POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Enhancing User Experience and Usability in Enterprise Software: A User-Centered Design Approach

Tesi di Laurea Magistrale in
Computer Science and Engineering - Ingegneria Informatica

Author: **Salvatore Fadda**

Student ID: 944786
Advisor: Franca Garzotto
Co-advisors:
Academic Year: 2022-2023

Abstract

In recent years, correlated with the rise in remote work, there has been a significant surge in the use of corporate software technologies. These technologies are the software tools users employ to perform their work. Their diversity and complexity are undeniably extensive.

The term 'Enterprise User Experience' pertains to the user experience designed for enterprise software. In this master's thesis, grounded in a redesign project of the enterprise financial software by Quantyx, I explore the application of the User-Centered Design (UCD) approach to such systems.

Firstly, I introduce a Systematic Literature Review to comprehend the prevailing state of the art concerning using the User-Centered Design approach in enterprise software. Based on its findings, a UCD process is subsequently proposed for the tangible enhancement of user experience in enterprise software applications.

The process outlined is applied to real-world software, QuantyxRM, which serves as a case study. This software is redesigned using user-centric studies. Specifically, a new Information Architecture and User Interfaces are established. This transformation is facilitated by a tool that allows for the comprehensive representation of both components.

The results are subsequently tested, and an answer to the primary research question of the thesis is provided, confirming the applicability of the UCD approach to enterprise software and to what extent.

Keywords: User-Centered Design, Enterprise Software, User Experience, Enterprise User Experience, Software Redesign, Information Architecture, Card Sorting, Three Testing

Abstract in lingua italiana

Negli ultimi anni, in correlazione con l'aumento del lavoro a distanza, si è verificato un aumento significativo nell'uso delle tecnologie software aziendali. Queste tecnologie sono gli strumenti software che gli utenti utilizzano per svolgere il proprio lavoro. La loro diversità e complessità sono innegabilmente ampie.

Il termine "Esperienza utente aziendale" si riferisce all'esperienza utente progettata per il software aziendale. In questa tesi di master, basata su un progetto di riprogettazione del software finanziario aziendale di Quantyx, esploro l'applicazione dell'approccio User-Centered Design (UCD) a tali sistemi.

In primo luogo, presenterò una revisione sistematica della letteratura per comprendere lo stato dell'arte prevalente riguardo all'utilizzo dell'approccio User-Centered Design nel software aziendale. Sulla base dei risultati ottenuti viene successivamente proposto un processo UCD per il miglioramento tangibile dell'esperienza dell'utente nelle applicazioni software aziendali.

Il processo delineato viene applicato al software del mondo reale, QuantyxRM, che funge da caso di studio. Questo software è stato riprogettato utilizzando studi incentrati sull'utente. Nello specifico, vengono stabilite una nuova architettura dell'informazione e interfacce utente. Questa trasformazione è facilitata da uno strumento che consente la rappresentazione completa di entrambe le componenti.

I risultati vengono successivamente testati e viene fornita una risposta alla domanda di ricerca principale della tesi, confermando l'applicabilità dell'approccio UCD al software aziendale e in che misura.

Parole chiave: Progettazione centrata sull'utente, Software aziendale, Esperienza utente, Esperienza utente aziendale, Riprogettazione del software, Architettura dell'informazione, Card Sorting, Three Testing

Contents

Introduction

The User Experience (UX) is gaining significant importance in various industries; this is confirmed by the advancement of the level of UX maturity, an aspect that remains largely unaffected by the geographical location or the scale of the companies involved [20].

Company and Project

I was hired as a User Experience Engineer by Quantyx Advisors (Quantyx), an employee-owned firm established and managed by experienced risk management professionals. Quantyx operates within the financial sector, specifically focusing on alternative investments, which include real estate, private equity, private debt, venture capital, infrastructure, credit lending, and funds of funds. The company delivers risk management and asset valuation services.

Quantyx has developed an in-house software called QuantyxRM (QRM), which specializes in data management and automated calculations. Company employees use this software internally to upload, save, and view data used for the analyses conducted by consultants and to automate these processes.

The company's objective is to create a cloud-based Software as a Service (SaaS) solution that employees can conveniently use and sell to external companies to do this type of analysis internally. The envisioned software would be fast, easy to understand, and capable of automating the primary analyses required in alternative investment. This would allow users to save significant time in completing their work.

The product can be defined as Enterprise Software, i.e. software used daily by users for business purposes [64]. In Chapter 1, I define what this implies in terms of analysis complexity and user experience.

Quantyx development team

Unlike the entire Quantyx financial unit, which includes just over 80 employees, the business unit responsible for developing QuantyxRM software currently has seventeen members. I am the sole design and user experience team member.

Thus, regarding its operational dynamics, this business unit is equatable to a Start-Up. Given that, the techniques employed for the AS-IS software analysis are tailored for a Start-Up context [4].

The company software, QuantyxRM

As mentioned previously, despite the development team being in a situation comparable to a Start-Up, the user base and the amount of information to be managed by the QuantyxRM software are vast. QRM is a well-established and long-standing cloud-based Enterprise Software with a high content complexity designed to meet specific and well-defined business needs.

QRM was initially released in 2017 as a support tool for managing the large volume of daily financial data the company utilizes. As the company grew, investments towards the software increased, leading to its gradual evolution into a more complex system. What started as a simple data management software transformed into a solution encompassing data analysis and automated financial calculations.

Project

This thesis is based on a project started in mid-February 2023 with Quantyx.

The project's initial objective was to develop a Design System. With my advisor, we began evaluating the QuantyxRM software to identify its primary challenges. What emerged was that the main issue with the software was not the absence of a design system but a general lack of the software user experience. Therefore we decided to undertake a redesign process to enhance the user experience.

We started restructuring the software with continuous guidance from the Product Owner. Based on my supervisor's advice, I studied how to employ the UCD approach to redesign Enterprise Software. In Chapter 3, I introduce a variant approach called User-Centered Enterprise UX Design, which will be employed as the execution process for redesigning

QuantyxRM, which will be used as a case study for this thesis.

Research Topic and Scope

This thesis focuses on understanding how to enhance user experience and usability in enterprise software through methodologies grounded in User-Centered Design (UCD) principles.

The two concepts of User-Centered Design, which places the user and their needs at the centre, and Enterprise Software, which emerges to address business requirements, are compared, and the first objective of this thesis is to understand how these two themes are interconnected [22, 64].

Research Questions

The research conducted for my master's thesis addresses three specific research questions.

Research Question 1: *What is the current state of the art regarding the UCD approach in the design of enterprise software?*

The primary objective of the first research question is to determine whether the UCD approach has been employed in the design of Enterprise Software within academic research, and if so, in which instances and with what outcomes.

Research Question 2: *Which user-centred process should be employed to redesign an Enterprise Software for enhancing the user experience?*

Research question two identifies a user-centred process for improving enterprise software's user experience.
I want to determine the practical steps for redesigning the Enterprise Software QuantyxRM.

Research Question 3: *If applying user-centred processes is feasible in real business software, does it bring real benefits? If yes, to what extent?*

The third research question goal is to understand the thesis outcomes, specifically whether

using user-centric methodologies leads to tangible benefits in the design of Enterprise Software.
To address this query, tests were developed comparing results from a standard design based on business criteria with those derived from a user-centric approach.

Thesis Structure

My master's thesis is structured into eight chapters.
The first chapter covers the introduction, which has already been presented.

Chapter 1 delves into the background, where I expose the principal topics the thesis addresses.

Chapter 2 presents methods used to obtain the preliminary findings and results. In this chapter, the methods are theoretically defined, and a brief description of how they are implemented within the thesis is provided.

Chapter 3 outlines the methodology employed for the research and presents the findings related to RQ1 and RQ2. The UCD process used to carry out the case study is presented in this chapter.

Chapter 4 presents the first steps of the case study conducted in this thesis about the QuantyxRM software. Initially, it introduces how the software aligns with the business domain. Subsequently, the User-Centered methods employed to realize the redesign of QuantyxRM are presented.

Chapter 5 focuses on the QuantyxRM redesign results. Here, it describes how user-centred methods are employed, their outcomes, and how these are translated into design solutions. In the end, RQ3 is answered following the user test results.

In Chapter 6, the thesis work is discussed. First, the limitations of the work performed are presented. Next, the future work that should be done to broaden the research on this topic and further validate the main findings is presented. Finally, the next steps to be taken to conclude the design work of the QuantyxRM software are listed.

Finally, Chapter 7 encompasses the concluding section, summarizing the essential findings and implications of the thesis.

1 | Background

1.1. UX and Usability

There is no single definition of user experience; it is interpreted as subjective, context-dependent, and dynamic [27]. Formally, UX is defined in ISO 9241-210 as:

> *"user's perceptions and responses that result from the use and anticipated use of a system, product or service"* [22]

Norman, the inventor of the term UX design, considers the user experience to be everything involving interaction with a specific system, product, or service [45]. In this case, the vision of user experience is likened to an exchange, which is a less specific term compared to usage. Expanding on what is defined as user experience, with everything a person has directly or indirectly experienced through interacting with a system.

Closely related to the concept of UX is that of usability. However, usability can be considered part of UX, which is understood to be a broader concept [4]. UX involves the comprehensive perspective of how a user interacts with a system, including its learnability, efficiency, pleasantness, and beyond. Usability, on the other hand, is an essential quality attribute of the UI and focuses primarily on the system's ease of learning and use. [45]

Referring back to ISO 9241-210, the term usability is defined in the following manner:

> *"the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"* [22]

However, this concept is more precisely defined through its division into five key aspects, also described as usability goals - the usability objectives a design must meet [51].

Firstly, 'learnability' assesses the ease with which users can complete basic tasks upon their initial encounter with the design. Next, 'efficiency' measures the speed at which

studies can be performed once users have familiarised themselves with the design. The third aspect is 'memorability', which gauges how straightforward it is for users to regain proficiency when returning to the plan after a period of non-use. Fourthly, 'errors' look at the frequency and severity of mistakes made by users and how readily they can recover from them. Finally, 'satisfaction' evaluates the overall user enjoyment and comfort in using the design.
These five dimensions together form the holistic notion of usability, defining the user-friendliness and efficacy of a design. [40]

I greatly appreciate this definition of usability because, although it may not be the best or most generally applicable, it is perfect for understanding what is considered usability within the context of a software product or service. This is the context in which I am working on my research thesis.

1.2. Enterprise UX and Enterprise Software

Enterprise User Experience has become increasingly pertinent, especially given the recent surge in remote work [43]. This principle fundamentally underpins the design of Enterprise Software, emphasising a user-centric approach within the business environment [64]. The term 'Enterprise User Experience' emerged when companies began to realise that the concept of User Experience wasn't merely a design buzzword used for consumer applications but that when applied to the internal applications used by employees, the ROI tied to their development was substantial [57]. In principle, Enterprise UX illustrates the growing recognition of user experience as a critical component in developing and utilising workplace software solutions.

Enterprise UX is intrinsically linked to Enterprise Software, specifically designed for business domains and tailored for unique roles [58]. This software is characterized by its complexity, specialization, the dichotomy between procurers and end-users, specific user needs demanding comprehensive functionality, a unique market structure with reduced competition, and the influence of legacy solutions [64].

Despite the substantial financial commitment, the inherent benefits justify investments in Enterprise Software [57]. However, its complexity often poses challenges for company employees. Six emphasizes inefficiencies resulting from poor design, elevated training and support costs, and potential declines in morale and productivity among users. These challenges elevate operational costs and risk employee job satisfaction and security. [58]

More generally, Enterprise UX lacks a universally accepted definition, as it is not defined within a standard or consistently identified by industry experts. However, a good definition is provided by Walter and Braunsdorf:

"the design of products for people at work"

This definition, no longer directly linked to information technology, characterises Enterprise UX as all designs created for products used during employees' work. However, when Walter and Braunsdorf delve further into the primary application areas for this concept, they will only analyse software products. This highlights how Enterprise UX, for non-software products, has yet to gain interest or may not be effectively applicable. [64]

1.3. User-Centered Design

User-Centered Design (also called Human-Centered Design, HCD) is the most used approach for Interaction Design and HCI development [27, 51].

The concept of Interaction Design was initially articulated by Moggridge and Verplank in the late 1980s [23]. This approach focuses on the design of interactive products and services, with a specific emphasis on expected user interactions [51].

Concurrently, the closely related discipline of Human-Computer Interaction emerged during the same period. HCI is fundamentally rooted in the interaction between humans and tools and information processing at the genesis of the computing era. It integrates psychology and information processing elements to enhance how individuals communicate and engage with technology. [27]

In the 90s, Gould, Boies, and Lewis defined the four rules of 'integrated design', which laid the groundwork for what would later become the ISO 13407 standard Human-centered design processes for interactive systems [44]. ISO 13407 has subsequently been updated and evolved into ISO 9241-210 [22], with the latest update related to 2019. In this, the HCD is introduced as:

"an approach to interactive systems development that aims to make systems usable and useful by focusing on the users, their needs and requirements, and applying human factors/ergonomics, and usability knowledge and techniques. This approach enhances effectiveness and efficiency, improves human well-being, user satisfaction, accessibility and sustainability; and counteracts possible adverse effects of use on human health,

safety and performance."

The ISO 9241-210 standard delineates the formulation of any interactive system that requires incorporating four interconnected human-centred design activities, emphasizing their crucial role in this development process. [22]

The method of user-centered design begins with the recognition of a design need. Following this, the designer must grasp and define the context of use, which involves pinpointing the user, the user's environment, and their objectives and tasks. Next, the designer is tasked with identifying the user and the requirements. This entails outlining the desired features, usability goals, and user experience objectives that need to be attained. The outcome of this design process is then manifested in the solution. Evaluation is carried out on the design to ascertain whether it fulfils the user's needs. This sequence of steps should be iterated until the design solution meets its predetermined objectives.[8]

This process is illustrated in Figure 1.1.

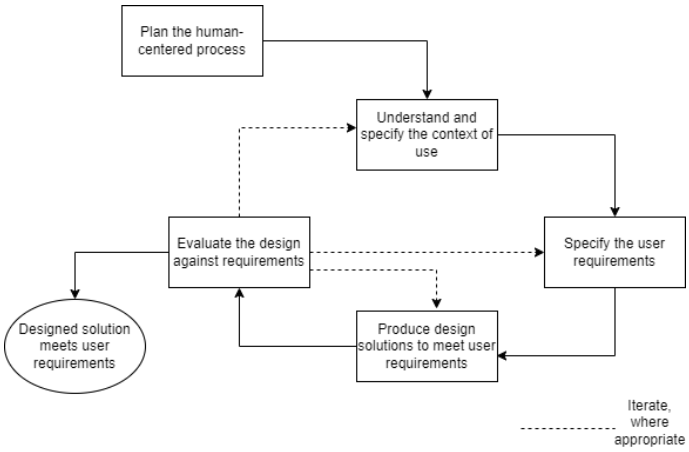


Figure 1.1: UCD process [22]

1.4. Information Architecture

The modern information explosion poses challenges and opportunities for communication and information design. The field of Information Architecture (IA) offers valuable insights and methodologies for effectively structuring, organizing, and labelling content to address these challenges. Incorporating IA principles into the design and development of information systems can improve usability, ensuring that complex multivariate information is accessible and easily understood by audiences. [31]

Starting from the mid-1970s, the discipline of Information Architecture, understood as the solution to basic problems of accessing and using vast amounts of information, began to evolve. The term was first used to signify its modern meaning during the American Institute of Architecture conference in 1976 by Richard Saul Wurman. [53]

However, it was deeply examined and elucidated by Morville and Rosenfeld in 1998, within the book *"Information Architecture: For the Web and Beyond"* which has been updated in four editions, with the latest one dating back to November 2015 [55].

Based on the insights from the latest edition of Morville and Rosenfeld's book, Information Architecture can be defined as:

> *"The creative and scientific techniques used to structure and categorize information elements, with the purpose, is to facilitate smooth search and navigation actions for users on a website, ultimately enhancing the overall user experience."*

This definition emphasizes the relationship between IA and UX, clarifying that a well-implemented IA can enhance the user experience. Indeed, in this thesis, we will refer to this definition when discussing the topic of IA. More broadly, we will frequently reference the book by Morville and Rosenfeld, often regarded as "the bible of IA". [61]

1.4.1. Information Architecture Components

Morville and Rosenfeld delineate the concept of Information Architecture into four distinct components.

Organization system refers to how information is arranged. Defining an organisational schema is crucial in this component, according to which information is grouped and segmented.

Labeling System refers to how the system "communicates" with the user. The labelling system should ensure that the terms used within the system are understandable to the user and cannot be misconstrued.

Navigation system refers to how the user moves within the system. It outlines the paths from one element to another, delineating the possible routes to reach a specific point, starting from your initial state.

Searching System determines what information can be searched and how it's done. It establishes the search algorithms, how queries can be executed, and how the results are displayed. However, this is the only component not addressed in this thesis. [55]

1.4.2. User-Centered Information Architecture

The Information Architecture, as presented by Morville and Rosenfeld, can be designed using a user-centred approach based on the needs and models of the user, collected through User-Centered Design methods [19].

1.5. Visual Hierarchy

Visual attention significantly influences viewing behaviour, as individuals typically process one visual stimulus at a time, with adjacent items often competing for attention. This principle holds particularly true in web page design, where perceptual elements like text, images, videos, and font size communicate with and impact the user. [50]

In this context, Visual hierarchy is defined as the order in which information is communicated to a user [12]. A clear visual hierarchy guides the eye to the most important elements on the page, and it can be created through variations in colour and contrast, scale, and grouping (proximity and common regions) [17].

In this study, we employ these principles to enhance the software's user experience, having understood the information hierarchy through the UCD approach.

1.6. User Research

User research is a part of the UX design process, and the methods associated with it are fundamental for collecting information about users [28].

