CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

A single source of text: Supporting text management in cross-functional organisations

A qualitative case study informing design recommendations to streamline text management processes and synchronise teams to improve cross-functional collaboration

Master's thesis in Computer science and engineering

MAJA ULÉN
TARA LYNCH

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2023

Master's thesis 2023

A single source of text: Supporting text management in cross-functional organisations

A qualitative case study informing design recommendations to streamline text management processes and synchronise teams to improve cross-functional collaboration

MAJA ULÉN

TARA LYNCH

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2023

A single source of text: Supporting text management in cross-functional organisations
A qualitative case study informing design recommendations to streamline text management processes and synchronise teams to improve cross-functional collaboration
MAJA ULÉN
TARA LYNCH

Supervisor: Mohammad Obaid, Department of Computer Science and Engineering
Advisor: Juntima Nawilaijaroen, Ericsson
Examiner: Palle Dahlstedt, Department of Computer Science and Engineering

Master's Thesis 2023
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2023

A single source of text: Supporting text management in cross-functional organisations
A qualitative case study informing design recommendations to streamline text management processes and synchronise teams to improve cross-functional collaboration
MAJA ULÉN AND TARA LYNCH
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

While previous research address problems related to cross-functional collaboration and content management, more research is needed regarding managing text strings within and across teams. As organisations become more aware of the importance of UX writing, the need for processes and tools that support it increases. This project was in collaboration with Ericsson in Gothenburg and investigates an internally facing problem regarding text management for their Network Management Portal. Following the design thinking process, several studies, including designers and developers, were conducted to uncover and address challenges with current practices. Initial studies explore the challenges within current text management processes to understand what factors impact productivity and text quality. The identified challenges were highly related to the cross-functional environment in which participants work. Issues such as a lack of shared understanding, manual work, different responsibilities and perspectives, and the need for one source of truth are discussed. The challenges uncovered during the empathy phase guided subsequent phases focusing on designing solutions to address the experienced challenges. This thesis presents requirements for a text management tool aiming to streamline text management processes and synchronise teams to improve cross-functional collaboration. Furthermore, the main result includes a set of 13 design recommendations connected to the requirements intended to be an actionable plan in mitigating the challenges, encompassing features, design proposals, design principles, and implementation considerations. The recommendations were informed by interviews, competitor analysis, a design workshop, and evaluations carried out throughout the project. The findings emphasise practical implications and highlight additional needs beyond content management, such as considering multiple perspectives and situating text strings in a larger context to provide meaning.

Keywords: User-centred design, User experience, Text strings, Cross-functional collaboration, UX writing, Content strategy

Acknowledgements

First, we would like to express our gratitude towards Ericsson, who provided us with this opportunity and supported us throughout the entire process. We want to thank the people at Ericsson for welcoming us with open arms. Your engagement, insights, and feedback have been invaluable to this thesis. Moreover, we would like to express a warm thank you to our supervisor at Chalmers, Mohammad Obaid, for your dedication and, of course, patience throughout this process. Your support has been incredibly meaningful to us. Finally, we want to express gratitude towards our partners and family for their unconditional support and ability to motivate us when things get tough. We are grateful for every person throughout this thesis who has gotten engaged and allowed us to explore and improve in many different areas.

Maja Ulén and Tara Lynch, Gothenburg, May 2023

Contents

List of Figures **xi**

List of Tables **xiii**

1 Introduction **2**
1.1 Problem . . . 3
1.1.1 Research questions . . . 3
1.2 Delimitations . . . 4

2 Background **5**
2.1 Ericsson Dedicated Networks . . . 5
2.1.1 Ericsson private 5G . . . 6
2.1.2 Network Management Portal . . . 6
2.1.3 Definition of text and text management . . . 6
2.1.3.1 UX text management process . . . 7

3 Previous research **9**
3.1 Cross-functional collaboration . . . 9
3.1.1 Challenges in cross-functional collaboration . . . 9
3.1.2 Mitigating challenges in cross-functional collaboration . . . 10
3.2 Managing content in organisations . . . 11
3.3 Towards UX writing . . . 13

4 Theory **16**
4.1 User-centred design . . . 16
4.1.1 Design thinking . . . 16
4.2 User experience . . . 17
4.3 Guiding principles . . . 18
4.3.1 Design principles . . . 18
4.3.1.1 Learnability . . . 18
4.3.1.2 Ease of use . . . 19
4.3.1.3 Safety . . . 20
4.3.1.4 Accommodation . . . 20
4.3.2 Psychological principles and interface design . . . 21
4.3.2.1 Memory and attention . . . 21
4.3.2.2 Perception . . . 22
4.4 Atomic design . . . 22

5 Methodology **24**
5.1 Empathise . . . 24
5.1.1 Semi-structured interviews . . . 24
5.1.2 Competitor analysis . . . 25
5.2 Define . . . 25
5.2.1 Affinity diagram . . . 25
5.2.2 Empathy maps . . . 26
5.2.3 Defining and prioritising requirements . . . 26
5.3 Ideate . . . 26
5.3.1 Design workshop . . . 27
5.3.2 Brainstorming and sketching . . . 27
5.3.3 User flows . . . 28
5.4 Prototype . . . 28
5.4.1 Wireframes . . . 29
5.4.2 High-fidelity prototyping . . . 29
5.5 Evaluate . . . 29
5.5.1 Expert review . . . 29
5.5.2 Design walkthrough . . . 30

