions.com/blog/?cat=1> [Accessed 5 March 2017].

Rouse M., n.d.a. Pareto diagram. Techtarget Network, WhatIs.com. Available at: <http://whatis.techtarget.com/definition/Pareto-principle> [Accessed 25 June 2017].

Rouse M., n.d.b. Fishbone diagram. Techtarget Network, WhatIs.com. Available at: <http://whatis.techtarget.com/definition/fishbone-diagram> [Accessed 25 June 2017].

ServiceNow, 2017, IT webinars, Available at: <https://www.servicenow.com/events.html > [Accessed 07 May 2017].

Shrivastava S., 2012. Womack and Jones' 5 principles of lean. Growth Machine a blog for project management practitioner, [blog] 16 June. Available at: <http://saurabh-growthmachine.blogspot.fi/2012/06/womack-and-jones-5-principles-of-lean.html> [Accessed 25 June 2017].

University of Jyväskylä, 2010. Koppa, Strategies. Available at: <https://koppa.jyu.fi/avoimet/hum/menetelmapolkuja/en/methodmap/strategies> [Accessed 1 July 2017].

USC Libraries, 2017. Organizing Your Social Sciences Research Paper: Quantitative Methods. Available at: <http://libguides.usc.edu/writingguide/quantitative> [Accessed 04 July 2017].

Waters K., 2009. Prioritization using MoSCoW. 101 Way Blog, [blog] 12 January. Available at: <http://www.allaboutagile.com/prioritization-using-moscow/> [Accessed 22 March 2017].

Wikipedia, n.d. the free encyclopedia. Affinity diagram. Available at: <https://en.wikipedia.org/wiki/Affinity_diagram> [Accessed 22 June 2017].

Writing@CSU, 2017. The Writing Studio. Available at: <https://writing.colostate.edu/guides/page.cfm?pageid=1285&guideid=60> [Accessed 7 July 2017].

The Free Dictionary, n.d. Available at: < http://www.thefreedictionary.com/categorisation > [Accessed 01 May 2017].