A globally accepted standard definition of User Research, similar to that given for User Experience in ISO 9241-210:2019, does not yet exist, and the term can sometimes be vague [15]. Generally speaking, user research can be the foundation upon which projects that follow a user-centred approach are carried out [7].

There are numerous user research methods, and it's essential to select them based on

the researcher's goal and the context of use [54]. In this thesis, user research is heavily employed, and the methods used will be applied for gathering user requirements, testing the application, and validating results and theories.

2 | Methods

2.1. Systematic Literature Review

The Systematic Literature Review (SRL) is considered the gold review standard. SLR, mainly established in medical science, synthesises research findings in a methodical, clear and repeatable manner [9]. A systematic review is a method to gather all pertinent empirical evidence based on set criteria to answer a specific research question or hypothesis. Using explicit and systematic methods in reviewing articles and all available evidence helps minimise bias, resulting in reliable results that can inform decision-making and support drawing valid conclusions. [36]

The primary attributes and procedures linked to Systematic Literature Review and its affiliated process include:

(i) Clearly defining the research question that the study aims to address,

(ii) Establishing distinct objectives that utilise a transparent and replicable method,

(iii) Constructing search strings that encompass all pertinent studies meeting the eligibility criteria,

(iv) Evaluating the quality and validity of the selected studies, for instance, assessing the risk of bias and confidence in aggregate estimates,

(v) Systematically presenting and amalgamating the data gleaned from the chosen studies,

(vi) Ensuring the study's findings are accessible for scientific endeavours and decision-making. [37]

However, it's essential to understand in which cases it is important to employ this research

approach. Literature reviews lay the groundwork for constructing novel conceptual models or theories. They prove beneficial when the objective is to offer a comprehensive overview of a specific issue or research problem [59]. Based on the last sentence, I understood that it was necessary to use an SLR methodology to be able to answer RQ1.

2.1.1. PSALSAR Framework

Over time, various methods have been established for conducting a literature review, primarily chosen based on the research question and topic. From these, numerous standards and guidelines explicitly address how literature reviews should be reported and structured [59].

The topic I'm conducting the literature review is still relatively unexplored in research, prompting me to choose the PSALSAR Framework. This tool facilitates a comprehensive quantitative and qualitative content analysis assessment within the literature review. [34]

PSALSAR framework derived from The Search, Appraisal, Synthesis, and Analysis (SALSA). SALSA is a methodology used to define the search protocols that should be adhered to in a Systematic Literature Review [18]. Compared to SALSA, the PSALSAR framework has added two new steps: Protocol and Reporting. The former defines the purpose of the study. In contrast, the latter describes how the analysis results should be reported, which is helpful if you want to create a Journal article related to the SLR or if you need to summarise the report result for the larger public [34]. In my research context, I focused on analysing the truly beneficial elements, specifically the initial five components, using the PSALSA framework.

Step 1: Protocol

A research protocol for systematic literature reviews is essential to ensure clarity, consistency, and the ability to replicate the study, which are vital traits that define the systematic nature of literature reviews [3]. The main objective is to define well-targeted research questions for identifying the correct research method. In general, the research questions must aim at defining this set of concepts - the current state-of-the-art, the types with the most and most miniature studies, the standard approaches used, the varying development paths and existing gaps, the challenges hindering these studies, and the learned lessons and future directions for these studies. [34]

Step 2: Search

The second step aims to define the way research is carried out. The search needs to be executed using specific search strings tailored for databases most relevant to the research topic. These search strings must be finely tuned and closely aligned with the research questions. [10, 34]

Step 3: Appraisal

The appraisal step involves assessing the chosen articles about the review's objective. This entails sifting through the selected literature to pinpoint papers pertinent to the review. Two primary steps are involved: first, the selection of studies based on predefined inclusion criteria, and second, the assessment of the quality of these selected studies. The criteria for inclusion and exclusion are appraised during the initial screening of titles and, subsequently, the abstracts. The quality of the articles is then assessed by conducting a detailed review of the full text of the remaining papers. [34]

Step 4: Synthesis

The synthesis step involved extracting and categorising pertinent data from the chosen papers [34]. The selected papers will then be organised one by one in such a way as to be prepared for honest analysis.

Step 5: Analysis

The step of analysis involved evaluating the compiled data and extracting significant insights and conclusions from the selected papers. During this stage, answers to the research questions would be formulated. This phase includes qualitative and quantitative interpretations and narrations of the results, discussion generation, suggestions for future research directions, and conclusion. [34]

2.2. Information Architecture Diagramming

Representing the product of work done in Information Architecture is an extremely complex issue. There are no comprehensive ideal solutions, and standards for its representation have not yet been defined [55].

How Information Architecture is represented depends largely on the audience it is targeting and the specific components chosen to be incorporated within it. Blueprints and

Wireframes are typically the most used models for IA graphical representation. [55].

In this thesis, we will use both. By utilizing Blueprints and Wireframes, along with their respective tools, we can accurately represent the entire Information Architecture of our software across two comprehensive models. Blueprints allow us to comprehend the organization of the pages, the application's navigation, and the labelling system from a perspective where the page acts as the elemental unit. On the other hand, Wireframes facilitate detailed visualization of the page content, its hierarchy, and its interaction within the overall site navigation, where the page is considered a macro-element.

2.2.1. Blueprints

Morville and Rosenfeld define "Blueprints" as a valuable tool for representing the interconnections between pages and other content elements. These are crucial for illustrating the organizational structure, navigation, and labelling systems. As shown in Figure 2.1, a commonly known example of a Blueprint is a "site map". [55] There are various types

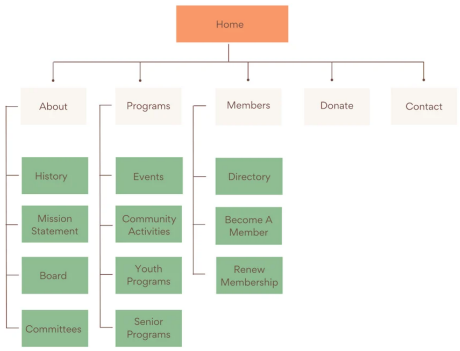


Figure 2.1: Generic Site Map [14]

of Blueprints, and their structure and representation significantly differ, primarily based on the type of software or interface they are intended to represent. In this thesis, I will use Visual Vocabulary as a foundation. It is a framework providing a clear set of terms and syntax for visually communicating components and their connections, suitable for creating structured representation diagrams. [16, 55]

2.2.2. Wireframes

Wireframes can be defined as high or low-fidelity representations of webpages, demonstrating the concept of page-level layouts. They serve a different purpose than Blueprints. Indeed, Wireframes primarily focus on the content part of website pages and how this

content interacts with navigation. Thus, they present the information hierarchy on a page. [29, 55]

Figure 2.2 displays an example of a Wireframe.

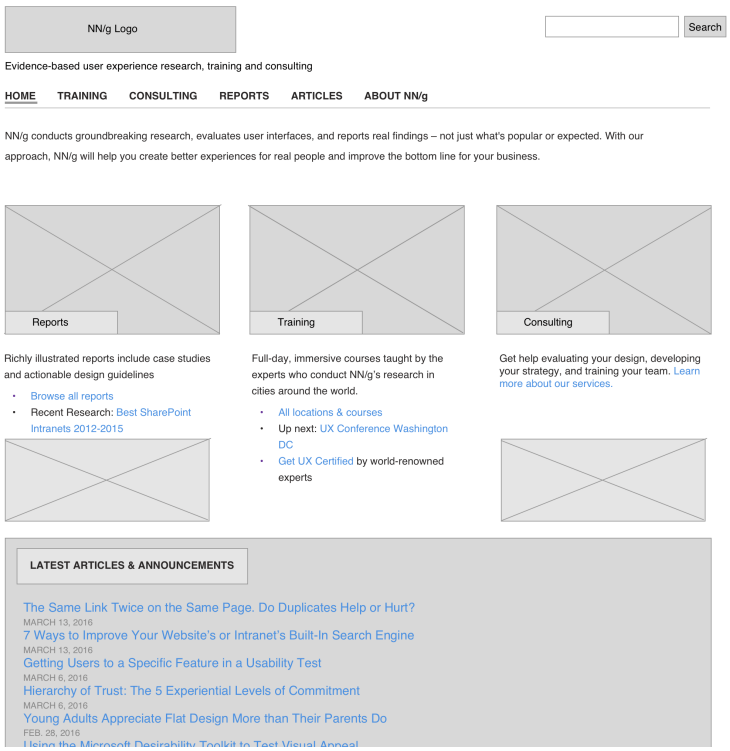


Figure 2.2: Generic Wireframe [29]

In this thesis, I employ Wireflows, a hybrid tool that combines the features of Wireframes and Flowcharts to illustrate the interactions among different Wireframes [29].

2.3. Jesse James Garrett's Visual Vocabulary

Jesse James Garrett's Visual Vocabulary is based on a simple conceptual model encompassing Information Architecture and Interaction Design. However, each aspect of the diagram serves slightly different purposes. In both cases, the obtained diagram focuses on the macrostructure, offering enough detail to allow team members to see the big picture. In this vocabulary, the system presents pathways to the user; the user moves along these pathways through actions, which prompt the system to generate outcomes. There are two main components, the entities, i.e. pages, files, and their groups, and the relationships, i.e. connectors and arrows. [16]

Only the components used within the thesis are presented below, but other types, not included, could be helpful in other cases.

2.3.1. Simple Elements

The foundational elements of this vocabulary are pages, files, or their groups, as depicted in Figure 2.3. Pages represent the essential navigation elements, each corresponding to a

Figure 2.3: Starting from the left, Page, File, Group of Pages, Group of Files [16]

unique URL. In contrast, files correspond to data packets that do not possess navigational properties. A rectangle symbolizes pages, while an icon with a folded corner represents files. A stack of pages indicates a group of pages with similar functionalities, the navigation properties of which are irrelevant to the site's overarching structure. Similarly, a file stack signifies a group of files treated again from a navigation perspective and can be classified as a single entity.

2.3.2. Relationship

The relationships between elements are represented with simple lines or connectors. These indicate how the elements are connected, while the arrows indicate the directionality of these links, i.e. how the user can move from a given page. These arrows are not like the ones that point to a one-way street; the user is not prohibited from going back; the arrow indicates the preferred direction for the user to follow. If upward movement should be prohibited, a bar is used at the opposite end of the arrow.

There is the possibility of adding labels to the connectors or arrows to indicate an action, a note or a reference.

2.3.3. Concurrent Sets

A concurrent set is utilized when a user's action generates multiple and simultaneous results; an example is shown in Figure 2.4.

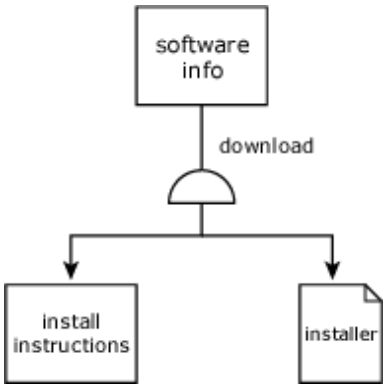


Figure 2.4: Concurrent Set Example [14]

2.3.4. Continuation Points

Continuation points are used to divide diagrams into multiple pages by linking them together. Figure 2.5 shows a diagram with four entities, where entity D is shown within the second diagram, which specifies how this is a continuation of diagram A.

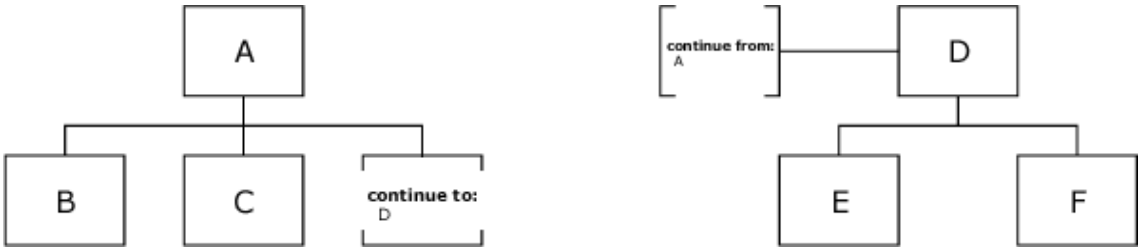


Figure 2.5: Continuation Points [14]

2.3.5. Areas and Iterative Areas

The "area" element is used to identify a group of pages that share one or more common attributes. It represents elements such as pop-up windows and pages with unique designs. In the case of lists, for instance, pages containing a series of identical items with links to specific, similar pages, the "iterative areas" element is used. This appears as a stack of areas, precisely as in the case of groups of elements. Figure 2.6 shows an example of a pop-up window.

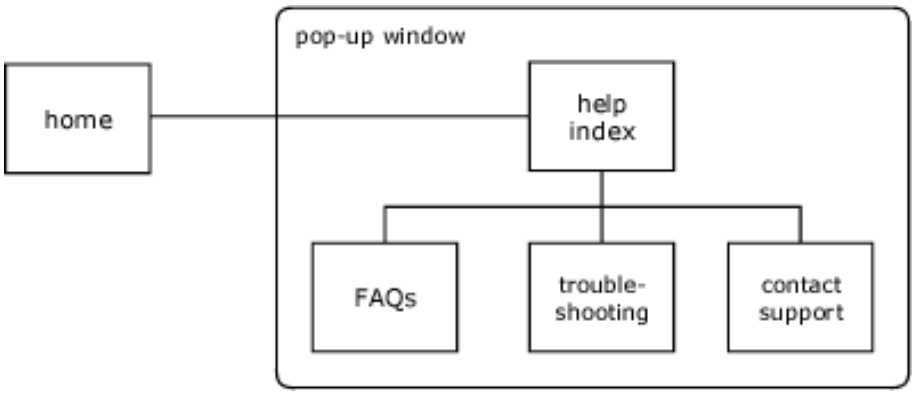


Figure 2.6: Area element for pop-up windows [14]

2.3.6. Conditional Elements

Mainly presented in Figure 2.7, conditional elements allow the user to access specific pages, thus blocking or revealing certain paths based on a specific condition. Generally,

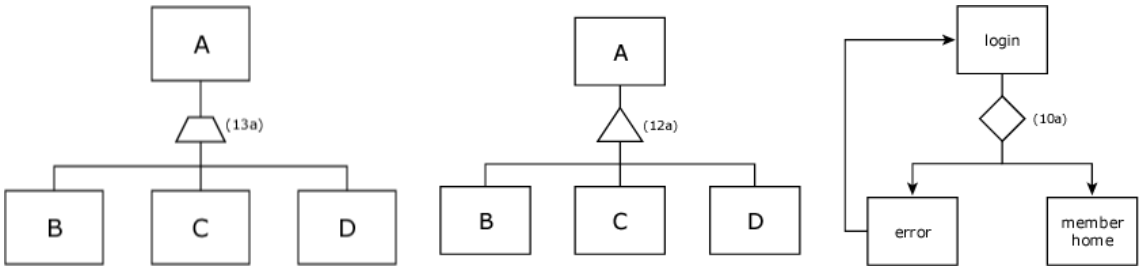


Figure 2.7: Conditional selector, Conditional branch and Decision point [14]

the system monitors one or more attributes. These attributes can be user-specific (referring to the user profile), session-specific (related to the login status), content-related (linked to the topic), or related to "world" elements (such as time or date). These attributes possess values, and the pairing between an attribute and its corresponding value is termed a condition. The system evaluates these conditions to establish their validity.

Garrett outlines five conditional elements, which vary according to the type of condition you want to depict. The simplest is the conditional point, symbolized by a square tilted at 45 degrees. Following a user action, one path is pursued if a condition is satisfied. Otherwise, another one is taken. The second is the conditional connector and arrow, represented by a dashed line (or dashed arrow), which can only be followed when the condition is met.

Next, we have the conditional branch, represented by a triangle. In this case, the downstream paths are mutually exclusive. Only one way can be selected by the system. This is

very similar to the conditional point, but with the difference that the decision is not made after a user action but is predetermined by the system. From the user's perspective, there will only be one path. The fourth element is the conditional selector, represented by an isosceles trapezoid. This functions as a filter, where only paths that meet the condition are shown, and subsequently, the user can decide which path to follow.

2.4. Wireflows

Wireflows were conceived to address the challenges associated with communicating the design of interfaces using only Wireframes. They stand as a simple and robust deliverable, defined by Laubehimer as:

"a design-specification format that combines wireframe-style page layout designs with a simplified flowchart-like way of representing interactions."

Starting from its definition, Wireflows fundamentally rely on the page concept. It begins by representing the content on a specific page and then illustrates how it interacts with the website's navigation flow. It is critical to highlight the elements that enable a certain interaction or movement within the application's navigation structure. [29]

Furthermore, in the case of a conditional branch, it is always vital to specify the condition that must be met to follow a particular path. In the model presented in this thesis, we will use red to indicate elements that "trigger" an interaction and unidirectional arrows to depict the transition from one page to another.

2.5. User Interface Prototype

In the context of a human-centred design methodology, prototyping plays a pivotal role and serves many functions [21]. They are the most straightforward method for testing design solutions, as users can directly interact with them [49]. This thesis will use them to test design solutions.

Pernice presents various prototypes based on the specific situation to be represented and tested. Indeed, prototype selection will significantly depend on the test's objectives, the design's maturity, the tools utilized to create the prototype, and the resources accessible for support before and during usability tests. Regardless of the type of prototype used, testing it will provide valuable insights into users' interactions and reactions, thereby informing further design improvements. [49]

User Interface Prototypes are divided into High-Fidelty and Low-Fidelty Prototypes [26, 49]. In general, High-fidelity prototypes endeavour to emulate the final product with a high degree of accuracy, encompassing aspects such as layout, hues, dimensions, and the like. These are typically fashioned using specialised instruments. Conversely, low-fidelity prototypes centre primarily on fundamental concepts, such as the array of controls and the sequence of operations, and can be crafted utilising rudimentary materials like paper and pencil. [26]

The prototype will be paired with a testing methodology in this study to evaluate the design solution's page navigation and visual hierarchy. It is essential that these prototypes closely mirror real user interfaces to achieve accurate results. Thus, High-Fidelity prototypes will be employed, featuring clickable links and menus, automatic responses to user actions, a realistic visual hierarchy, prioritization of screen elements, and proper screen size and content.