6 Process **31**
6.1 Interviews . . . 31
6.1.1 Pilot study . . . 31
6.1.2 Participants . . . 32
6.1.3 Procedure . . . 32
6.1.4 Analysis . . . 33
6.1.5 Findings . . . 34
6.2 Empathy maps . . . 47
6.3 Competitor analysis . . . 48
6.3.1 Software . . . 48
6.3.2 Procedure . . . 49
6.3.3 Findings . . . 50
6.4 Design workshop . . . 56
6.4.1 Participants . . . 57
6.4.2 Procedure . . . 57
6.4.3 Analysis . . . 58
6.4.4 Findings . . . 59
6.5 Defining requirements . . . 62
6.5.1 Procedure . . . 62
6.5.2 Requirements list . . . 62
6.6 Low-fidelity prototypes . . . 65
6.6.1 Mind maps . . . 65
6.6.2 Sketching . . . 67
6.6.3 User flows . . . 69
6.6.4 Wireframes . . . 73
6.6.5 Evaluation . . . 73
6.6.5.1 Participants . . . 74

6.6.5.2 Procedure 74
6.6.5.3 Findings 76
6.7 Semi-functional prototype 77
6.7.1 Practical implementation and current software 84

7 Result **85**

8 Discussion and Conclusions **114**
8.1 Research questions 115
8.2 Process 118
8.3 Ethical considerations 119
8.4 Generalisability 120
8.5 Limitations 121
8.6 Future work 122

Bibliography **122**

A Project plan **II**

B Interview protocol **IV**

C Consent form **VI**

D Affinity diagrams **VIII**

List of Figures

2.1 An overview of the teams within Dedicated Networks. 5
2.2 UX text management process. 8

4.1 The design thinking process. 17
4.2 Components of Atomic design. 23

6.1 Empathy map UX. 47
6.2 Empathy map FE. 48
6.3 Competitor analysis summary. 51
6.4 Requirements list part 1. 63
6.5 Requirements list part 2. 64
6.6 Dashboard mind map. 65
6.7 Editor mind map. 66
6.8 Repositories mind map. 66
6.9 Teams mind map. 67
6.10 Dashboard sketch. 68
6.11 Editor side sheet sketches. 68
6.12 Editor sketches. 69
6.13 Flow of locating a text. 70
6.14 Flow of editing a text. 71
6.15 Flow of adding a text. 72
6.16 Atomic design in wireframes 73
6.17 Editor 74
6.18 Dashboard 75
6.19 Side sheet editor 75
6.20 Atomic design updated. 78
6.21 Global navigation. 78
6.22 Dashboard. 79
6.23 Editor. 80
6.24 Side sheet that slides in from the right where users can edit text, see text details, and text history. 80
6.25 Modal for adding new text. 81
6.26 Modal for exporting text. 81
6.27 Text history for all text strings. 82
6.28 Repository for managing variables. 82
6.29 Settings for managing files. 83
6.30 Settings for managing team members and permissions. 83

7.1 Editor triggered by selecting a text string, enabling direct edit. 87
7.2 Export modal. 87
7.3 Edit pages. 88
7.4 Move objects. 89
7.5 Connection between JSON syntax and the representation in the interface. 91
7.6 Add new text. 91
7.7 Editor as a sidebar allow for contextual information to stay visible. . 92
7.8 Contextual navigation within the dashboard. 93
7.9 Contextual navigation within details. 94
7.10 Table with text strings. 95
7.11 Expandable list items in the local navigation. 95
7.12 Details with related information and editor with error log. 97
7.13 Locate and change version in the side navigation bar. 99
7.14 Text status. 99
7.15 Multiple ways to locate text. 101
7.16 Active and inactive filters. 102
7.17 Find and replaced collapsed. 104
7.18 Find and replace expanded. 104
7.19 Undo text changes. 104
7.20 An example of a shared repository with allowed and deprecated terms. 105
7.21 Editor detecting deprecated terms. 106
7.22 Add new items or insert into editor. 108
7.23 Variable indication without (left) and with (right) visual feedback. . . 108
7.24 Text-specific history integrated into the editor. 110
7.25 History page concerning all text strings. 110
7.26 Insert links to Jira and Slack. 111
7.27 Team settings. 112

List of Tables

6.1 Interview participants. 32
6.2 An overview of interview themes and sub-themes. 33
6.3 An overview of the software included in the competitor analysis. . . . 49
6.4 Competitor analysis framework. 50
6.5 Themes emerged from workshop analysis. 59

7.1 An overview of the design recommendations. 85

List of Abbreviations

API Application Programming Interface

BE Back-end

CCMS Component Content Management System
CMS Content Management System
CPI Customer Product Information

DAM Digital Assets Management
DT Design thinking

FE Front-end

ID Identifier

JSON JavaScript Object Notation

NMP Network Management Portal

UCD User-centred design
UI User interface
UX User experience

1

Introduction

The emergence of new technology continuously changes the digital landscape in which organisations operate. Thus, organisations must respond and adapt to the dynamic environment to maintain competitiveness [1]. The rapid change in technologies and industries is referred to as Industry 4.0. It is a set of solutions based on advanced technologies such as cloud computing and services, autonomous robotics, and the industrial internet of things [2]. Industry 4.0 secures companies within industries through greater flexibility and productivity; however, it completely changes how industries operate, including how to develop, design, and produce products and services [3].