2.6. Card Sorting

Card sorting is a design framework based on Information Architecture employed to identify, classify, and structure website content [11]. In brief, it is a data-gathering technique that effectively captures users' perspectives on the connections between different items [46].

Sherwin defines card sorting as a UX research method in which study participants sort individual labels written on notecards (cards) based on criteria they deem logical. It reveals the structure of the target audience's domain knowledge and aids in crafting an information architecture that aligns with the users' expectations. [56]

Various ways exist to conduct card sorting, and selecting the method best fits the specific case is important. Card sorting primarily has two variations: Open and Closed Card Sorting. In Open Card Sorting, users must assign names to the groups of cards they have created. On the other hand, in Closed Card Sorting, users must organize the cards into pre-established categories. I will utilize open card sorting in this thesis. This approach accurately reflects the user's mental models without bias or limitations arising from predefined categories. [11, 56]

In detail, the card sorting study used in this thesis was conducted online, utilizing the

designated tool from OptimalWorkshop [1]. The Open Card Sorting procedure involved 29 cards, each representing the primary navigation pages of the software. Users were initially required to regroup these cards. Subsequently, they were asked to assign a name to each group. In addition, users were given the context and functionality of each card and the option to rename it. This procedure allowed for initial validation of the labelling section of the information architecture. More study details can be found in Chapter 5.

2.7. Surveys

There are numerous types of surveys, and in this thesis, we will use them to understand which information contained within the pages of the QuantyxRM software should be inserted and how to hierarchy them. Generally, Muller defines a survey as a method for gathering data from a subset of individuals to generate results that can be generalized to a broader target population [39].

Surveys are particularly effective when used with other methods [39]. In this thesis, they were also employed as pre-study questions to categorize participants and enable us to identify the various subgroups within the broader target participant group.

The survey process can be systematically broken down into six sequential steps: establishing research goals and constructs, determining the population and sampling techniques, designing the questionnaire and addressing potential biases, reviewing and pretesting the survey, implementing and launching the study, and finally, analyzing the data and reporting the findings [39].

The initial two stages involve defining the objective and identifying the target participants based on this objective. Particular emphasis should be placed on the stages related to questionnaire design and biases, as well as data analysis and reporting. The latter will be elaborated and described in Chapter 5.

2.7.1. Questionnaires design

The cornerstone of surveys is the questionnaire. A questionnaire comprises a series of questions that can be classified into two categories: Open-ended or Closed-ended. Open-ended questions allow the respondent to express their answer in their own words. Conversely, in closed-ended questions, the respondent must choose an answer from the given options. It is possible to combine these two types by using closed-ended questions that include an

option for respondents to provide their answers if they believe the provided choices are not suitable. [24]

In this paper, I employed two distinct types of questionnaires. The first is a pre-study questionnaire, introduced at the outset of each User Research Method to gather professional demographic data on the participants. Each questionnaire consists of 5 single-choice closed questions, where the participant must choose the option that best describes them. An example is presented in Figure 2.8. The second type is a genuine survey. This was

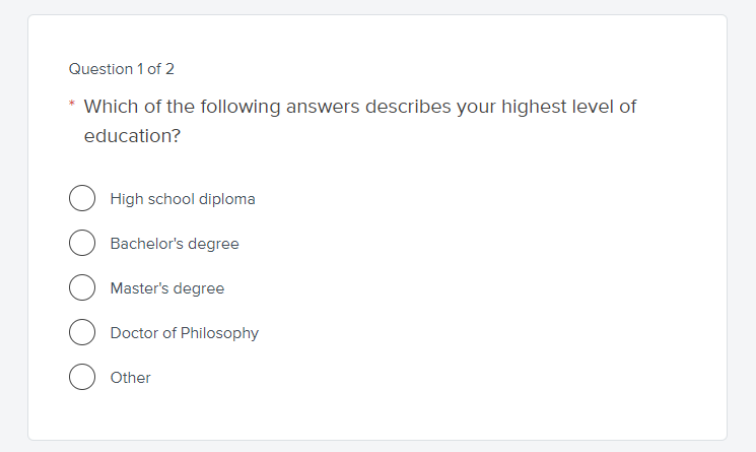


Figure 2.8: Example of a single-choice question

divided into two sections.

The first section aims to gather data concerning the organization of information within the main pages of the software, corresponding to the three primary services offered by the company: Risk Assessment (evaluating the risk associated with a fund or financial asset), Performance Simulation (analysis of projected cash flows and fund metrics), and Peer Group Analysis (examining the competitive positioning of a managed fund). To achieve this, ranking questions were employed, where users must prioritize a series of items based on their significance within the specified context [39]. An example of such a question is illustrated in Figure 2.9. In the second segment of our study, I sought to identify the

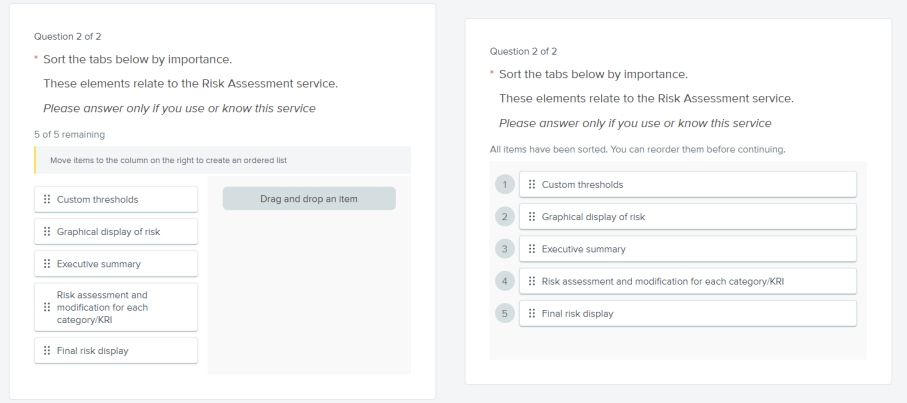


Figure 2.9: Example of a ranking question, first and after the ranking

essential features that users expected from the two components: the Home Page and the Table. I designed two multiple-choice questions, allowing users to supplement their answers with additional comments or ideas. The list of potential features provided in the options was derived from competitive analysis and discussions with the Product Owner.

2.7.2. Considered Biases

In conducting a comprehensive survey, being fully aware of potential biases that can arise during this User Research Method [24, 39] is crucial. The biases considered in my surveys are listed below.

Satisficing Bias - Participants are generally expected to undertake four steps when approaching a question. Initially, they need to understand the question and decipher its purpose. Following that, they should retrieve pertinent details from their memory. Subsequently, they must consolidate the information they recall into a singular opinion. Lastly, they have to convey this opinion by choosing from the options provided by the question [24]. This type of process is cognitively demanding. Therefore, if the participant is not assisted in reducing this cognitive load, either by removing some of these steps or by aiding in their completion, there is a risk of falling into the satisficing bias [39]. Due to the excessive cognitive burden, a participant might choose the first seemingly acceptable response. Without considering other options or which one might be the best for him [24, 39]. Questions are formulated concisely and unambiguously to mitigate this bias without including a "no option." Only essential questions are selected to ensure the brevity of the questionnaire.

Acquiescence Bias - When faced with agree/disagree, yes/no, or true/false prompts, some

tend to agree with the statement regardless of its actual content [39]. Questions with the above prompts and agreement statements are excluded to avoid this bias.

Social Desirability Response Bias - This bias is related to the desire to be viewed favourably by others, which can lead to increased rewards and decreased punishments [24]. Primarily encountered in in-person, nominal, or recorded surveys, such behaviour can arise even in online surveys when voting behaviour, religious beliefs, sexual activity, patriotism, bigotry, intellectual capabilities, illegal acts, acts of violence, and charitable acts are included [24, 39]. Thus, exercising caution and avoiding these pitfalls is vital when designing a survey. To avoid this bias, the surveys are made fully anonymous and self-administered.

Response Order Bias - Research has indicated that the sequence in which answer choices are displayed can influence their selection [24]. Response Order bias refers to the inclination to choose options at the start or the finish of a given list or scale. Participants often subconsciously perceive a relationship between items placed close to each other in a sequence. Items at the beginning or on the left are often seen as "premier," while central options in a scale with no inherent sequence are seen as the normative choice [39]. This bias is closely related to the satisfaction bias. It underscores the importance of ensuring that responses are not excessive and are perceived by the user as a cohesive whole rather than a disjointed series [24]. Responses to demographic questions were arranged in ascending order to address this bias, while those related to software topics and content were randomized.

Question Order Bias - Surveys can also be affected by the questions' order. Every question asked in a survey can influence respondents, potentially skewing their responses to the following questions. Generally, it is crucial to follow a consistent logical model when sequencing questions. It's essential to avoid exhausting or frustrating the participant from the outset [39]. My surveys organise questions from the most general to the most specific. Initial questions are demographic, ensuring that participants are not overwhelmed at the outset. Furthermore, the questions have been grouped into three sections based on type and topic: multiple-choice demographic questions, ranking for understanding visual hierarchy, and checklists of key content. This structuring aids in easier comprehension for the participants and ensures a more fluid completion process.

Recall Error - Significant yet straightforward bias arises when participants are asked to recall past behaviours, actions, or attitudes [39]. This task requires them to delve into their memories, potentially leading to prolonged time spent on the question and subse-

quent frustration, especially if they cannot retrieve the desired information. To address this, it's essential to assist participants with prompts or phrases that can help jog their memory [24]. To avoid this bias, all questions and answers were presented straightforwardly. Additionally, all necessary information for answering was provided within the test itself. For questions that required specific knowledge, responses were only solicited from individuals directly involved in that field.

Broad, leading and double-barreled questions - Avoiding vague questions that lack clarity or contain ambiguities is crucial. Avoid questions that might be biased or lead respondents to a particular answer by implying a desired response from the researcher. Also, sidestep double-barreled questions that tackle multiple subjects but only permit a singular answer, as they can compromise the reliability and validity of the data [39].

2.8. Tree Testing

The Tree Test was first introduced by Spencer in 2003 as a tool to assess the hierarchical structure of Information Architecture. It is presented as a user research method that allows participants to evaluate the Information Architecture quickly and without needing visual elements or the development of interfaces. [4, 62]

A formal definition of the Tree Test is provided by the Nielsen Norman Group (NNg). They compare it to a usability test, as both focus on completing a specific task, specifically locating a particular resource. The definition given by NNg is as follows:

"A tree test evaluates a hierarchical category structure, or tree, by having users find the locations in the tree where specific resources or features can be found. [30]."

There are several tools available for conducting a Tree Test. In this study, we have employed Treejack [2]. This tool is among the most popular due to its ease of learning, user-friendliness, clear and understandable interaction, and the simplicity with which users can master it [62].

2.8.1. Tasks

Once the tree structure is established and the testing tool is chosen, it is essential to define the tasks that users must perform for the testing [62].

How users execute tasks is largely consistent; it typically involves searching for a tree

resource that they believe represents the correct solution. However, these tasks can be employed differently, depending on the specific aspect under investigation [30]. This section will enumerate the types used in the Tree Test implemented in this thesis.

Resource finding - Participants must identify a specific resource within the tree structure. I employ this task to determine whether the primary business services have been appropriately hierarchized and labelled.

Potential problem areas - Participants were asked to identify a resource within the tree that pertains to a specific issue or challenge that emerged during the card-sorting process.

When designing tasks, it is crucial not to embed the answer label within the question or use overly lengthy and complex sentences to convey the concept to the user. The correct approach is to provide participants with a clear and concise description of the desired resource to test. [30]

2.9. Usability Test

Usability testing is one of the most commonly employed methods in UX research. This is largely because it is cost-effective and serves multiple purposes. This study identifies design issues, generates insights for enhancement, and evaluates and compares it against another design that fulfils the same requirements. [38]

In detail, a usability test involves a study in which the researcher asks the participant to perform specific tasks using an interface. As the participant completes these tasks, their behaviours are closely observed. Feedback from the user is collected both during and after the task completion. [38]

Usability testing comprises three main components: the facilitator, the task, and the participant. The facilitator's role is to observe and, when necessary, respond to user inquiries, assisting without influencing the user's behaviour. The task represents a specific activity the participant must perform during the test, mirroring an activity a user would realistically undertake. Lastly, the participant should be an authentic user, fitting the defined characteristics of the user group for the given interface or product. [38]

The number of participants in a usability test can vary based on the type of test being conducted. Generally, it is considered ideal to have five users for a usability test [41].

Various approaches to conducting a Usability Test are contingent upon the specific objective. However, I delineate the two primary methods. I identify these as the Qualitative and Quantitative Usability Test methods. Subsequently, I will detail the specific type utilized in this thesis [6].

The Qualitative Usability Test relies on user feedback and findings from observing their behaviours during task execution. These tests provide a wealth of information and are highly valuable when conducted rigorously without influencing participants, who must represent target users. With well-defined tasks, even a few users can offer insightful observations and highlight potential issues. [6, 38]

In qualitative testing, the "Think Aloud" technique is commonly employed. This method involves asking the participant to verbalize their thoughts continuously, providing researchers with insights into the participant's decision-making processes and the reasons behind specific choices. While this technique is valuable, it may not always come naturally to participants. Furthermore, potential complications can arise when researchers must engage with the participant for clarification or other reasons. Therefore, providing participants with clear instructions on effectively employing the "Think Aloud" method is essential. [42]

On the other hand, we have Quantitative Usability Tests, which necessitate a substantial number of participants, typically at least 30. These tests are grounded on one or more metrics collected during the evaluation [6, 38]. Commonly used metrics include the First Click Test, Success Rate, and Time to Complete.

In this thesis, I employed a purely qualitative usability test. However, I also collected data on success rate and time to complete to provide additional, albeit limited, information for a better comparison of different designs.

An A/B Usability Test was conducted in which two distinct designs were presented to partecipants. After performing tasks on both interfaces, the participant was asked to select the design he found more comfortable to work with during task execution. Additionally, user behaviour was analyzed, and they were solicited for feedback concerning any unmet expectations and issues encountered during use.

3 | Preliminary Results

3.1. Systematic Literature Review

An essential step to address the first research question is to understand the relationship between the User-Centered Design process and the development of the design of Enterprise Software. Kuusinen asserts that while UCD is well-suited for developing consumer software, it falls short in developing enterprise software that incorporates a business process [27]. Given the gravity of this statement and the burgeoning growth of the Enterprise UX concept, I deem it necessary to employ a Systematic Literature Review approach to derive substantial conclusions regarding the relationship between UCD and Enterprise Software. Consequently, I have applied the PSALSA method, as detailed in Chapter 2, in a step-by-step manner.

3.1.1. Protocol

The initial step pertains to establishing a protocol, which aims to determine the purpose of the research. This approach minimises biases during the research phase [34].
The objective is aligned with that of research question 1:

"to understand the state-of-the-art of User-Centered Design approach in the development of Enterprise Software"

I thus regard this as the foundational concept upon which the Systematic Literature Review will be conducted.

3.1.2. Search

In this phase, the databases where research will be conducted are defined based on the area of the topic [48]. Subsequently, search strings are established, which will be used to search these databases [34].
After a preliminary search to understand which databases had major publications on this topic, only one highly relevant database resulted: the Association for Computing

Machinery (ACM). Table 3.1 displays the search strings and their corresponding results.

Databases	Searching String	No of articles
Association for Computing Machinery	business software AND user experience	403
	business software AND user-centered design	928
	enterprise software AND user experience	380
	enterprise software AND user-centered design	896
Total		2607

Table 3.1: Simplification of the search table of the PSALSAR method [34]

All terms from the search strings were queried at the Keyword levels. The research spanned the years from 2016 to 2023, and the latest date of result acquisition was the 25th of August, 2023.

3.1.3. Appraisal

The discovered articles are evaluated and filtered at this stage based on the inclusion and exclusion criteria presented in Table 3.2.

Criteria	Decision
When the primary topic is that of Human-Computer Interaction	Inclusion
The paper was published in a scientific peer-reviewed journal	Inclusion
The article should be written in the English language	Inclusion
When the article presents a user-centered approach employed for a corporate software solution	Inclusion
Papers that are duplicated within the search documents	Exclusion
Papers that are not accessible, review papers and meta-data	Exclusion
Papers that got published before 2016	Exclusion

Table 3.2: Source: Mengist et al. [34]

As shown in Table 3.1, the total number of papers found is 2607. Once these criteria are established, the papers sourced from the search are initially selected based on their title. The next step involves an examination of the abstract. If the documents progress beyond this stage, they are filtered according to the actual content within the paper. Finally, if the documents meet all the defined criteria, they are included within the SLR and advanced to the subsequent phase [34].

From the total number of papers found in the search phase, after an initial screening, 81 articles were selected as their titles indicated that the topic was relevant to HCI applied to software development. Among these, 38 were duplicates and were removed from the list. Therefore, 43 papers remained for further examination. Subsequently, I proceeded to analyze the abstracts of the remaining papers. Those not referencing User-Centered Design or User Experience applied in a corporate context were excluded. At the end of this phase, I obtained a set of 26 complete articles to analyze in detail. Two papers were excluded due to the language barrier, as they were not written in English. Ultimately, three papers were selected for analysis, as they specifically focus on the use of a user-centric approach in Enterprise Software.
Figure 3.1 presents a schematization of the article's selection process.