Industries and enterprises operating in a technology-driven environment must provide ubiquitous, real-time, and interconnected communication and collaboration between machines, systems, and humans [2]. Consequently, human-machine interfaces are used to monitor and increase performances, reduce risks and prevent errors. However, the emergence of advanced technologies increases the amount and complexity of data produced and collected. Accordingly, interfaces must be designed to support this complexity. Designing such interfaces is challenging as operators must quickly process and understand the information. Thus, the usability of such interfaces is vital [4]. Creating, providing, and managing more information and content require organisations to adapt their working methods. As a result, content management and content strategies have evolved, allowing businesses to support complex processes of managing content [5] while promoting content consistency [6].

Benefitting from globalisation requires organisations to digitally transform and adapt their organisational structures according to the transformation [7]. Increased and more efficient collaboration is crucial to solving complex issues that arise from an interconnected and data-driven environment. Thus, organisations are challenged to expand and re-strategise their way of working and collaborating [7]. As a result, cross-functional collaboration is no longer an option but rather a requirement [8], allowing organisations to work better and produce better results. However, successful cross-functional collaboration requires facing practical challenges such as reaching a shared understanding that facilitate synchronisation across teams [9, 10, 11, 12].

Ericsson Private 5G, a solution to power 5G networks throughout designated areas, is specifically tailored to drive Industry 4.0. Private 5G networks are managed through the Network Management Portal (NMP), a unified network management system.

Since the usability of such an interface is vital to monitor network performance [4], the text within the portal must be carefully considered. Furthermore, text is essential in communication between users and the systems. Therefore, terms and wording must be consistently used when referring to a specific concept or function [13].

1.1 Problem

The NMP is developed and maintained by Dedicated Networks, an organisation within Ericsson with several teams. More than six teams are involved in text management at Ericsson Dedicated Networks. In cases where no team specifically focuses on UX writing, the text must be aligned according to established guidelines across several teams [13].

The challenges regarding cross-functional collaboration are well documented. However, more research is needed on aligning text management processes between teams. Within Dedicated Networks, there is currently no intuitive way to align text management processes between teams, resulting in inconsistencies in the text and confusion about standards and potential updates. Many issues arise as text is not stored anywhere and, thus, not shared between teams. The lack of a shared space specifically for text management results in difficulties communicating cross-functionally. Furthermore, different teams have their processes because of different responsibilities. For example, the UX team uses software to design User interface (UI) elements, including text, whereas the Front-end team works with code and needs to handle their text in specific libraries. Therefore, the issue lies in collaboration, aligning processes to ensure standards are being followed, and ultimately the need for a joint solution specifically for text.

The internal issues regarding text management ultimately affect text quality and will make the portal less usable. Not only will it affect the user experience, but it also reduces productivity within Ericsson and causes unnecessary confusion and frustration. Furthermore, users must have complete control and overview of the network resulting in NMP containing a significant amount of text. Thus, a streamlined process is required to facilitate future text updates and further development regarding localisation and globalisation.

1.1.1 Research questions

This thesis aims to investigate text management processes to identify challenges that hinder cross-functional collaboration and present solutions for a text management tool that mitigates the discovered challenges.

Empathising is the first crucial step in understanding the problem space and user needs, both hidden and apparent. Revealing these needs is the first step towards generating innovative solutions [14, 15]. Identifying and understanding the challenges behind current text management processes and working cross-functionally is essential before focusing on solutions. Thus, this thesis asks the following question:

1. *What are the main challenges with current approaches in a cross-functional organisation that affect teams' productivity and ability to synchronise their text management processes?*

Research regarding improving cross-functional collaboration and developing frameworks to aid such collaboration exist [16, 17, 18, 10, 19]. However, more tools are needed to align teams efficiently, especially regarding text management. Nevertheless, teams work differently, and one of the most important aspects of cross-functional collaboration is finding a way that suits the specific organisation [19].

It is essential to first fully understand what causes friction and issues within current text management processes. However, this thesis aims to shape solutions accordingly by focusing on streamlining text management processes and synchronising teams to improve cross-functional collaboration. Requirements will be defined based on the first research question and informed throughout the process. These requirements will be exemplified through a tool and result in a set of design recommendations. Thus, a second research question addresses the outcome of the process, formulated as follows:

2. *What design recommendations can be proposed for a text management tool to streamline text management processes and synchronise teams to improve cross-functional collaboration?*

1.2 Delimitations

While links between teams will be explored, the thesis will focus on the UX team's process without further investigating other teams' processes. Thus, no changes will be suggested to their text management processes. Furthermore, no changes to the NMP will be suggested. Instead, the focus is on designing a new tool that will be added to streamline the text management processes. As this is a case study, requirements will be identified based on data regarding internal challenges within Ericsson Dedicated Networks. As a result, it cannot be ensured that the design recommendations and requirements are generalisable over other organisations. Nevertheless, steps to improve the applicability outside of Ericsson are taken.

Lastly, a limited number of interviews will be conducted, raising awareness of the sample size. While the tool will be used across teams, the focus is still on the UX team. Thus, the study is limited to the number of team members. Nevertheless, they are expert users; thus, a smaller sample size is acceptable [20].

2

Background

The following sections provide background related to the problem and an overview of current processes within Ericsson. First, it briefly introduces Ericsson Private 5G and its associated management portal, the NMP. References to teams and processes within Ericsson are exclusively regarding Ericsson Dedicated Networks. Then, internal text management processes are discussed to understand the current practices within the UX team.

2.1 Ericsson Dedicated Networks

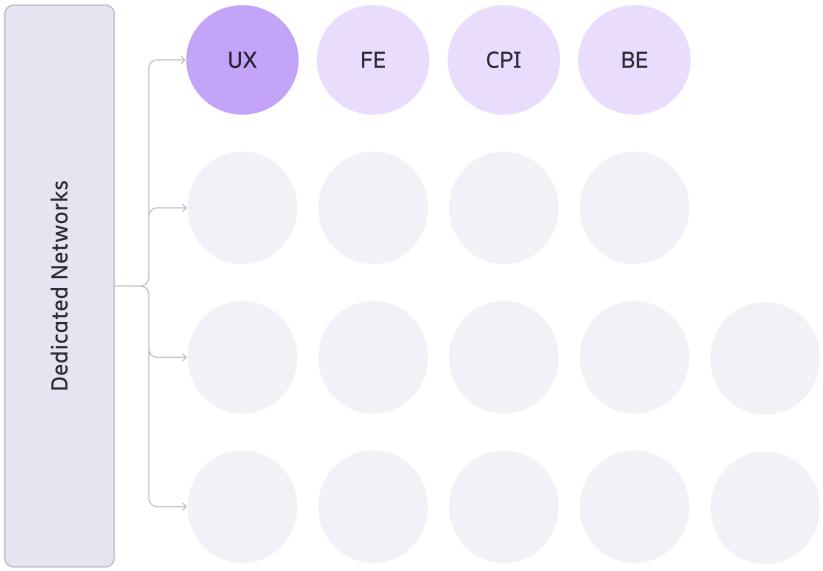


Figure 2.1: An overview of the teams within Dedicated Networks.

Ericsson is one of the world's largest equipment manufacturers for building mobile communications networks. As industries like manufacturing, ports, mining, airports, and energy change, Ericsson transforms from analogue to digital, involving sensors and artificial intelligence. A modern communication backbone to support the adoption, use and scaling of new and emerging data-driven technologies is required to support the change. As part of Ericsson, the Dedicated Networks department focuses on this transformation, ensuring dedicated networks for industrial connectivity

through automation and digitalisation. It includes developing and maintaining Ericsson Private 5G and its associated management portal NMP.

Dedicated Networks comprise four different, internally referred to as, streams. Each stream consists of separate teams. This thesis focuses on the stream highlighted in figure 2.1. These teams are directly involved in designing and implementing content for the NMP. Teams within specific streams collaborate the most. Thus, while the primary focus is on the UX team, other teams will naturally be involved.

2.1.1 Ericsson private 5G

Ericsson Private 5G [21] is a cellular connectivity solution that provides high-speed, secure 4G and 5G connectivity in designated private areas. It is tailored to drive Industry 4.0 by accelerating the digital transformation of organisations. It supports adopting, using, and scaling new and emerging data-driven technologies.

Ericsson Private 5G currently ensures indoor and outdoor connectivity of various use cases, industries, and enterprises. Thus, it is imperative to provide fast, intelligent, and reliable networks to meet a variety of connectivity needs. Managing such networks covers everything from performance, vulnerability, and infrastructure management achieved through remote and automated monitoring, troubleshooting, configuration, and network optimisation [22]. Measuring, understanding, and managing networks in real-time allow organisations complete control over the connectivity and connected devices [21]. The usability of such an interface is vital since operators must quickly process and understand the information provided [4].

2.1.2 Network Management Portal

Network management, utilising automated resolutions, is imperative in detecting, monitoring, and managing threats [22]. Ericsson Private 5G is managed through the NMP, a unified network management system designed to manage and control the network and connected devices [21]. It is a cloud-based web portal for users to install, configure, manage, and monitor their wireless networks. NMP features an intuitive interface to support several different use cases. Thus, the portal must stay consistent regarding UI elements and text. NMP's stakeholders include external users, i.e., the end users and internal users at Ericsson. To ensure a safe, secure, and continuously connected enterprise environment, different roles within NMP will have access to different information and functionality. Accordingly, the correct text and information must be exposed to the correct roles.

2.1.3 Definition of text and text management

Throughout this thesis, the terms text and text management are used frequently. Text refers to text strings used in the NMP, including UI text such as labels, inputs, event logs, error messages, etcetera, and associated text such as emails. However, it does not refer to technical documentation guides within NMP. Most importantly, text strings solely contain text; modules such as tables, images, and videos are not

considered. Text strings can consist of one word, a sentence, and sometimes one or more paragraphs.

Text management refers to how text strings are managed, including:

- The flow of text, from decision to implementation
- Decisions regarding text; adding new text and updating, deleting, and moving existing text
- Communicating and documenting decisions regarding text
- How text is stored

Thus, text management does not refer to how text is written in terms of terminology and tone of voice; rather, it refers to ways of working with text prior to publishing.

2.1.3.1 UX text management process

Currently, teams within Ericsson have control over separate text for NMP and have different processes and software to manage text. In addition, as there are no shared processes or guidelines for text, teams are not sufficiently synchronised. Therefore, difficulties arise in acquiring an overview of text content before implementing it in the NMP. As a result, there are several consistency issues regarding writing style and phrasing.

This thesis focuses on the UX team's process. Figure 2.2 illustrates the UX team's process of understanding, defining, and implementing text for the NMP. The process is a standard process involving research, ideation, prototype, evaluation, and implementation. First, the UX team must understand the current state of text within NMP and how it affects users to identify potential issues or new content that must be created as new features are developed.

The research phase is followed by ideation, where terms and phrases are brainstormed concerning different features and elements. During this phase, UX remains a continuous communication with the Customer Product Information (CPI) team to ensure that text is suitable and consistent with the overall language used in the portal. Once the terms and phrases are established, the UX team updates their prototypes. These prototypes are accessible to the Front-end (FE). Thus, they are informed of updates by viewing the prototypes and through personal communication. Then, usability testing evaluates the added or updated text elements. If potential re-terming is needed, it is again communicated to FE using the same principles. FE implements the changes, and the process is then iterated. An aspect to consider with the current setup is the chance that FE has accessed the initial prototypes and implemented them accordingly while missing potential updates following usability testing.