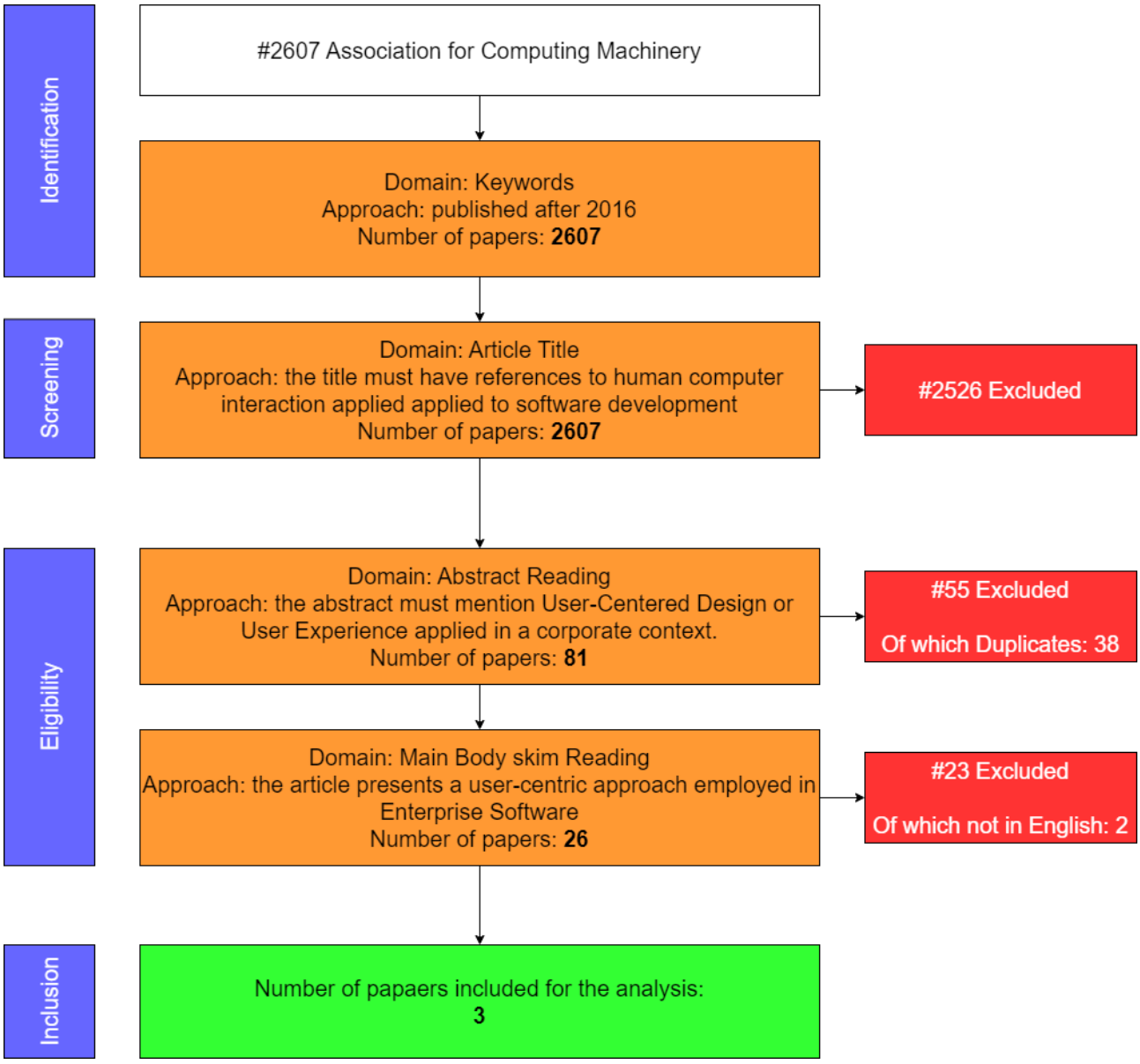


Figure 3.1: SLR articles selection flow diagram[34]

3.1.4. Synthesis

During synthesis, pertinent information is extracted and classified from the chosen papers to derive knowledge and draw conclusions [34]. In this study, qualitative data is collected. Table 3.3 displays various aspects for each included paper - the type of software being studied, the research goal, if available, the definition of users, and the methodologies employed.

Title	Software	Users	Goal	Methods
User Interface Redesign of Dental Clinic ERP System using Design Thinking: A Case Study [60]	Dental Clinic ERP	Front office personnel, Medical staff, Back office personnel	To reduce the high complexity of the user interface by implementing a redesign that is guided by the needs of stakeholders	Design Thinking, Empathy Map, Usability Test
PRO-UX: System Redesign Process for Improving the UX [47]	Web system for controlling service payments	Employees of the company	Defining a redesign process, and verifying its feasibility in actual software	Interaction Design Process, Contextual Inquiry, Heuristic evaluation
Iterative Mixed Method Approach to B2B SaaS User Personas [5]	B2B SaaS company that provides advertising technology solutions for clients such as brands and retails through a suite of products	Employees of the company that purchased the software	To generate user personas based on actual interaction of users with the platform for the internal leadership, support, development, product, and design teams to leverage their work	Surveys, User Interviews

Table 3.3: Sythesis of selected articles

In Table 3.4, for each article, the most crucial and relevant points are summarized, specifically focusing on those that pertain to the research question of this SLR.

Title	Summary
User Interface Redesign of Dental Clinic ERP System using Design Thinking: A Case Study [60]	Due to the high complexity of the software, employees faced challenges in performing their tasks effectively. A redesign process focused solely on the user interfaces was initiated to address this issue. This process employed the Design Thinking methodology. Given that multiple types of corporate users used the software, initial attention was directed towards creating user personas. Subsequently, an empathy map was used to identify the areas that caused the most frustration for users. Design solutions were then defined for these problematic areas and tested using prototypes. The results demonstrated that applying Design Thinking to redesign enterprise software yielded significant benefits.

PRO-UX: System Redesign Process for Improving the UX [47]	Due to numerous complaints from corporate employees about the excessive complexity of using the enterprise software, they undertook a study to develop a tailored redesign process. This process is grounded in interaction design principles and outlines a comprehensive workflow from initial design to actual development. The methodology is demonstrated through a case study. Initially, data is gathered on the most significant shortcomings of the software using contextual inquiry and heuristic evaluation. Subsequently, the software's information architecture is redefined to address these issues. Interfaces are created and tested through prototypes, first against business requirements and later with actual users. Finally, an iterative method is introduced for conducting the development phase. Upon testing, it was confirmed that this methodology can be effectively employed for redesigning enterprise software.
Iterative Mixed Method Approach to B2B SaaS User Personas [5]	This paper explores the need to understand user groups of a specific Enterprise Software as a Service (SaaS) application. We describe using a user-centric approach to categorize these groups, employing surveys and interviews for data collection. Beyond the methodology, our findings indicate that, despite various user groups within this enterprise software, they can be effectively condensed into fewer macro groups. This study also argues that enterprise software users are more straightforward to categorize than consumer software users, largely because they share a common goal that often aligns with business objectives.

Table 3.4: Summary of relevant information of selected articles

This approach provides more comprehensive information for the final data analysis.

3.1.5. Analysis

The last step of analysis focuses on the evaluation of the data obtained to draw real conclusions regarding the research and to be able to answer the research question [34].

Based on the literature reviewed, there is a consensus that Enterprise Software has thus far been inadequately developed from a User Experience perspective. User experience is considered crucial in these types of software, yet optimizing it is particularly challenging due to the inherent complexity of enterprise applications. [47, 60]

Another valuable insight is that Enterprise Software Users are typically employees of the companies that either purchase or develop the software [5, 47, 60]. Furthermore, my analysis confirms that users, even from diverse backgrounds, share common objectives that align with the organisation's goals [5].

A key conclusion drawn from the analysis of the articles is that user-centric processes applied to the redesign and improvement of User Experience for Enterprise Software are effective.

In the papers reviewed, I observed the application of Design Thinking and Interaction Design processes to Enterprise Applications for enhancing User Experience. Various user-centric methods were employed to collect user data and test new designs. These methods include Surveys, Interviews, Contextual Inquiry, Empathy Maps, and User Testing.

Based on my observations, in response to the research question posed in the Systematic Literature Review, I can assert that user-centric methodologies apply to Enterprise Software. The current state of the art focuses on enhancing User Experience by identifying and addressing enterprise users' critical pain points, primarily associated with these software systems' high complexity.

3.2. Enterprise Users

To better understand the current state of User-Centered Design in Enterprise Applications, the first critical step is to outline users' position and role within business software.

Mehta refers to the users of Enterprise Software as Enterprise Users [33]. These users have specific characteristics that define them as a unique user group. Notably, they use

the product for business purposes and are employees of the company that owns or has purchased the software. It's also important to note that Enterprise Users typically do not have the autonomy to choose the software they use; it is often contractually determined for them. [58, 63, 64]

In essence, Enterprise Users seldom have the opportunity to choose the software they use or the purpose of its use. The business makes these decisions for them, dictating their tasks, how they perform them, and where they do so. [35, 58]

This, however, does not rule out the validity of a User-Centered approach, nor does it diminish the utility of tools such as User Personas, User Journey Maps, Stakeholder Maps and so on. On the contrary, it underscores the need to adapt User Research methodologies for Enterprise Users. [5, 52]

3.3. Enterprise User Needs Vs Business Needs

The second critical aspect to clarify is how the requirements for design implementation are established.

User requirements are detailed descriptions of the functionalities, constraints, or properties needed to meet the user's demands, all written from the user's perspective. They stem from user needs, considering the users' goals versus the present and usage circumstances. This includes the characteristics of the users, their ongoing tasks, and their surroundings. [25]

In the case of Enterprise UX, user needs often align with the business's. This is because Enterprise Software is used for work-related purposes, making their primary needs centred on the correct and effective execution of their work [35, 63].

To apply a user-centric approach, adapting the design to users' needs and verifying that the design satisfies the business needs is essential [47].

3.4. User-Centered Enterprise UX Design

Based on the findings, I have chosen to chart my course by defining a more detailed process. Precisely a variation of the User-Centered Design approach specifically focused on redesigning Enterprise Software to enhance User Experience. While the processes exam-

ined in the SLR were grounded in Design Thinking and Interaction Design and indeed employed user-centric methodologies, they did not explicitly outline a User-Centered Design process. Therefore, I am defining such a process in my research thesis.

Miller has previously touched upon this subject, introducing the term "User-Centered Enterprise UX Design". However, he did not provide a clear definition [35].

As the basis of my theoretical definition of the UCD variant, I supplemented the SLR results with additional articles and sources to broaden the scope of information referenced. It should be noted that some insights have been obtained from non-academic articles and books, and their trustworthiness has been verified based on the reliability of the publishing entity and the author.

When designing Enterprise Software, the primary goal should be to simplify as much as possible the enormous complexity provided by the corporate domain and the large amount of data managed [35, 64]. Not focusing primarily on functionalities, which are business-driven and highly specific, requiring a high level of expertise and are inherently very complex [58].

It is crucial to understand that a minimal learning curve for software usage is acceptable as long as it helps minimize task completion times and the overall use of the software [13]. The ideal software usability should strive to be perfectly simple and intuitive. However, software with such high complexity might lose output quality if overly simplified. In this case, the solution is to make the First-Use Experience as comprehensive as possible, a topic that this thesis will not cover. [32]

3.4.1. Definition

Based on what has been discussed so far, in this section, I attempt to define the concept of User-Centered Enterprise UX Design, a variant that will follow the same process as the UCD approach.

Let's adopt the original definition given for User-Centered Design in ISO 9241-210 [22], adapting it to the concept of Enterprise UX.

> *"An approach to enterprise software development that makes the software usable and useful by focusing on reducing the complexity of its use by enterprise users, who are considered specialized users in the business domain."*

The User-Centered Enterprise UX Design is a subset of User-Centered Design specifically targeting user requirements that do not seriously compromise business domain and requirements. The primary objective is to reduce the complexity of the software, which, based on this research results, appears to be a consistent feature in this type of application.

In summary, in the context of enterprise software, it is advisable to prioritise simplifying functionalities execution, which are usually business-driven.

The UCD process is adapted based on the previous definition in the following.

The initial phase, about the specification of the context of use, is expanded because Enterprise Software's context is the workplace, and the users are the company's employees. Consequently, this phase will also focus on analyzing the business context.
The term "business context" refers to the company's domain and comprehends the business needs the software addresses. It is important that the business needs are clear and the software truly presents the business domain.

The second part, related to user requirements, will prioritise gathering user requirements focused on reducing complexity. This is to design business-required functions that are clear and efficient for enterprise users.

The third part delves into devising design solutions that align with user needs, while the final focuses on evaluating these solutions.

The process is illustrated in Figure 3.2.

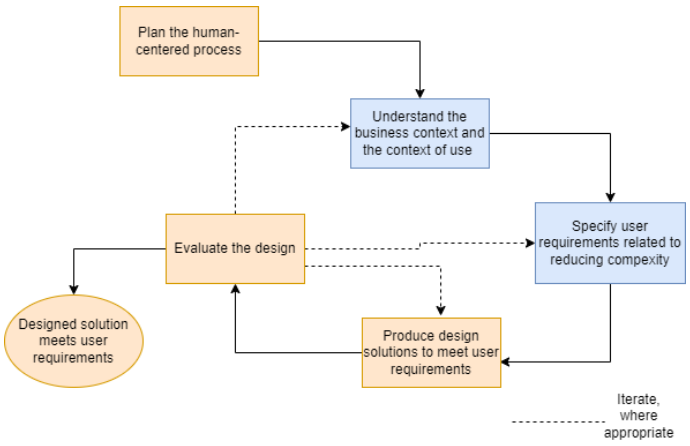


Figure 3.2: User-Centered Enterprise UX Design process

4 | Case Study

As mentioned in Chapter , this thesis will use the enterprise software QuantyxRM as a case study.

The previously defined UCD process is employed for the user-centric redesign of QuantyxRM. This approach first involves understanding the work and business context, then collecting user requirements that focus on reducing system complexity using specific user-centric methodologies.

I will collect user data in this study to develop a new Information Architecture and Visual Hierarchy that aligns with the users' mental models.

Following these two primary design components, a new structure and navigation are defined. User interfaces and a prototype are developed.

Outputs will be tested with the end users and compared to the design generated solely based on business requirements to evaluate the quality of the outcomes and the validity of the employed methodology as far as possible.

4.1. New Design: alignment with the business context

This section briefly presents the first redesign work conducted for the QuantyxRM software. This study was carried out to understand the business domain and the specific business requirements.

Firstly, with the assistance of the quantitative analysis team, I sought to understand the specific activities for which the software was designed.

We identified all distinct functionalities. Whether these were applicable depended on

the type of financial data being processed. From our analysis, and consistent with findings from similar financial software systems, such as Bloomberg Terminal, we discerned that two primary variables played a pivotal role - the specific financial data and the desired function to be applied to that data [65]. The software's response is determined based on these two user-selected variables.

Below, I delineate the characteristics of these two variables and expound upon their properties.

Financial Data pertains to financial elements employed by users for work purposes. These elements are organized hierarchically, following the "Client then Asset Management Company (AMC) then Fund then Asset". The Entity-Relationship diagram is presented in Appendix **??** to clarify this structure.

Function refers to the action that the user intends to perform. The functionalities available to the user vary according to the selected data.

After understanding the primary business requirements and how they impact the user, we analyzed how this was represented in the current software. Regrettably, there was no clear distinction between data selection and the action to be performed on that data. Some functions can be accessed through a wizard from a collapsible sidebar, shown in Figure 4.1.

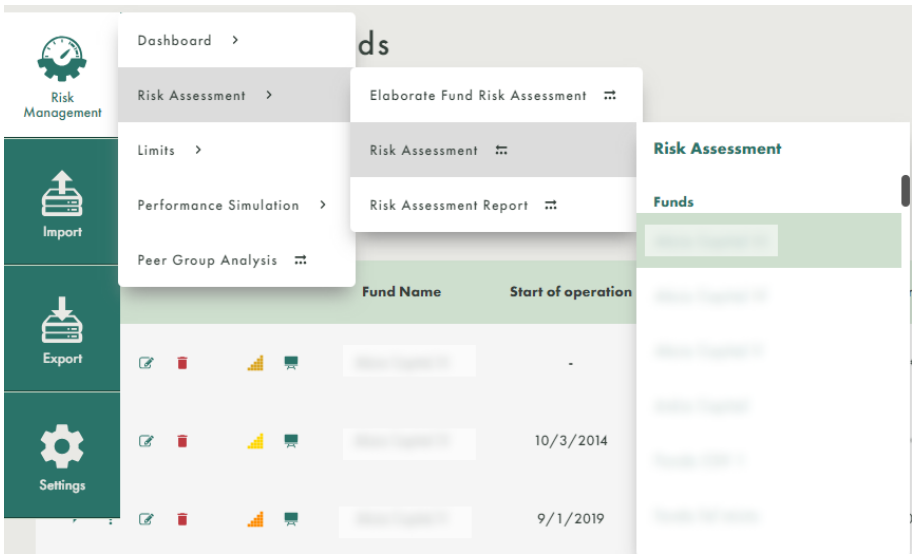


Figure 4.1: Wizard for fast access to main software functions in the actual QuantyxRM design

The remaining functions are available from a dropdown menu located on the line of the specific financial element. This menu selects functionalities and data navigation without clear differentiation, as illustrated in Figure 4.2.

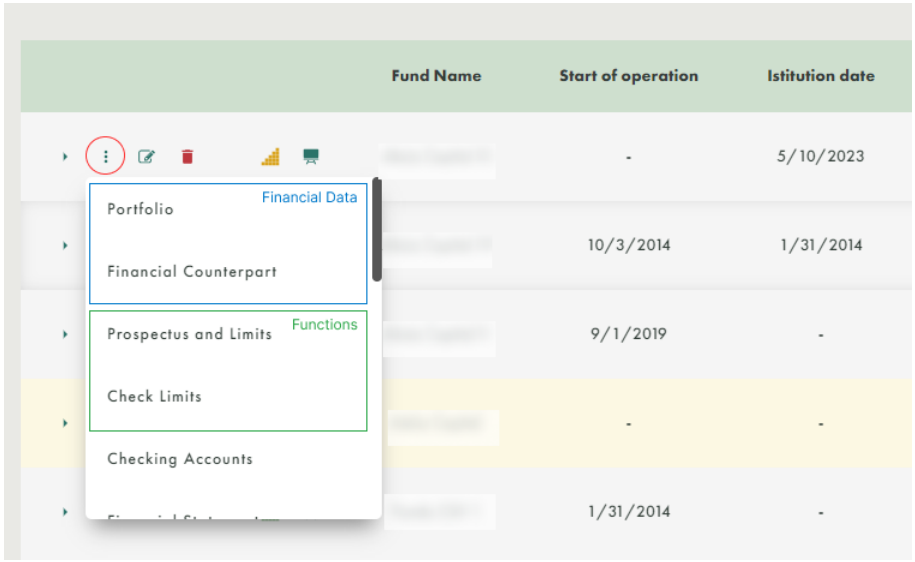


Figure 4.2: Main component for navigation in the actual QuantyxRM design

Upon realizing that the design was misaligned with the business context, we undertook a redesign to address this primary issue. The initial redesign effort involved separating data selection from the choice of function to be performed on the data. A process that had already significantly simplified the software navigation.

This process was carried out with the Product Owner, ensuring the new design aligned with business requirements. While the first redesign process will not be detailed in this thesis, it is the foundational design upon which the user-centric study is applied.

4.2. User-Centered Design Methods

Four User-Centered Methods were selected - two for collecting user requirements and two for testing. These methods are detailed in Chapter 2, while their implementation and the results gathered are presented in Chapter 5.

I employed the Card Sorting method to collect user requirements, specifically to understand the users' mental models. This was done to define the Information Architecture, mainly related to restructuring the pages' organisation and improving software navigation. I will utilise the tree-testing method to test and better define the obtained IA.

A survey was conducted to understand the essential information and how to accurately represent it with the appropriate visual hierarchy on pages.

Finally, I will employ the Qualitative Usability Test to evaluate user interaction with the new interfaces and use an A/B test to compare the business design to the new one.

5 | Results

In this chapter, I present the results of the methods employed to develop the new design, user studies, and testing.

I detail how these methods were applied in a real-world case study. We also describe the methodologies used to obtain the results and explain their rationale.

5.1. Card Sorting

In the conducted Card Sorting study, all included participants are company employees and use the software for work-related purposes. Every participant completed the Card Sorting task, and all 29 cards were consistently included in a cluster. Of these partecipants, 40 per cent are enterprise users with moderate to low proficiency in the company software, while 60 per cent are enterprise users who are highly skilled in using it.

Due to the constraints of the free version of the software tool used and the extended time required to recruit the necessary participants, the results include data from only 10 participants out of a total of 15.

For the analysis of the Card Sorting results, three methods were selected and identified as the most effective for Open Card Sorting with a low number of participants. The Best Merge Method (BMM) dendrogram, the Similarity Matrix, and Participant-Centric Analysis (PCA) are these methods. [1, 46]

5.1.1. Partecipant-Centric Analysis

PCA is closely related to the concept of Information Architecture. Specifically, PCA reveals the most popular IA configurations based on participant choices. Essentially, it identifies clusters of cards that receive the highest level of acceptance from participants. This method is particularly effective in cases where the number of cards is small, i.e. less than 30. [1, 46]

A category is selected if it exhibits more than 50% similarities with categories created by other participants. This method serves as a valuable foundation for identifying categories but does not account for labels. [1]

5.1.2. Best Merge Method dendogram

The BBM dendrogram is the most effective method for understanding clustering patterns when dealing with a few participants. The BMM breaks down each instance of a category from each participant down to its base pairs. The pair that scores the highest is then confirmed or "locked in." This process is repeated. If a newly locked-in pair intersects with a previously locked category, it merges with that category. Any smaller groups within this new, larger category are then removed. The scores generated by BMM indicate that X% of participants concur with elements of this particular grouping. [1, 46]
Figure 5.1 shows the BMM dendrogram obtained in this study.

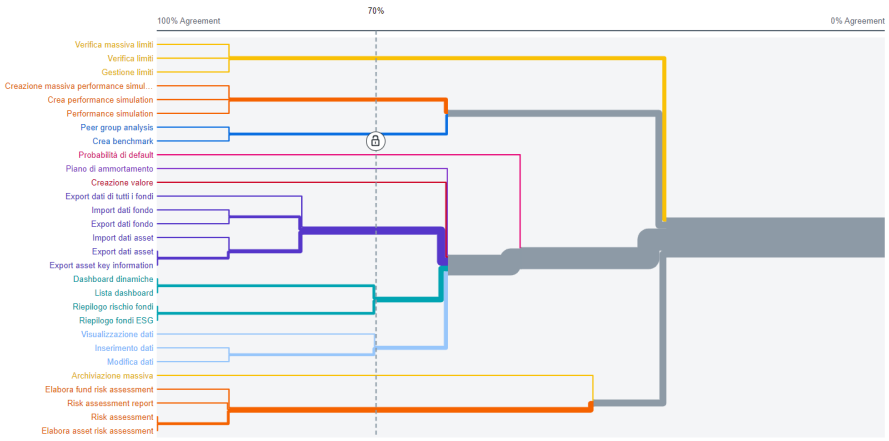


Figure 5.1: Best Merge Method dendrogram of the Open Card Sorting study [1]

5.1.3. Similarity Matrix

The similarity matrix provides information on the percentage to which two specific cards have been grouped within the same cluster. A higher percentage indicates that more participants have placed them in the same cluster. [1]

Figure 5.2 shows the similarity matrix obtained in this study.

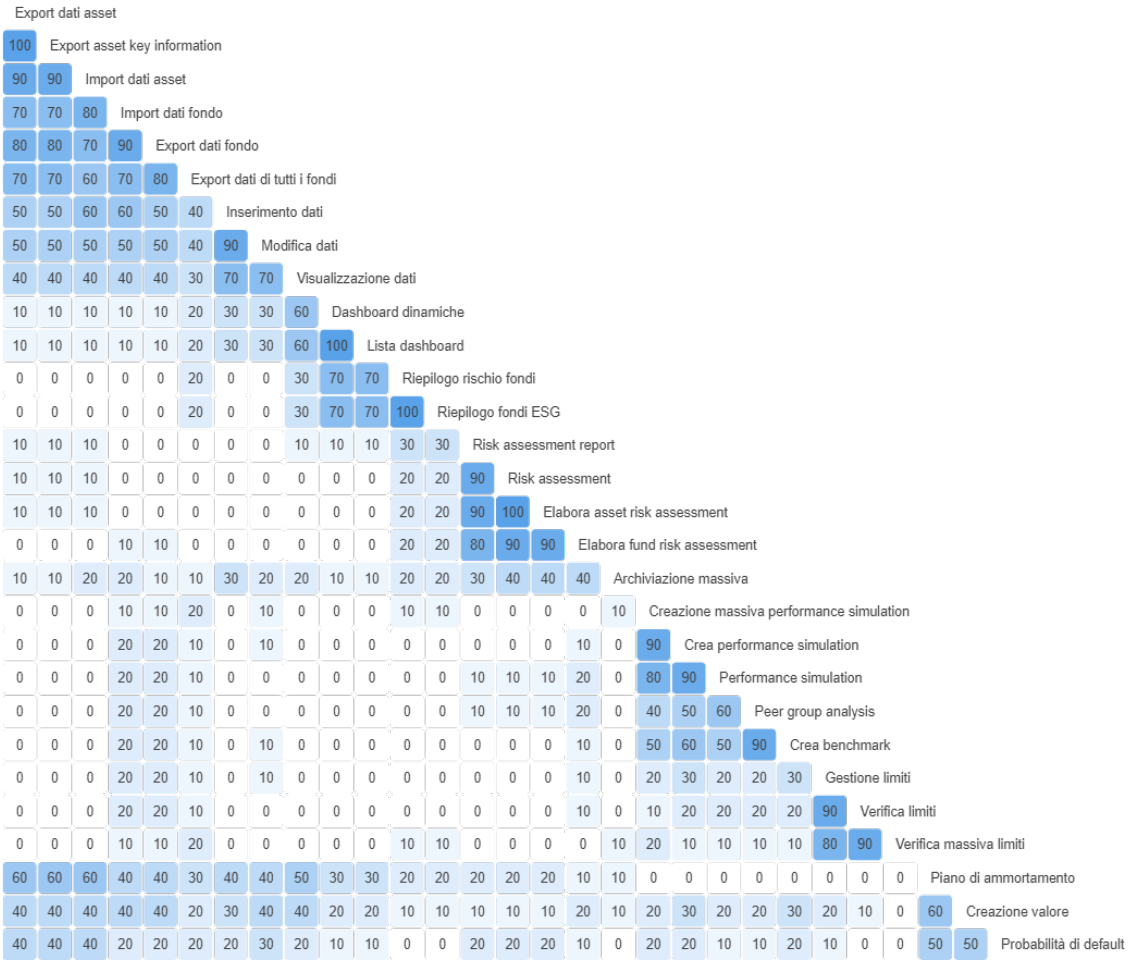


Figure 5.2: Similarity Matrix of the Open Card Sorting study [1]

5.1.4. Chosen Clusters and Labels

In the decision-making process for the card sorting results, the first step was to verify whether PCA produced highly acceptable outcomes. In this case, PCA identified an Information Architecture where 7 out of 10 participants grouped the same cards at least 50% of the time.

The Best Merge Method starts from the Information Architecture identified through the PCA. With the BMM, the acceptance percentage of each group of the selected Information Architecture was verified, and the groups with an acceptance percentage higher than 70 per cent were confirmed.

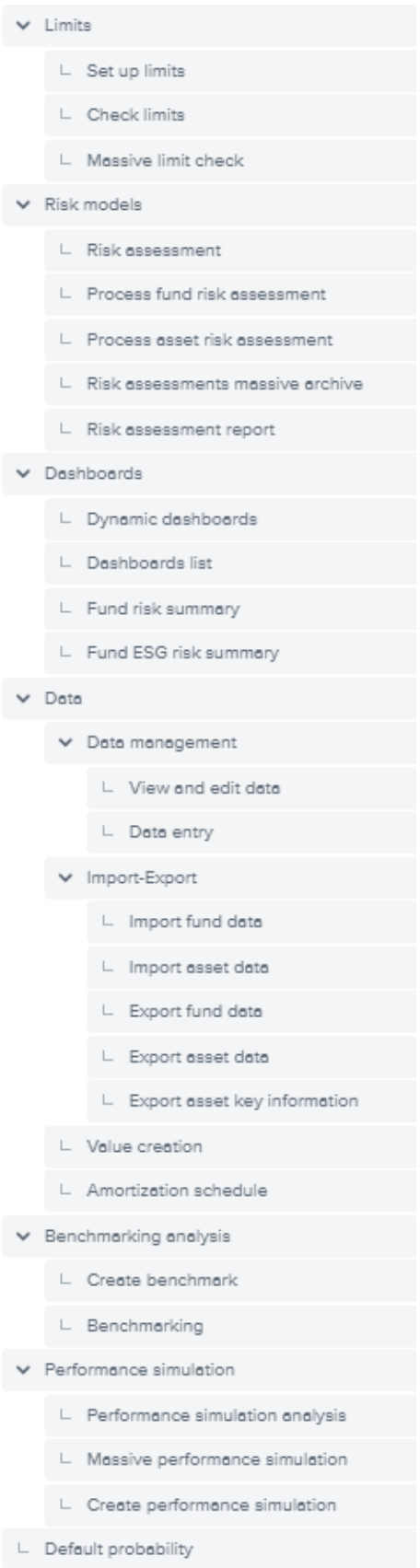
Once the valid groups were confirmed, the excluded cards were evaluated using the similarity matrix. If the cards paired by PCA did not achieve at least a 70% match rate in the

similarity matrix, the group was adjusted based on the results from the BMM dendrogram.

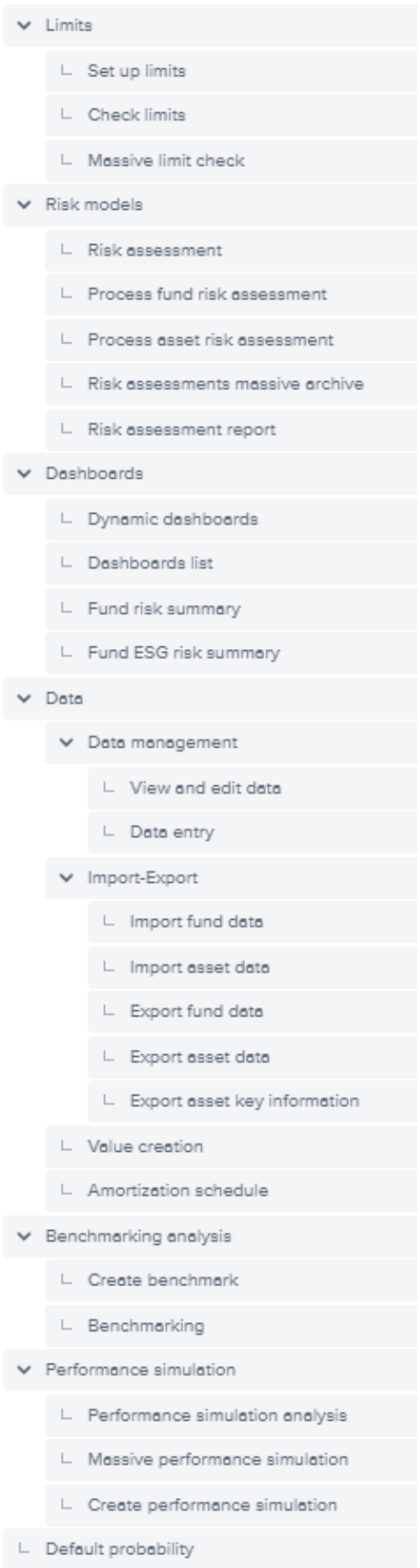
Upon completing this process, I identified ten distinct clusters with a reliability exceeding 70%. This number can be reduced to seven clusters with a reliability of at least 60%. In the first scenario, the main navigation menu will display ten clusters. In contrast, the second scenario will show seven clusters, one containing two large internal sub-clusters. Both of these configurations will be evaluated using a tree test. Given the high acceptance rates achieved in my study of card sorting analysis methodologies, I anticipate that both configurations will yield high completion rates. The primary question is to determine which of the two configurations is superior.

The second crucial aspect involves the labelling of cards and groups. Participants were asked to specify which labels they found challenging to understand for the cards and why. For the groups, an analysis was conducted on the terminology used by users to identify groups based on similarity. Labels were then chosen that demonstrated the greatest inclusivity relative to those provided by the participants.

In Figure 5.3, the trees obtained through the described procedure are presented.



((a)) Card Sorting result with 60% acceptance rate



((b)) Card Sorting result with 70% acceptance rate

Figure 5.3: Card Sorting results

5.2. Tree Testing

The Tree Testing method is employed to assess the outcomes of Card Sorting. Specifically, two Tree Tests were conducted.

The primary reason for conducting two Tree Tests was to determine which of the two structures identified during the Card Sorting was more favourably received by users. This testing method, also known as A/B Tree Testing, was utilized to evaluate labels that had yielded ambiguous and problematic results during the Card Sorting. [66]

5.2.1. Participants and Tasks

In the Tree Test study, there were a total of 11 participants. Participants cannot undertake both tests for studies of this nature, but only one of the two [66]. In this instance, 6 participants were randomly assigned to the first Tree Test, while the remaining 5 were assigned to the second.

To ensure the reliability of this study, participants were individually contacted after verifying their eligibility. Specifically, participation required advanced knowledge and experience in the business domain of alternative investments. It was essential that participants were familiar with the processes described in the tasks; otherwise, their responses could not be considered valid.

In the study, all participants were presented with seven tasks. After completing each task, participants were prompted with a post-task question, inquiring whether they faced any difficulties during the execution. This post-task question gathers qualitative data, which would be paired with quantitative data to provide a more comprehensive analysis. However, if there were highly divergent results from the task, participants were asked if they would be willing to be re-contacted to provide further information on this behaviour. It should be noted that this secondary option was never employed, as no participants provided inconsistent responses.

The seven tasks were consistent across the first and the second Tree Tests. Specifically, these tasks were employed to identify two distinct types of information. They were used to ascertain the optimal labels when the Card Sorting did not yield satisfactory insights. Different labels with identical paths were used in the two trees in such instances, as illustrated in Figure 5.4.

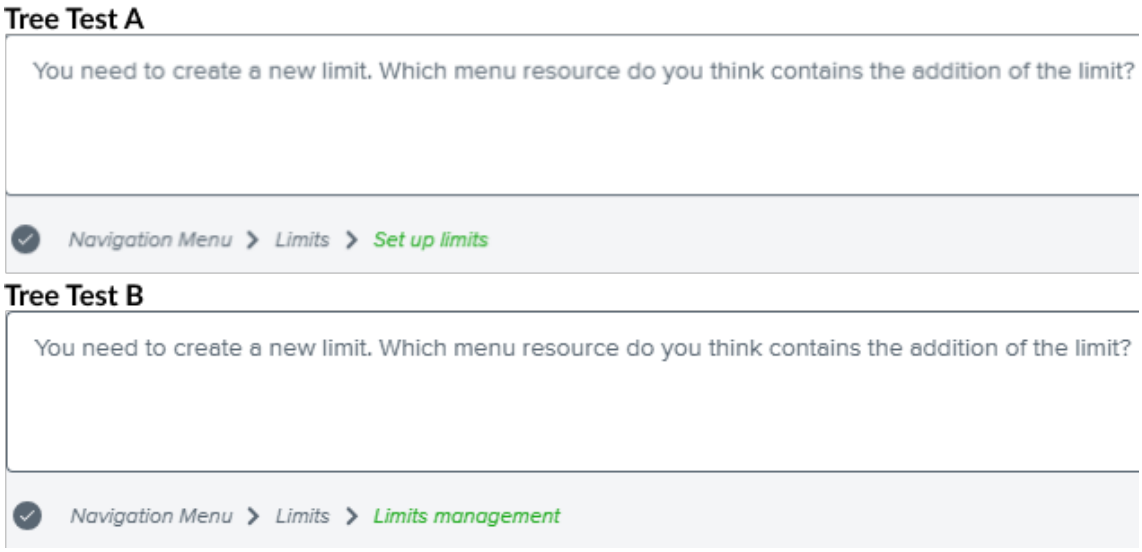


Figure 5.4: A/B task about labels [2]

The second category aimed to understand which groupings of trees were most favoured by users. On this specific aspect, various analytical methods were employed to achieve a satisfactory solution.