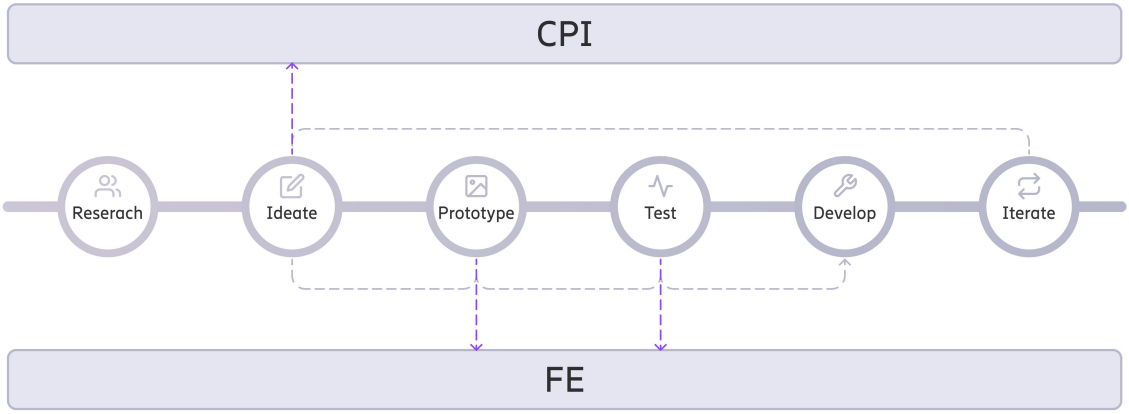


Figure 2.2: UX text management process.

While cross-functional collaboration works well in several aspects, text management must be improved. Some reasons for the current inconsistencies in the text could be identified during initial discussions with the UX team at Ericsson. Currently, there are no shared processes or standards between teams, meaning that all teams have software specific to their processes. In some instances, developer teams had the impression that they did not need a close collaboration with the UX team as their domains are so different. As a result, some teams may produce text without communicating it to the UX team.

Furthermore, while FE has access to UI prototypes produced by the UX team, they must actively check for any updates if they are not updated about changes. Communicating standards or changes is reliant on personal communication. Ultimately, an overview of the text for NMP and a shared database for content or style guides must be implemented. Several guidelines developed by the UX team encompass all forms of content and UI elements. Thus, a style guide for text production is also available. However, it is not easily accessible to all teams.

3

Previous research

This chapter examines previous research related to the background of this study to situate it in research and topics. First, it addresses cross-functional collaboration and discusses its challenges. It then proceeds to discuss content management and content strategy. While there is a need for more research regarding text management, several studies focus on content management and content strategy. Finally, technical communication is briefly addressed and discussed in relation to UX writing.

3.1 Cross-functional collaboration

A growing understanding of the importance of UX increases the demand for close collaboration between teams, for example, designers and developers [17]. Thus, in modern organisations, cross-functional collaboration is a requirement rather than an option. Cross-functional collaboration refers to a collaboration between teams within different specialisations, all striving towards a common goal; the benefit of cross-functional collaboration is the ability to synergise work processes while obtaining different perspectives [8]. However, cross-functional collaboration is associated with several challenges.

3.1.1 Challenges in cross-functional collaboration

While cross-functional collaboration is intended to allow organisations to work better and produce better results, research shows the practical challenges in such organisations. Agile is a standard software development approach based on cross-functional collaboration within and between teams [10]. Work is done in time-restricted sprints. Therefore, User-centred design (UCD) must often be adapted and integrated within the established agile working method. However, traditional UX methods and approaches do not always align well with the short sprints that agile involves [12]. Synchronising team processes has been identified as a primary issue in integrating UCD and agile methods [9]. However, finding a way to synchronise teams without unnecessary time or work is challenging. Especially as self-organisation is one of the pillars of the agile approach [8]. Previous research identifies that synchronising different teams and their processes is a common issue. Furthermore, tools must be aligned with the specific processes. However, there is both a lack of tools available and a lack of research regarding existing tools. As a result, they can often lead to

inefficiency rather than fulfilling the goal of streamlining cross-functional collaboration. Nevertheless, the main issue discussed in almost all research is communication.

Friberg and Johansson [16] interviewed designers and developers and identified issues regarding processes, information-sharing, feedback, and decision-making. Processes proved to be non-existent or non-structured; in many cases, they were disconnected. Disconnected processes were considered a widespread issue as they led to a lack of shared understanding, which impacts workflow and problem-solving. As a result, Friberg and Johansson present a set of guidelines for optimising cross-functional collaboration: including developers during pre-study, increasing communication, and sharing all files.

Similarly, Barke and Prechelt [19] investigate motivating and de-motivating aspects in a cross-functional environment. They use the term cross-fertilisation to explain the potential benefits of cross-functional teams and argue that merely being a cross-functional team is not enough to create an optimal work environment. The inclination to interact cross-functionally is crucial to achieving cross-fertilisation, which the authors conclude is dependent on the perception of inefficiency, desire to learn, sense of domain responsibility, issues with career progression, domain distance, level of detail, and focus. In summary, the main issue cross-functional teams face is related to communication. However, several factors are, in turn, affecting the inclination to interact cross-functionally.

Furthermore, research indicates that communication problems can arise due to non-structured communication, a lack of tools or shared tools. Barke and Prechelt [19] indicate that non-structured communication channels hinder efficient collaboration. Decisions that are not adequately communicated between teams, thus, result in missing information [23]. A lack of tools for documentation and information sharing can also hinder communication [18]. Furthermore, a lack of shared tools can lead to rework and redundancy [23].

3.1.2 Mitigating challenges in cross-functional collaboration

Despite a growing need for collaboration tools, Chasanidou et al. [24] argue that there is limited knowledge about how the tools work in practice. Thus, the authors conduct an exploratory study testing the task management tool UpWave. They conclude that tools will be utilised differently across organisations. Companies will shape the tool's application depending on their expertise and working methods. Furthermore, team structures directly impact how the tool is used. Chasanidou et al. suggest that having teams in the same space limits the need for collaboration tools. However, that would require a naturally functioning collaboration and communication, all of which have been identified as challenges in previous research. Collaboration tools should not just be intended for remote collaboration.

There are existing tools aimed at designer and developer collaboration; however, these tools often lead to inefficiency rather than an improved process [17]. Leiva et al. [23] conclude that this is due to the different teams' tools operating in isolation. Bexiga et al. [17] present an approach designed to improve current tools' efficiency.

The approach helps automate the conversion of design artefacts to low-code UI components using model transformation and meta-modelling techniques. They suggest a solution involving creating an extensive style guide, including all patterns and components according to the current project's brand guidelines. While the suggested approach is somewhat complex, the main factors to improve designer and developer collaboration are simplifying creating style guides, maintaining shared repositories, and streamlining the design-to-code conversions by addressing both design and development in the style guides.

A common theme throughout cross-functional collaboration research is the need for detailed documentation and reliable communication. Not having a shared understanding affects how well teams can synchronise their work [9, 10, 11, 12]. However, despite established processes, difficulties in collaboration still exist due to tensions and breakdowns in processes [23]. Asynchronised processes, in turn, impacts consistency. Consistency is one of the main factors impacting user trust [11]. Interfaces must be visually consistent and contextually aligned. Inconsistencies in UI elements, text content, or interactions may lead to a more significant cognitive load as users must spend more time understanding and learning the system. One solution for mitigating consistency issues is adopting style guides [19, 10, 11]. Style guides are developed by designers aimed at obtaining a shared understanding to improve consistency throughout an interface or between platforms. Furthermore, they can be used as a means of communication. Style guides can define colours, typography, layout, components, icons, etcetera. Moreover, style guides can be supplemented with, for example, writing guides defining tone of voice and other semantic rules for text production.

Successful cross-functional collaboration depends on a shared understanding and synchronised teams. While research highlights the importance of documentation, it also highlights the need for them to be accessible across teams. Having a shared platform to manage text is a particular problem Ericsson is currently facing. Separated processes and solely relying on personal communication are unrealistic to improve text management. Nudelman [12] exemplifies this issue by explaining that a developer will not wait until the next meeting or until someone has communicated changes; they will make assumptions to finish their work.

3.2 Managing content in organisations

Digitisation requires organisations to produce, provide, and manage large volumes of content; images, text, video, audio, etcetera. Traditionally, a Content Management System (CMS) is used to manage content, in which content is managed on a document level. These documents are intended for specific contexts, audiences, or devices. However, as content increases, it becomes demanding to maintain. Thus, research and practice have moved towards component content management, creating and managing content through smaller chunks, i.e., components, instead of entire documents. A modular approach to content management promotes organisations to reuse content across information products and adjust the content to specific users

and devices. A Component Content Management System (CCMS) allows users to manage content on a component level [6].

The relevance and advancement of content management are proven by the fact that standards have been established around the subject, including ISO/IEC/IEEE 26531:2023, a standard for managing content for users, the product life cycle, and service management items [25]. Hackos [26] provides a framework with key concepts from ISO/IEC/IEEE 26531:2015 to promote its importance for managing content. Hackos further argues that standard implementations ensure interchangeability, allowing organisations to cope with digital transformation as information resources evolve.

Content strategy and content management are largely intertwined in existing research. For example, while creating and managing large volumes of content requires a content strategy [5], content management is considered a content strategy best practice [27]. Thus, the practices are often discussed together to be dependent on each other to obtain the most significant benefit. While literature addresses content strategy to a great extent, Getto et al. [27] highlight that there are no established and universal best practices. Thus, the authors call for the involvement of content strategists in research to translate best practices into literature. Furthermore, practices need to be tested across organisations of different sizes and with different needs to comprehend what can be generalisable and applied as a standard and which are highly contextual.

Issues experienced in practice before a CCMS implementation include difficulties with auditability and tracking content changes due to software discontinuity, difficulties in fulfilling the increase of specialised documentation, and increased cycle times and redundant translation as authors and translators operate in isolation [28]. Related to the first issue, McCarthy et al. [29] identify that transmitting documents for revision through multiple mediums cause difficulties in tracking and reusing content. Case studies implementing CMS have been conducted to address some of these issues. Bailie and Huset [30] implemented two different CMS in two organisations with a four-phase training plan, focusing on the effect of such implementation. They concluded that it is vital to determine specific needs when choosing a CMS and to ground writers' theoretical knowledge. As writers need to learn content structures and standards and to do so collaboratively, their needs must match the training concerning the system's complexity. Accordingly, they highlight that technology affects writing styles and processes and vice versa.