5.2.2. Metrics

Various metrics can be employed to analyze a Tree Test derived from the data collected during the test. The selection of these metrics depends on the type of task and the specific objective to achieve through that task [66].

Before establishing the metrics, it is essential to define the potential outcomes of a task. There are three possible outcomes - success, failure, and skip. A "success" indicates that the task was answered correctly, "fail" implies an incorrect answer, and "skip" means no answer was provided. Each of these outcomes can be achieved either directly or indirectly. A direct outcome signifies that actions were performed following a single path. In contrast, an indirect one indicates that the participant navigated through multiple branches of the tree and, therefore, revisited previous points at least once before answering or skipping the task [2].

Based on these definitions, I first introduce the two primary metrics that provide the most crucial information regarding task completion and, consequently, the results of the Tree Test. These are the *success rate* and the *directness*. [2, 66]

The success rate for that task represents the proportion of users who accurately identified the correct location in the tree and recognized it as the appropriate spot to finish that task.

Directness indicates the number of users who navigated straight to the correct answer without backtracking or switching categories. This seamless navigation type is often called the 'happy path' because it implies an effortless interaction with little to no distractions or missteps. [66]

In my primary analysis metric for evaluating results, I employed an algorithm that calculates the weighted average between the squares of the success scores and directness, favouring success with a 3:1 ratio, and scaled to be a value out of ten [2]. This metric will be called the *Task Score*.

In analysing the results, the First Click, the Paths and the Time Taken were other crucial metrics employed.
The First Click indicates whether the first selection made by participants was on the right track towards the intended destination. This helps assess the clarity of the top-tier labels and how effectively they suggest the correct route based on the task. If a large percentage is on the right path, the top-level labels are well-defined for the given task.
The Paths table displays the routes individuals took through the tree for every task, indicating whether participants reached the correct destination directly or took a detour.
The Time Taken is the median time it took participants to finish the task represented by the line in the centre of the light blue box, measured in seconds. Using this 'duration metric' is valuable during A/B testing, as it distinctly indicates if participants take more time navigating one tree over another.

Various metrics can be applied when evaluating Tree Tests. However, in this study, I focused solely on the metrics pertinent to an investigation with limited users. Had the Tree Test involved a larger user base, it would have undoubtedly necessitated additional metrics and a shift towards quantitative rather than qualitative analysis, as seen in this instance. [66]

5.2.3. Definitive Structure

To determine the final structure, objectives related to each task were considered. Based on these objectives, results from Tree Test A were compared with those from Tree Test B.

I initially compared Task A's Task Score with Task B's. A Task Score is deemed acceptable if it exceeds 7; otherwise, the specific task undergoes more detailed analysis. Regarding issues arising from Card Sorting misalignments, there were no problematic tasks; tasks with the best label or position all scored above 7.

In tasks done for understanding the structural differences between categories in Tree A and Tree B, I observed moderate to low scores that were relatively inconsistent. This necessitated a more detailed examination of participant behaviour.

To analyse the optimal category structure, I focused on the two metrics that provided the most insight in an A/B Test - the Task Score and the Time Taken. This approach allowed for a more concentrated examination of the deeper metrics within a single Tree Test.

Tree Test A had an average time of 47.34 seconds per task and an average task score of 6. Conversely, Tree Test B recorded an average time of 43.68 seconds per task with an identical average task score of 6. However, it's crucial to understand the distribution of tasks. For instance, in the first tree, the initial task showed excellent results, while subsequent tasks experienced a decline, with average times reaching up to one minute and scores as low as 2, with 0 directness. In contrast, in the second tree, where information was more structured, after a less efficient initial task (though with a score of 9), subsequent tasks displayed significant improvement in the Time Taken compared to the first. Given these insights, I found it beneficial to repeat the analysis, excluding Task 1.

Excluding Task 1, Tree Test A had an average time taken of 54.97 seconds per task and an average task score of 4. On the other hand, Tree Test B had an average time of 30.52 seconds per task, with an average task score of 4.5. Based on these results, I began the analysis with Tree Test B, which proved more efficient after the first navigation.

Two main issues emerged when analyzing the metrics 'First Click' and 'Paths' on a participant-by-participant basis and comparing them with comments from the post-task questions. The first concern was related to the labels and the second was about the data.

Specifically, 90% of the First Clicks were accurate, yet the success rate was around 70%, while the directness was below 50%. This suggests a discrepancy in the labels [66]. Upon further examination of the paths, it was observed that even though users initially entered the correct branch, most would exit, try a different path, and the majority would eventually return to the correct one and find the solution.

In examining the participants' feedback, it became evident that not only were the labels unclear, but there was also a need to augment the navigation with a component allowing for selecting the data source and the specific functionality.
The resulting structure is depicted in Figure 5.5. It is supplemented with a secondary nav-

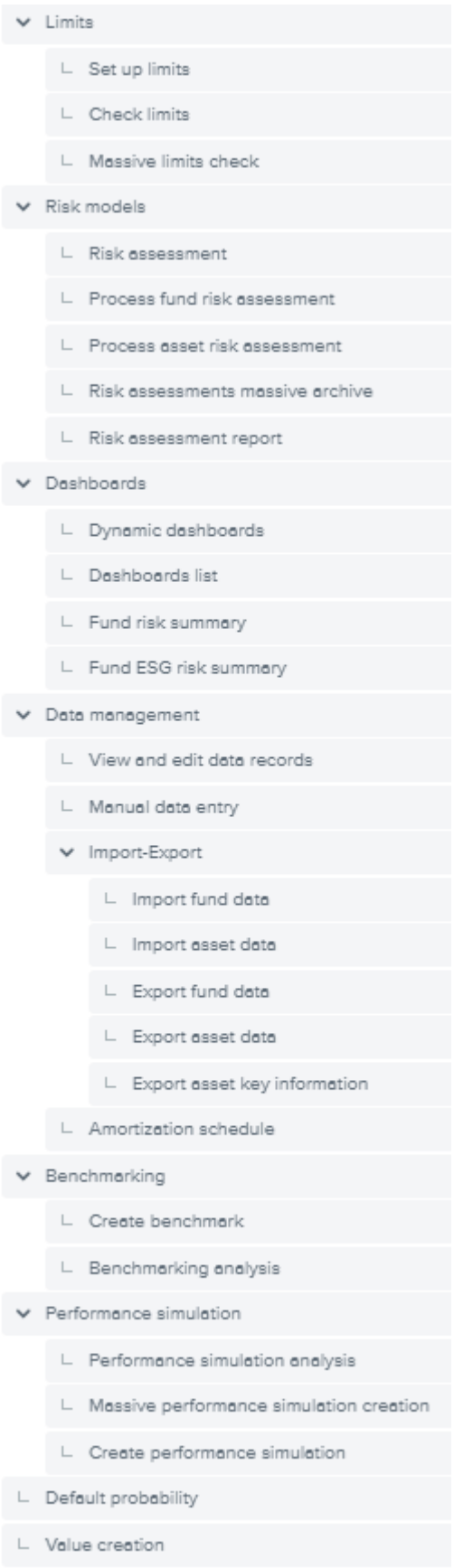


Figure 5.5: Definitive structure obtained after Tree Testing analysis

igation feature, allowing users to choose the type of data they wish to apply the selected functionality. This secondary navigation is dictated by the business requirements, which specify the data types for specific actions. However, user feedback indicated a preference for immediate data selection. Consequently, the design will address this need by seeking a solution to accommodate this preference.

5.3. Survey

The survey conducted in this thesis understands the hierarchy by which enterprise users of the QuantyxRM software identify information. The main functionalities, Risk Assessment, Performance Simulation, and Peer Group Analysis, were selected for investigation. Key components were identified within these functionalities, and participants were asked to rank them in order of importance using ranking questions.

In addition, two multiple-choice questions were included in the study, where participants were asked to select the features most important to them for display on data visualization pages. Specifically, the questions focused on the Home Page and Table View preferences.

The survey was administered to twelve participants, but four had to be excluded as they did not meet the minimum eligibility criteria detailed in Table 5.1.

Inclusion Criteria
Minimum education Master's degree in finance
Work within the financial sector for at least 3 years
Who has been working within the company for at least a year
Medium to a high level of knowledge of QuantyxRM software

Table 5.1: Criteria for inclusion of survey participants

The survey consists of questions that, to be answered effectively and avoid the biases discussed in Chapter 2, required a high level of knowledge and understanding of business domain topics. These criteria were selected to ensure the reliability of the results; only specialized users with extensive financial expertise in the business domain were evaluated.

5.3.1. Ranking Questions

The outcomes of the ranking questions were determined based on the average position and the standard deviation of each page content. To proceed, participants were required to rank all the content items.

The standard deviation is assessed when the average position has a similar value for two or more pieces of content. The standard deviation serves as an indicator of how much the variable deviates from the average position value. Specifically, if the standard deviation is high, it suggests that even though the content holds a similar average position, there is a high degree of variability in how users perceive it. Consequently, I have chosen to give the contested position to the content with the lowest standard deviation..

Based on the results obtained from comparing the average position and standard deviation, design solutions that adhere to the principles of visual hierarchy for content presentation on the page have been developed. These designs will be integrated into high-fidelity prototypes for testing.

5.3.2. Multiple-choice Questions

In the multiple-choice questions, participants were presented with various features to be implemented for navigating pages and viewing and editing data within the software.

These were selected for inclusion in the new design based on the frequency with which survey participants chose them. Additionally, participants had the option to add further suggestions. However, only one participant provided input, suggesting adding a global search feature within the software.

Only features with an acceptance rate greater than 50% were selected in the design. However, it is crucial to note that there was a significant discrepancy among the options. Specifically, all included features had a frequency rate of over 60%, while, except one out of nine, the excluded features had a frequency rate of less than 25%.

5.4. New Design with UCD approach

Following my research, I developed a new Information Architecture, and the user interfaces were redesigned.

The tool used for their representation is discussed in Chapter 2. This encompasses two interconnected diagrams. Combined, they comprehensively represent the Information Architecture and the page design, excluding the search component and including visual hierarchy and page interactions.

5.4.1. Blueprint diagram

The first diagram was constructed based on the Visual Vocabulary outlined by Jesse James. This framework presents the navigation and organization of pages grouped into distinct categories.

The entirety of the diagram cannot be displayed on a standard sheet. Therefore, it is segmented into multiple sections, which are interconnected using a specific component called the "continuation point".

The first section depicted in Figure 5.6 represents the login, which leads directly to the initial home page.

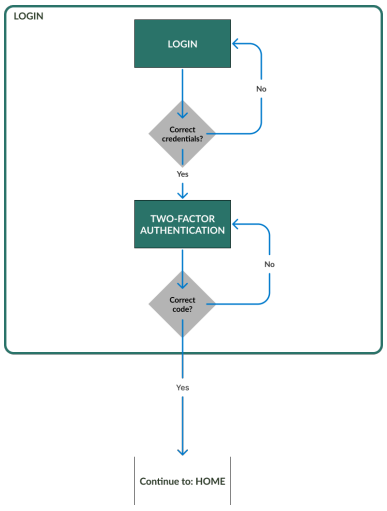


Figure 5.6: Information Architecture Login

After accessing the homepage, it is possible to navigate all the pages and sections corresponding to the IA identified from the Tree Test.

Before introducing the next section, it's important to note that arrows do not represent a unidirectional flow. Instead, they indicate a flow that can be traversed in reverse unless specifically represented by blocking arrows. These blocking arrows cannot be traversed

backwards. However, no such blocking arrows are present in the navigation described here.

Another crucial aspect to address is the operation of the "conditional selector" component, which acts as a filter. In the current diagram, it's used in a scalar manner to illustrate how users navigate the data flow. Certain pages are displayed while others are not, depending on the specific data point they are at. It is also important to note that the chosen design for navigation allows access to pages linked to the "previous" or parent data but restricts access to pages linked to the "subsequent" or child data. This design choice is consistently represented with bidirectional arrows in the visualization.

Figure 5.7 depicts the data filtering process. Depending on the specific data the user selects, access to certain pages may or may not be granted. Different colours have been used for the data, and the arrows emanating from its filter facilitate comprehension.

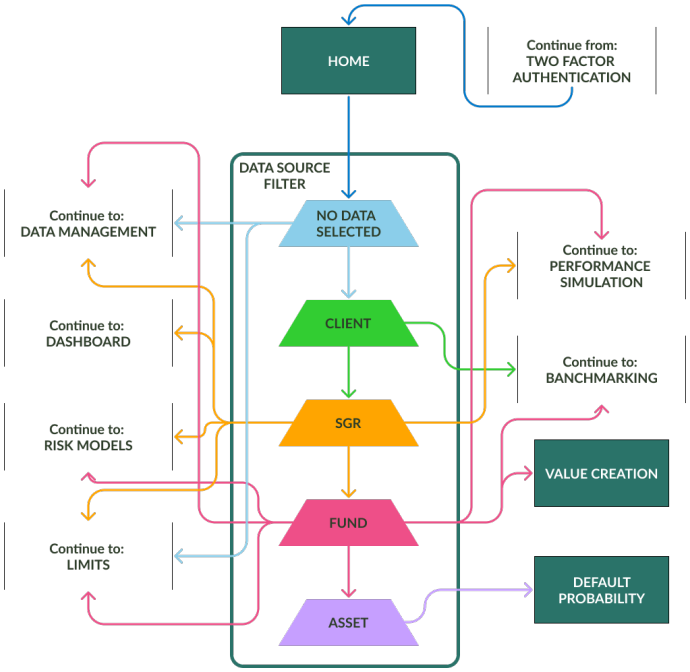


Figure 5.7: Information Architecture Data Filters

In the subsequent figures, the various pages corresponding to the groups identified through user studies are presented.
Figure 5.8 illustrates the Data Management section. This section delineates all possible data-related actions, including insertion, modification, viewing, and bulk import/export.

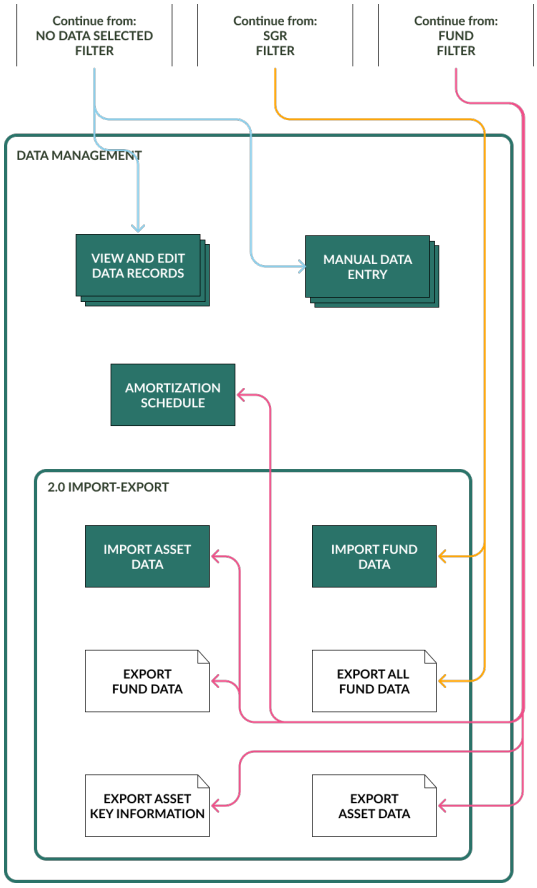


Figure 5.8: Information Architecture Data Management

In Figure 5.9, I present the section dedicated to customized data visualization, namely the dashboards.

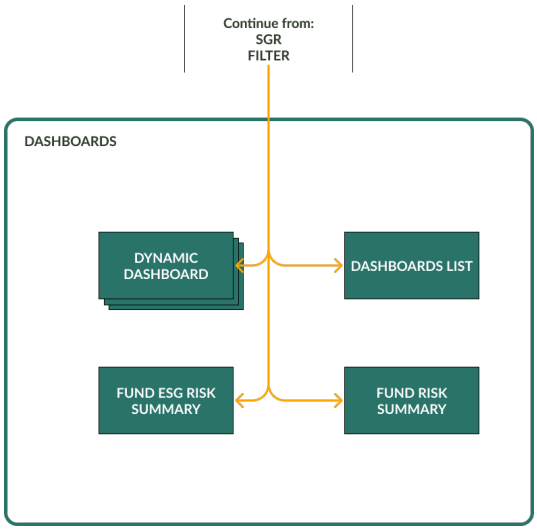


Figure 5.9: Information Architecture Dashboard

In Figure 5.10, the section about the automatic calculation functionality is presented, representing the software's primary service: the Risk Model.

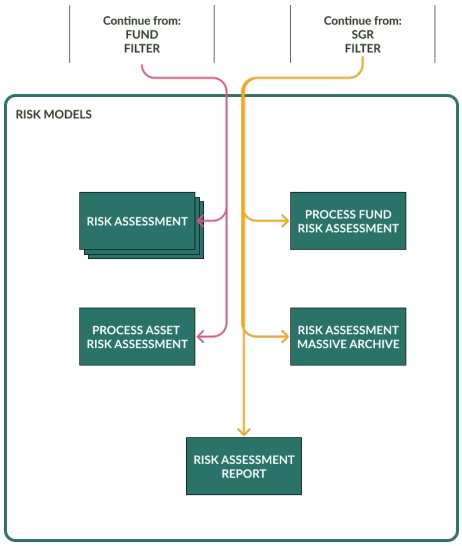


Figure 5.10: Information Architecture Risk Models

In Figure 5.11, the section about another software feature, the "Limits Management," is presented.