Similarly, McCarthy et al. [29] conducted a multiyear study, introducing a CMS in a large organisation to understand how it affected writing practices. The results were not anticipated, as users implemented it in new and creative ways. In addition, practitioners had different experiences induced by technical and social factors. Thus, the authors argue that technologies benefit from flexibility, enabling writers to be innovative and adapt tools based on social and environmental needs.

One of the benefits of CMSs is the ability to store much structured and unstructured data. However, research also demonstrates its technical constraints. Moreira et al.

[31] discuss the challenges that emerge once data increases. Currently, there is no intuitive way to contextualise data in stored information. As a result, the systems do not meet user requirements and become increasingly complex to use. Furthermore, a large number of data reduces the chance of retrieving valuable information. Thus, Moreira et al. propose implementing data contextualisation into a CCMS by supplementing data with semantic information. The lack of context has also been addressed in studies about achieving multilingual quality in information products. For example, the literature review conducted by Batova [32] shows that research suggests that providing practitioners with sufficient contextual data implies either providing translators with complete information rather than granular content or utilising metadata, terminology databases, and style guides.

While Moreira et al. [31] explicitly focus on search systems within content management systems, Weitzman et al. [33] adopt a broader perspective and present a case study exploring the Franklin CMS developed by IBM's technology group. While initial responses to the implementation were that it was complicated, the perception changed once users were allowed to familiarise themselves with the system. Franklin was specifically designed to improve content reusability and streamline content management to ensure consistency in content. Similar to this thesis, the authors base their research on a specific use case but derive a set of challenges and solutions that can be applied externally for content management systems. Users particularly appreciated Franklin's ability to publish content without developers, facilitate publishing data in all formats, streamline content updates and ensure the content is changed in multiple places, previews, and share content fragments.

While the relevance of component content management has grown in practice [6], limited research exists on the topic. Thus, Andersen and Batova [5] call for additional research studies within component content management that address research questions based on practical implications.

3.3 Towards UX writing

Technical communication is generally defined as simplifying the complex, explaining how something works in plain and understandable language [34]. It generally focuses on writing support documentation to help people understand how to use complex systems. However, LaRoche and Traynor [34] emphasise that user-centred design processes have evolved the technical communication role, requiring them to move closer to front-end developers and UX teams. Furthermore, they explain that writing has become more than documentation and support guides, encompassing every text string within a system and must align with the brand, goals, and users. Thus, the authors suggest that there is less demand for simply authoring content and instead seeing technical communication as a fundamental part of development processes and shifting towards developing content strategies and encompassing user experience.

While the role of technical communication is still highly valuable, there is an increasing demand for content strategy and user experience skills [34]. Therefore, terms such as UX writing have emerged and sometimes replaced technical communication within organisations. Text plays an essential role in communication between users and the system. Content needs to be accessible, purposeful, concise, conversational, and clear [13]. Within clarity lies consistency. Terms and wording must be consistently used when referring to a specific concept or function. The biggest hurdle in achieving consistency, according to Podmajersky [13], is having several people writing content separately. Thus, there needs to be a standard guide to align all the different departments as text management processes directly impact the quality of the text.

Research further highlights the corporate relevance of UX writing [35]. Visual factors are insignificant if text elements are grammatically incorrect, confusing, or misleading. Thus, if all factors except text have been carefully considered, research highlights that users will likely still have a poor experience. Users should always feel informed regardless of the situation. Well-written text guides users throughout a system and provides information facilitating enlightenment and intuitive task completion. Neglecting text ultimately leads to confusion and causes unnecessary friction [35], and a bad user experience impacts the overall success of a product.

Podmajersky [13] highlights three critical steps in the UX writing process: drafting, reviewing, publishing, and tracking. While Podmajersky presents best practices in the current state, it is apparent that there are issues within the approaches. Podmajersky points out that no one tool currently fits all. Thus, most UX writers use various tools for different parts of the process. Furthermore, teams are involved in different stages of the process and may not have access to or knowledge about the various tools. As a parting note, Podmajersky mentions that he hopes there will be more tools for UX writing and collaboration that encompasses more parts of the process. Thus, more than simply establishing guidelines regarding UX writing is required as there is a leap between theory and practice.

While research is yet to address such problems, it is a discussion that can be seen within the design field [36, 37, 38]. Several people are experiencing challenges with a lack of tools that adequately cater to their text management needs, leading to inconsistencies, delays, redundancies, and confusion. They express a need for one source of truth as several tools ultimately hinder workflows. They speak about streamlining workflows and looking for better ways to handle their text strings. Thus, the issue can be seen outside of this project's specific case; however, it is not apparent within the research. Each text string throughout a system impacts the user experience and is part of a larger conversation with the users. Thus, there is a need to highlight the practical implications of UX writing in research, especially since it is a growing area and something more and more organisations are paying close attention to.

In summary, UX writing is a relatively new term within UX. However, it has emerged as research highlights the impact of text on usability. While the formulation in itself

is essential, it is evident that text management processes affect the possibility of implementing well-written and consistent text. Thus, the process must be addressed before addressing the content. Furthermore, issues in the content do not necessarily imply a lack of knowledge regarding UX writing. Instead, the culprit is generally a lack of shared understanding among teams and issues stemming from text management processes. However, while research shows a need for establishing standards and guidelines within organisations, there needs to be more research regarding the text management process and how to implement such guidelines and ensure standards are being followed. Research is especially needed regarding how to manage text content within organisations that do not use any form of content management systems and instead rely on code repositories.