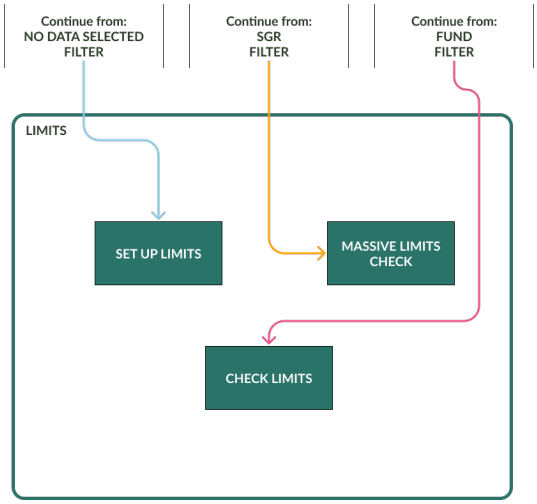


Figure 5.11: Information Architecture Limits

In Figure 5.12, I present the section about another software feature - Performance Simulation.

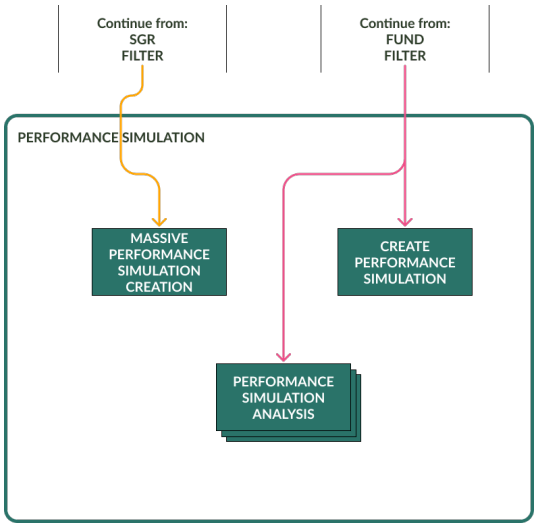


Figure 5.12: Information Architecture Performance Simulation

In Figure 5.13, we present the section about another feature of the software - Benchmarking.

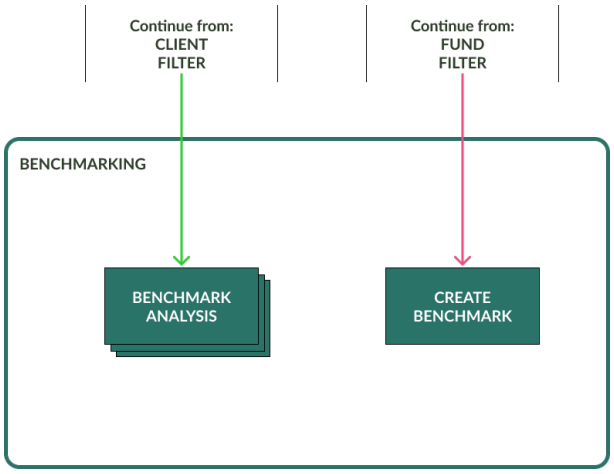


Figure 5.13: Information Architecture Benchmarking

5.4.2. Wireflows

Wireflows are diagrams that merge traditional flowcharts with interface design and can be presented at varying levels of detail. In this study, given that these tools will be used by the development team for implementation, I employed high-fidelity representations. These were subsequently used in the interactive prototype for user testing.

Wireflows have a minimum 1:1 ratio regarding the number of elements (pages) embedded compared to the blueprint diagram. This means that every page in the blueprint diagram corresponds to at least one page in the Wireflows.

The approach to managing interactions is subjective and depends on the work's desired presentation and the type of software designed. For example, I have added labels to the arrows to elucidate various interaction typologies. These labels specify the outcome of each particular action, reducing the number of required interfaces and allowing content sharing.

Wireflows describe software navigation regarding the specific component of the page that allows such navigation, such as a button. These components are typically colour-coded to immediately tell the reader which part is causing movement in the flow. I chose to represent them in red.

For clarity, given the vast amount of data and interactions, I provide an example in

Figure 5.14 of the actual Wireflows presentation.

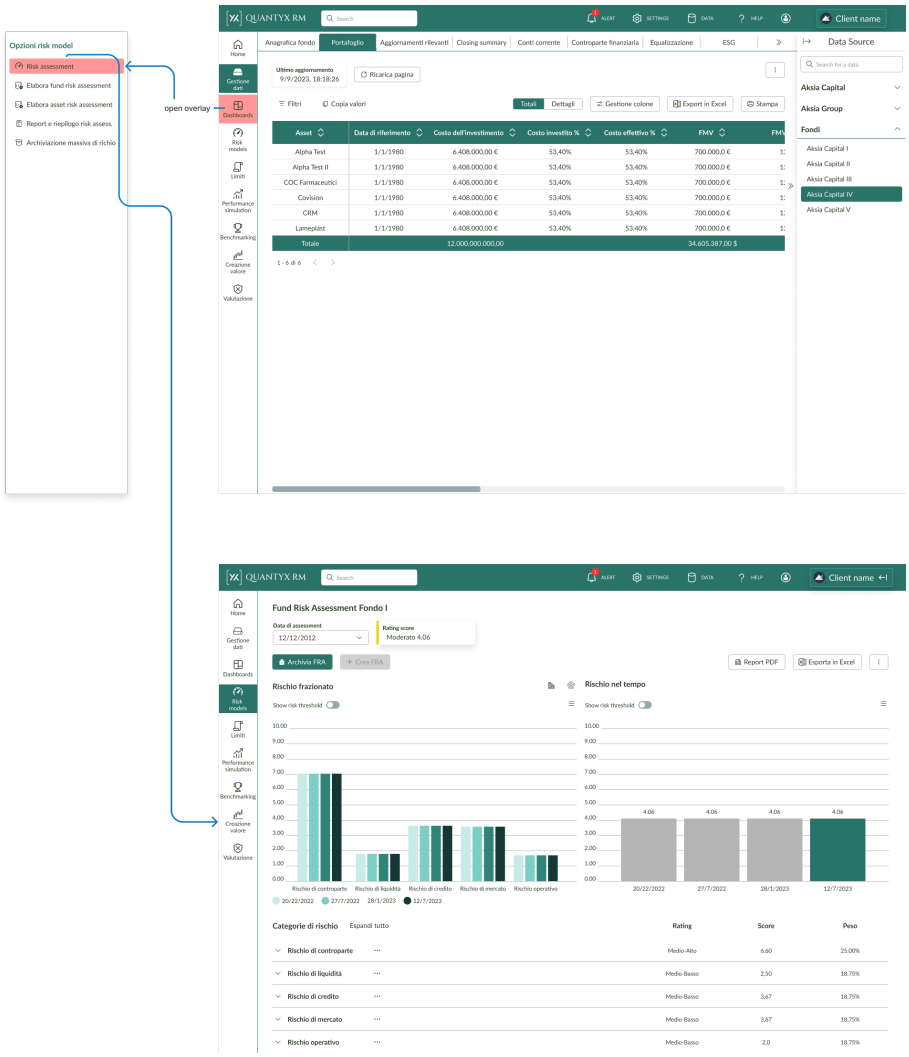


Figure 5.14: Snapshot of Wireflows

This figure illustrates a section of the Wireflows, explicitly depicting the navigation between the record display page and the software's primary function, the Risk Assessment.

Furthermore, in addition to illustrating the specific element of the page that triggers navigation, Wireflows also depict the page's structure and how information and content are organized.

In Figure 5.15, I present a high-fidelity wireframe as an example of the type of wireframe integrated into Wireflows. I use this image to delineate the design of the system's three main navigational elements.
The first element, highlighted in red, represents the main navigation, facilitating access

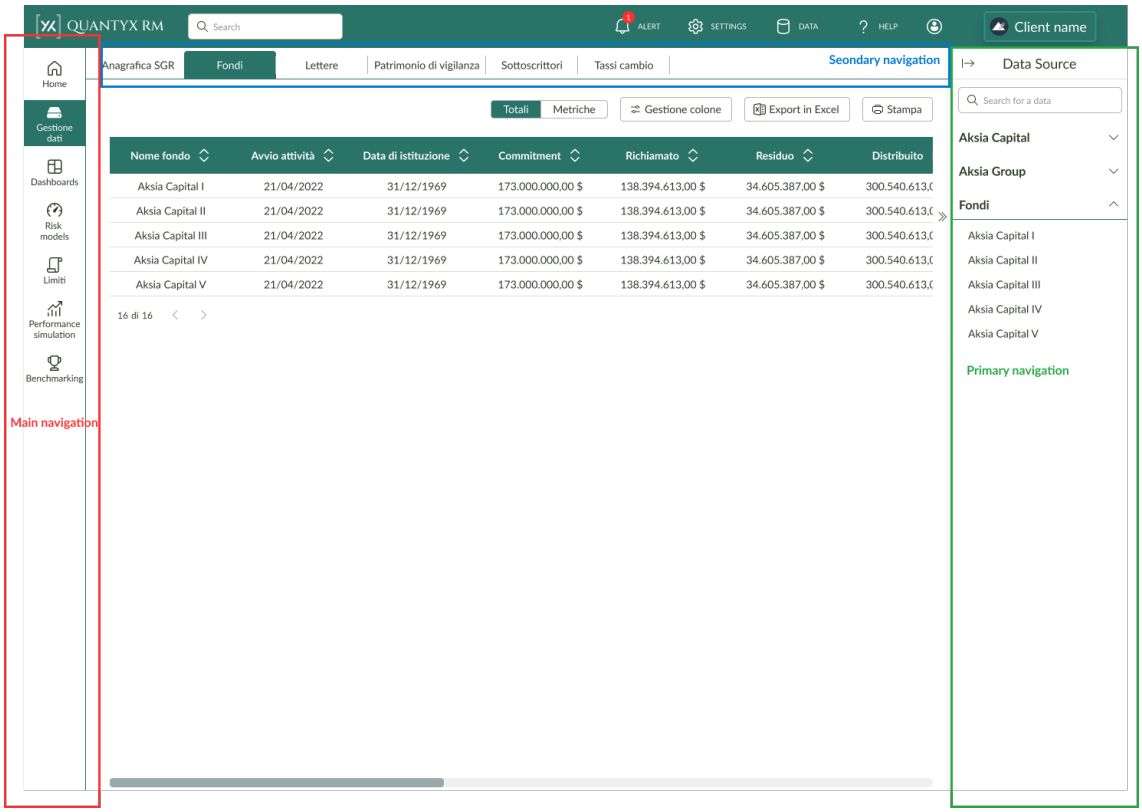


Figure 5.15: High Fidelity Wireflow with design of new navigation structure

to the software's core functionalities. The second, in green, pertains to primary data navigation, enabling movement between various data types in the system. This changes the interface, adjusting the available functionalities and the types of secondary data that can be accessed. Lastly, the secondary navigation in blue grants access to specific data linked to the selected primary data.

The navigation design and structure are informed by the insights gleaned from survey responses and a domain-specific study in the business context for which the software is designed.

5.5. Usability Test

The usability test conducted gathers qualitative data from actual software users. Specifically, the test was carried out to achieve two distinct outcomes. First, it compares the initial design derived solely from business requirements (A) with the design achieved us-

ing the User-Centered Design approach (B). Secondly, it aimed to identify any issues and gather insights into potential requirements that might have been overlooked during the initial user data collection phase.

5.5.1. Partecipants

For the usability test, five participants were selected. They were given a one-week window to complete the test, conducted in person, with my role as a facilitator. While all five participants confirmed their availability, only two actively participated in the test. Due to time constraints, the remaining participants provided feedback solely on the A/B comparison of the two designs.

It's important to note that both participants in the actual test are relatively inexperienced in using the platform but possess strong knowledge in the business domain.

5.5.2. Tasks

In the study, three tasks were assigned. The first task focused on validating the navigation across different pages of the platform and on data input. The second task dealt with data visualization elements via the table component, which was redesigned based on survey results. Lastly, the most complex task aimed to test data navigation and its associated features in conjunction with the newly defined visual hierarchy for the primary functions.

5.5.3. Some Quantitative Results

Despite their limited validity, I gathered quantitative data regarding the completion rate and time to complete the three tasks.

A - Tasks Success Rate: 50%, Average Time to Complete: 2min 48sec

B - Tasks Success Rate: 83%, Average Time to Complete: 1min 32sec

I can confirm that, while the data are limited and not highly significant, it already suggests that the design developed using user-centric methodologies has led to improvements. Specifically, considering that participants struggled to use the software designed solely based on business requirements, their feedback provides potentially valuable insights when comparing designs for new users.

5.5.4. Qualitative Results

Turning to the qualitative data section, all participants who compared the two interfaces unequivocally chose Interface B, developed using the UCD approach. Notably, feedback from long-term users affirmed their preference for the new interface, highlighting its improved comprehensibility and organized layout. They generally found it "cleaner" regarding how hierarchies and distinctions between functionalities and various data types were presented.

Regarding participant feedback on the tasks, at the end of each task, participants were asked about any challenges they encountered and, if they did not complete the task, the reason they did not find a solution.

The findings concerning issues were limited. Notably, a specific element within the table was expected and necessary for data comparison. The survey missed this element because it wasn't included in the analysis phase. Importantly, no survey participants had pointed out this omission. Regarding unresolved tasks, I realized how I managed the interaction to indicate that the tables had additional columns was not clear enough and needed revision.

Regarding my observations, I understood an excellent insight regarding the main navigation component, designed for data traversal, that was not present in the A design. The feature in question emerged from Tree Testing feedback. Interestingly, it was never addressed during the Usability Test, neither by the test participants nor those reviewing the interface. Upon inquiry, users indicated they perceived it as a visual component to understand positioning rather than an interactive navigation tool.

There was no other significant feedback. It can be noted that among the participants who completed the tasks and those who merely compared the two interfaces, none found difficulty discerning the interface's potential uses. Additionally, they did not feel disoriented in understanding or searching for specific features or components. Although the tasks associated with the new interface were reasonably complex, especially the last one, the interfaces and navigation appeared well-designed to not pose any real challenges for enterprise users.

6 | Discussion

In this chapter, I present my reflections on the research conducted in this thesis and describe the study's limitations and potential directions for future work.

I begin by discussing a variant of the UCD approach, as defined and used in this thesis. While it is not without flaws, the findings from the literature review have helped me understand how user-centric approaches, processes, and methodologies are applied and effective within enterprise software. Nevertheless, this is a highly debated topic, as some argue that these two areas may not be entirely compatible. I understood the underlying reasons for this viewpoint. The approach I have outlined remains highly beneficial for businesses because it primarily focuses on aspects that simplify usage. It does not seek to redefine the foundational elements, which, for now, are dictated by the business domain rather than the user's perspective.

It was crucial to understand this distinction. The modified User-Centered Design approach I have adopted specifically focuses on this point. In such systems, the differentiation between business and user requirements is subtle, especially when discussing the system's functional aspects.

As mentioned, this study focuses on enhancing software by addressing its complexity and usability. These two aspects are crucial for such specialized software, as they often exhibit a level of complexity significantly higher than average consumer software. Consider, for example, the difference between an Enterprise Resource Planning (ERP) system and an e-commerce platform.

Another observation during this study was that the software used as a case study exhibited limited adaptability to the business domain, irrespective of the user-centric study. This inadequacy introduced biases among the end users, who were company employees. It became crucial to develop externally from the old system, focusing solely on the software's objectives and services.

Employing methodologies that did not introduce bias was crucial for the user data collection phase. Users familiar with the software often adjusted their feedback based on their knowledge of the existing software they used extensively each week. Therefore, it is essential to approach users with well-framed questions and appropriate research methods, seeking insights on the relevant business domain without directly referencing the software's specific functionalities. This approach ensures the acquisition of genuine responses.

6.1. Limitations of the study

The limitations of this study are substantial, making it challenging to draw definitive conclusions from the results obtained.

The primary limitation was time. I dedicated a significant amount of time to completing the project and thesis. Nonetheless, this duration is not even remotely close to what would be required for a comprehensive understanding of this topic.

First and foremost, this research utilized a single case study, indicating that the results apply to this specific instance. However, they may not necessarily be generalizable to other enterprise software, even of the same domain.

Furthermore, my approach was designed to elucidate the clear distinction between a User-Centered Design approach applied to consumer software and the thought process to adopt for enterprise software. However, this does not exclude the possibility of other variations. Moreover, using the standard UCD approach might lead to better results. The fact that I did not use it represents a limitation. Given the assumptions based on research, even the standard UCD, for instance, could lead to different outcomes, which this thesis does not expose.

The most significant limitation pertains to the number of participants. In user studies, tests, and UX research methodologies employed, despite ample time and availability of numerous users, only a few participated. Furthermore, an even smaller subset were the appropriate target users for the specific study.

While the results of this study are qualitatively valid, it would have been highly beneficial to gather quantitative data. This research is well-suited for quantitative analyses. By comparing two different designs of the same system, both for user data collection

and testing results, comprehensive quantitative data would have offered more definitive insights into the validity of the approach presented. However, despite waiting for months, it was not feasible to obtain such data. Consequently, I relied on qualitative methods that still provided acceptable answers when applicable.

6.2. Future works

I do not discuss future work extensively. However, it is crucial to note that the current research merely represents the tip of the iceberg.

Indeed, despite the many limitations, it has become evident that this field is vast and largely uncharted. I think designers often underestimate it, treating it like conventional consumer software, which typically receives more research and understanding in the user experience domain. However, I believe this will change soon, given the significant advancements in recent years in cloud-based enterprise software, edge computing (IoT), and other technologies that enable the automation of work processes.

Future work should generally involve testing the UCD on a broader range of enterprise platforms. Moreover, it is essential to evaluate various iterations of this approach to ultimately establish an effective and secure user-centred approach for the design and development of software technologies employed by workers.

In conclusion, it is crucial to emphasize that this thesis does not address the actual modification of business-driven functionalities using user feedback. This topic is highly intricate and specialized, demanding considerable time and resources to understand and to provide definitive answers. Addressing this in the present work would have been unfeasible, especially since the company explicitly and reasonably prohibited me from doing so.

7 | Conclusion

This thesis focuses on a redesign project. Initially, I was only aware that the software pertained to the financial domain. Upon further investigation, I understood that the task was to redesign enterprise software used by corporate employees for their work.

Through this realization, I understood that it was not merely a redesign, but we were delving into a specific area known as Enterprise User Experience (Enterprise UX). As a result, I began to explore how to apply a user-centric approach within the domain of Enterprise UX.

The initial question was whether the UCD approach was applicable in this domain.

Based on the results, I can confirm that a user-centric approach applies to enterprise software. I can only affirm that the User-Centered Design approach can be employed in enterprise software to reduce complexity and enhance usability. Importantly, this does not alter the functional aspects of the software, which are dictated directly by the business needs and represent the core services provided by the system.

The methodologies and results I have achieved confirm that Card Sorting, Survey, Tree Testing, and Usability Testing are effective for tailoring a system to enterprise user needs. Furthermore, a robust Information Architecture and a well-defined Visual Hierarchy are essential in enterprise software, which is inherently specialized and complex. Indeed, refining these two core components has significantly improved user experience and usability.

Another point I wish to emphasize is that the UCD approach requires adaptation when applied to enterprise software. It is conceivable that a bespoke user-centric process specifically tailored for enterprise software might emerge. My proposition is merely an attempt, and I cannot assert that the process I have outlined is the best or the most appropriate for the software under consideration. I can only state that it has been effective and yielded positive outcomes.

In conclusion, the UCD approach for enterprise software is applicable. While it certainly aids in simplifying its usage, I cannot definitively speak to its impact on actual functional, business-driven aspects.

Acknowledgements

I am deeply grateful to several individuals whose support and guidance have been invaluable during my Master's thesis journey.

Firstly, I am profoundly grateful to my supervisor, Nieminen Mika P., whose patience and insights shaped the trajectory of this research. I also wish to thank my advisor, Riccardo Pesenti, for his invaluable advice and unwavering support.

I owe a debt of gratitude to my boss, Tucci Marilena, who provided me with an opportunity that significantly impacted my academic pursuits. I would also like to thank all my colleagues for their camaraderie and for creating an environment conducive to learning and growth.

To my family, your unwavering love and belief in me have been my rock. Your constant encouragement has propelled me forward, even in the face of challenges. My friends have been my pillars of strength and source of joy throughout this endeavour. Your understanding and companionship made this journey more rewarding. Thanks to my girlfriend, whose faith in my abilities and continuous support has been a guiding light. Your presence in my life has been a source of motivation and inspiration.

In conclusion, I want to express my gratitude to all the other individuals who interacted with me during this process and contributed in myriad ways, even if not explicitly mentioned. Their support, insights, and encouragement were integral to completing this thesis.

Bibliography

[1] Optimalsort: Discover how people categorize information. Online tool, Year of usage. URL https://www.optimalworkshop.com/optimalsort/.

[2] Treejack: Evaluate the findability of your content. Online tool, Year of usage. URL https://www.optimalworkshop.com/treejack/.

[3] A. Booth, D. S. Papaioannou, and A. Sutton. A., 2012. systematic approaches to a successful literature review, 61.

[4] L. Borisova. How startups should evaluate their website's UX? Master's thesis, Aalto University. School of Business, 2019. URL http://urn.fi/URN:NBN:fi:aalto-202001121502.

[5] R. E. Boyle, R. Pledger, and H.-F. Brown. Iterative mixed method approach to b2b saas user personas. *Proc. ACM Hum.-Comput. Interact.*, 6(EICS), jun 2022. doi: 10.1145/3534523. URL https://doi.org/10.1145/3534523.

[6] R. Budiu. Quantitative vs. qualitative usability testing. *Publication Name*, October 1 2017. URL https://www.nngroup.com/articles/quant-vs-qual/.

[7] A. Cooper, R. Reimann, and D. Cronin. *About Face 3: The Essentials of Interaction Design*. Wiley, Indianapolis, IN, 3rd edition, 2007.

[8] R. Daniel, A. Purwarianti, and D. P. Lestari. Interaction design of indonesian anti hoax chatbot using user centered design. In *2022 Seventh International Conference on Informatics and Computing (ICIC)*, pages 1–6, 2022. doi: 10.1109/ICIC56845.2022.10007024.

[9] J. Davis, K. Mengersen, S. Bennett, M. Arnold, A. Braunack-Mayer, and J. E. Hiller. Viewing systematic reviews and meta-analysis in social research through different lenses. *SpringerPlus*, 3:511, 2014. doi: 10.1186/2193-1801-3-511.

[10] I. F. del Amo, J. A. Erkoyuncu, R. Roy, R. Palmarini, and D. Onoufriou. A systematic review of augmented reality content-related techniques for knowledge transfer in maintenance applications. *Computers in Industry*, 103:47–71, 2018.

[11] S. K. Dey, M. Sherajul Islam Bappy, M. Sabrina Biswas, and S. Akter. A comparative usability experience analysis of card sorting and interactive dialogue model design technique. In *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, pages 268–273, 2018. doi: 10.1109/ICISET.2018.8745596.

[12] P. Faraday. Visually critiquing web pages. In *Multimedia'99: Proceedings of the Eurographics Workshop in Milano, Italy, September 7–8, 1999*, pages 155–166. Springer, 2000.

[13] A. Fard. Enterprise ux design: Make me think, Year of Publication. URL `https://adamfard.com/blog/enterprise-ux`.

[14] B. Foley. How to make a sitemap & why you should, June 15 2022. URL `https://www.nectafy.com/blog/how-to-make-a-sitemap`.

[15] J. Frich, M. M. Biskjaer, C. Remy, L. MacDonald Vermulen, and P. Dalsgaard. Revisiting the role of user research in design creativity. In *Proceedings of the 32nd European Conference on Cognitive Ergonomics*, ECCE '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450387576. doi: 10.1145/3452853.3452857. URL `https://doi.org/10.1145/3452853.3452857`.

[16] J. J. Garrett. A visual vocabulary for describing information architecture and interaction design, March 6 2002. URL `http://www.jjg.net/ia/visvocab/`.

[17] K. Gordon. Visual hierarchy in ux: Definition, January 17 2021. URL `https://www.nngroup.com/articles/visual-hierarchy-ux-definition/`.

[18] M. J. Grant and A. Booth. A typology of reviews: an analysis of 14 review types and associated methodologies. *Health Information & Libraries Journal*, 26(2):91–108, 2009. doi: 10.1111/j.1471-1842.2009.00848.x.

[19] J. Guo and P. Yan. User-centered information architecture of university library website. In *2011 3rd International Conference on Computer Research and Development*, volume 2, pages 370–372, 2011. doi: 10.1109/ICCRD.2011.5764153.

[20] A. Harley. The state of ux maturity: The 2021 ux maturity quiz. `https://www.nngroup.com/articles/state-ux-maturity-quiz/`, 2021.

[21] M. Hoggenmüller, M. Tomitsch, L. Hespanhol, T. T. M. Tran, S. Worrall, and E. Nebot. Context-based interface prototyping: Understanding the effect of prototype representation on user feedback. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Associa-

tion for Computing Machinery. ISBN 9781450380966. doi: 10.1145/3411764.3445159. URL https://doi.org/10.1145/3411764.3445159.

[22] ISO/TC 159/SC 4 Ergonomics of human-system interaction. Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. Published, July 2019. URL https://www.iso.org/standard/77520.html.

[23] J. Kjeldskov. *Interaction Design*, pages 19–31. Springer International Publishing, Cham, 2014. ISBN 978-3-031-02204-3. doi: 10.1007/978-3-031-02204-3_3. URL https://doi.org/10.1007/978-3-031-02204-3_3.

[24] J. A. Krosnick and S. Presser. Question and questionnaire design. In J. D. Wright and P. V. Marsden, editors, *Handbook of Survey Research*. Elsevier, San Diego, CA, 2nd edition, 2009. Chapter in the Handbook of Survey Research (2nd Edition).

[25] S. Kujala, M. Kauppinen, and S. Rekola. Bridging the gap between user needs and user requirements. In *Advances in Human-Computer Interaction I (Proceedings of the Panhellenic Conference with International Participation in Human-Computer Interaction PC-HCI 2001), Typorama Publications*, pages 45–50. Citeseer, 2001.

[26] C. Kussmaul and R. Jack. User interface prototyping: Tips and techniques. *J. Comput. Sci. Coll.*, 21(6):188–190, jun 2006. ISSN 1937-4771.

[27] K. Kuusinen. *Integrating UX Work in Agile Enterprise Software Development*. Doctor of science (technology) thesis, Technical University of Denmark, November 2015.

[28] C. Lallemand and E. Mercier. Optimizing the use of the sentence completion survey technique in user research: A case study on the experience of e-reading. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391573. doi: 10.1145/3491102.3517718. URL https://doi.org/10.1145/3491102.3517718.

[29] P. Laubheimer. Wireflows: A ux deliverable for workflows and apps. Online, December 4 2016. URL https://www.nngroup.com/articles/wireflows/.

[30] P. Laubheimer. Tree testing: Fast, iterative evaluation of menu labels and categories. Online, August 6 2023. URL https://www.nngroup.com/articles/tree-testing/.

[31] M. Li, R. Gao, X. Hu, and Y. Chen. Comparing infovis designs with different information architecture for communicating complex information. *Commun.*

Des. Q. Rev, 5(1):43–56, may 2017. doi: 10.1145/3090152.3090156. URL https://doi.org/10.1145/3090152.3090156.

[32] A. Linden. *The Future of Enterprise UX: Consumer-Grade UX in the Enterprise*. UXPin Inc., 2016. Head of Design at Asana.

[33] B. Mehta. What do enterprise software users really want?, Month of Publication Year of Publication. URL https://www.koruux.com/blog/what-do-enterprise-software-users-really-want/.

[34] W. Mengist, T. Soromessa, and G. Legese. Method for conducting systematic literature review and meta-analysis for environmental science research. *MethodsX*, 7: 100777, 2020. doi: 10.1016/j.mex.2019.100777. URL https://doi.org/10.1016/j.mex.2019.100777.

[35] A. Miller. Enterprise ux: the value of usability for enterprise software. *UX Magazine*, Jul 2021. URL https://uxmag.com/articles/enterprise-ux-the-value-of-usability-for-enterprise-software.

[36] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman. Preferred reporting items for systematic reviews and meta-analyses: The prisma statement. *Annals of Internal Medicine*, 151:264–269, 2009. doi: 10.7326/0003-4819-151-4-200908180-00135.

[37] D. Moher, L. Shamseer, M. Clarke, D. Ghersi, A. Liberati, M. Petticrew, P. Shekelle, L. A. Stewart, and P.-P. Group. Preferred reporting items for systematic review and meta-analysis protocols (prisma-p) 2015 statement. *Systematic Reviews*, 4:1, 2015. doi: 10.1186/2046-4053-4-1.

[38] K. Moran. Usability testing 101. *Publication Name*, December 1 2019. URL https://www.nngroup.com/articles/usability-testing-101/.

[39] H. Müller, A. Sedley, and E. Ferrall-Nunge. *Survey research in HCI*, pages 229–266. 03 2014. ISBN 978-1-4939-0377-1. doi: 10.1007/978-1-4939-0378-8_10.

[40] J. Nielsen. Usability 101: Introduction to usability. https://www.nngroup.com/articles/usability-101-introduction-to-usability/, January 2012.

[41] J. Nielsen. How many test users in a usability study? *Publication Name*, June 3 2012. URL https://www.nngroup.com/articles/how-many-test-users/.

[42] J. Nielsen. Thinking aloud: The #1 usability tool. *Publication Name*, January 15 2012. URL https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/.

[43] J. Nielsen. Enterprise user experience. https://www.nngroup.com/videos/enterprise-user-experience/, 2022.

[44] M. P. Nieminen. *User-Centered Design Competencies - Construction of a Competency Model*. Doctoral thesis, School of Science, 2015. URL http://urn.fi/URN:ISBN:978-952-60-6127-6.

[45] D. Norman and J. Nielsen. The definition of user experience (ux). https://www.nngroup.com/articles/definition-user-experience/, 2016.

[46] S. Paea and R. Baird. Information architecture (ia): Using multidimensional scaling (mds) and k-means clustering algorithm for analysis of card sorting data. *J. Usability Studies*, 13(3):138–157, may 2018.

[47] J. O. V. Paiva, B. Borges, P. Duarte, I. S. Santos, R. M. C. Andrade, and V. Lelli. Pro-ux: System redesign process for improving the ux. In *Proceedings of the XXI Brazilian Symposium on Software Quality*, SBQS '22, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399999. doi: 10.1145/3571473.3571501. URL https://doi.org/10.1145/3571473.3571501.

[48] D. Papaioannou, A. Sutton, C. Carroll, A. Booth, and R. Wong. Literature searching for social science systematic reviews: consideration of a range of search techniques. *Health Info Libr J*, 27(2):114–122, Jun 2010. doi: 10.1111/j.1471-1842.2009.00863.x.

[49] K. Pernice. Ux prototypes: Low fidelity vs. high fidelity, December 18 2016. URL https://www.nngroup.com/articles/ux-prototype-hi-lo-fidelity/.

[50] A. H. Phillips, R. Yang, and S. Djamasbi. Do ads matter? an exploration of web search behavior, visual hierarchy, and search engine results pages. In *2013 46th Hawaii International Conference on System Sciences*, pages 1563–1568, 2013. doi: 10.1109/HICSS.2013.181.

[51] N. R. Ramadhani, A. Mulyanto, and G. S. Niwanputri. Designing interaction and user interface of computational thinking digital game for children using user-centered design approach. In *2020 7th International Conference on Advance Informatics: Concepts, Theory and Applications (ICAICTA)*, pages 1–6, 2020. doi: 10.1109/ICAICTA49861.2020.9429049.

[52] S. Renuka. Identifying personas for enterprise applications, Month of Publication Year of Publication. URL https://www.koruux.com/blog/identifying-personas-for-enterprise-applications/.

[53] A. Resmini and L. Rosati. A brief history of information architecture. *Journal of*

Information Architecture, 3:33–46, 12 2011. doi: 10.55135/1015060901/112.006/3.024.

[54] C. Rohrer. When to use which user-experience research methods. Online, July 17 2022. URL https://www.nngroup.com/articles/which-ux-research-methods/.

[55] L. Rosenfeld, P. Morville, and J. Arango. *Information Architecture for the World Wide Web*. O'Reilly Media, Inc., 4th edition, October 2015. ISBN 9781491911686. URL https://www.oreilly.com/library/view/information-architecture-4th/9781491913529/.

[56] K. Sherwin. Card sorting: Uncover users' mental models for better information architecture. Online, March 18 2018. URL https://www.nngroup.com/articles/card-sorting-definition/.

[57] S. Silverstein. Breaking down enterprise ux design. *UX Collective*, Apr 2020. URL https://uxdesign.cc/breaking-down-an-enterprise-evaluation-part-1-the-unique-challenge-c299cf0a636a.

[58] J. M. Six. The differences between enterprise and consumer ux design. *UXmatters*, Jan 2017. URL https://www.uxmatters.com/mt/archives/2017/01/the-differences-between-enterprise-and-consumer-ux-design.php.

[59] H. Snyder. Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104:333–339, 2019. ISSN 0148-2963. doi: https://doi.org/10.1016/j.jbusres.2019.07.039. URL https://www.sciencedirect.com/science/article/pii/S0148296319304564.

[60] A. Suzianti and G. Arrafah. User interface redesign of dental clinic erp system using design thinking: A case study. In *Proceedings of the 5th International Conference on Industrial and Business Engineering*, ICIBE '19, page 193–197, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450376532. doi: 10.1145/3364335.3364369. URL https://doi.org/10.1145/3364335.3364369.

[61] G.-M. Tang, H.-Y. Hu, S.-Y. Chen, and W. Jeng. A cross-cultural study on information architecture: Culture differences on attention allocation to web components. In A. Sundqvist, G. Berget, J. Nolin, and K. I. Skjerdingstad, editors, *Sustainable Digital Communities*, pages 391–408, Cham, 2020. Springer International Publishing. ISBN 978-3-030-43687-2.

[62] A. Tapia, A. Moquillaza, J. Aguirre, F. Falconi, A. Lecaros, and F. Paz. A process to support the remote tree testing technique for evaluating the information architec-

ture of user interfaces in software projects. In M. M. Soares, E. Rosenzweig, and A. Marcus, editors, *Design, User Experience, and Usability: UX Research, Design, and Assessment*, pages 75–92, Cham, 2022. Springer International Publishing. ISBN 978-3-031-05897-4.

[63] H. Verma. How to create user-centric enterprise ux design effectively?, May 18 2022. URL `https://closeloop.com/blog/how-to-create-user-centric-enterprise-ux-design-effectively/`.

[64] J. Walter and C. Braunsdorf. Defining enterprise ux. *UX Matters*, September 2018.

[65] *BLOOMBERG FOR EDUCATION: A Bloomberg Professional Service Offering - Getting Started Guide*. Wenzhou-Kean University, February, 2012. URL `https://wku.edu.cn/wp-content/uploads/2018/04/Bloomberg-User-Guide.pdf`.

[66] K. Whitenton. Tree testing part 2: Interpreting the results. Online, July 9 2017. URL `https://www.nngroup.com/articles/interpreting-tree-test-results/`.