4

Theory

The following section introduces this study's fundamental approaches, concepts, and frameworks. First, user-centred design is introduced, followed by the design thinking process. This study adopts a design thinking process as it is a way of remaining focused on users and implementing User-centred design. Then, user experience is introduced, followed by a set of principles. Design principles are accumulated by researchers and practitioners intended to guide design decisions and base evaluation. They help designers remain focused on critical features affecting overall user experiences. Finally, atomic design is presented and discussed as a design system to streamline design work, ensure consistency, and promote interface understandability.

4.1 User-centred design

UCD is a philosophy more than a technique, dependent on an interplay between users and designers. Users are a fundamental part of UCD, informing design decisions. Thus, they must be involved throughout the entire process.

Sharp et al. [39] describe UCD as encompassing the core principles of philosophy, empirical measurements, and iterative design. Philosophy refers to the early phases of a design process concerning understanding users and their goals. Philosophy emphasises the importance of understanding users through observations in their natural environment and involving them throughout the design process. Furthermore, empirical measurements refer to evaluations throughout a process. Finally, iterative design describes the importance of continuous iterations to improve solutions and model the design process together with users. The principles are pillars of UCD and operate together to ensure a well-suited product for the users. Valuing the principles of a UCD approach is critical to ensure solutions are shaped according to users' needs [39].

4.1.1 Design thinking

Design thinking (DT) is a user-centred iterative approach, particularly advantageous when dealing with undefined or unknown problems. The design thinking process encompasses each pillar of UCD through its five phases: *Empathise, Define, Ideate, Prototype,* and *Evaluate* [14]; figure 4.1 provides an overview of the DT process.

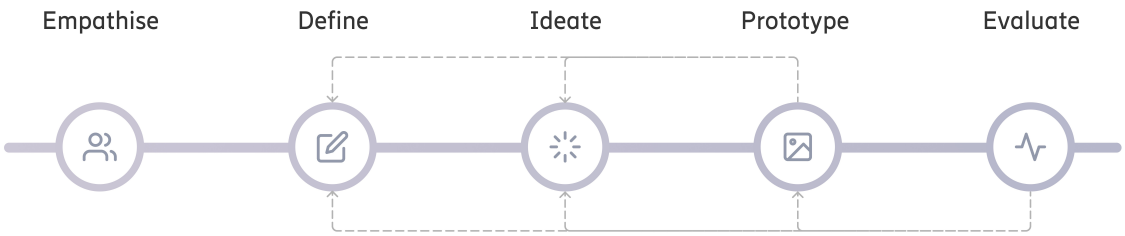


Figure 4.1: The design thinking process.

At its core, DT focuses on users and fosters innovation by constantly challenging assumptions and design in unity with user needs and goals [15]. Thus, empathy becomes an integral part of the process. Empathy is formed through ethnographic observations of users in their natural environments. Empathy helps practitioners to understand user needs and behaviours. Thus, enough time and effort must be devoted to empathy to gain genuine insight into a problem. A bottom-up approach is essential to avoid bias and hinder practitioners from rushing to conclusions while information is still missing. The brain replaces missing information with anticipated, causing bias and assumptions [15]. Empathy lays the foundation of an entire design process; ill-defined problems will ultimately affect the outcome. Thus, insights generated from empathy must inform each of the following phases.

Butler and Roberto [15] highlight the importance of continuous iterations to deepen the understanding of users and improve solutions based on user data. Designers must be open to ideas changing throughout a process based on new insights to avoid early fixations. "Fail often and early" is a pillar of the DT mindset [14]. Failed ideas should be discarded rather than rationalised. Thus, much effort should be put into the early phases. Avoiding becoming attached to initial ideas is imperative as it may otherwise skew the design in favour of designers' opinions rather than user needs [15]. The number of iterations depends on the outcome or organisational constraints [14]. Iterative design is imperative to the process; the more iterations, the more profound understanding of the problem space and the users. However, designers must also identify when it is suitable to move on from one phase to another, as "design never ends" [14].

DT goes beyond applying and following the steps; it is equally a mindset and toolbox [14]. DT helps practitioners identify strategies and solutions "that are not apparent with your initial level of understanding" [15]. DT helps teams uncover what Brenner and Uebernickel [14] call "hidden needs", that are not directly evident.

4.2 User experience

User experience (UX) is a widely used term and, thus, is as widely understood. It requires shifting focus from simply human subjects to real users, enabling designers to make products that meet users' needs and fit into their everyday lives. From

a business perspective, UX is critical in a competitive landscape where users are highly aware and can easily switch to a better solution.

Usability is closely connected to UCD and is vital in ensuring a good user experience. Usability refers to an interactive product's usability from users' perspectives [39]. While usability is fundamental to any design, UX goes beyond functionality and design. It bridges emotional, social, and cultural territories [40]. UX encompasses pragmatic and hedonic factors, combining usability measures with fun, enjoyment, and aesthetics [41] [40]. UX design can be highly complex; it requires understanding how users behave and think, considering mindsets and contexts closely. A system can provide a comprehensive solution with all necessary functionalities; however, the solution becomes available through an interface, becoming the primary artefact delivering experiences [40].

4.3 Guiding principles

UX design is complex. However, design principles can make the complex more tangible. While following design principles will not guarantee a good user experience, they can support design teams throughout the process by providing a guide and a way for designers to stay focused on the most pressing issues [42]. In summary, design principles express the importance of particular elements when making design choices. They also facilitate a common understanding among individuals who work together on a design project.

Principles can be general or more specific, and design teams should focus on specific principles based on goals, user needs, and identified problems [42]. This thesis utilises design principles discussed by Benyon [43], addressing core concepts that impact usability and UX.

4.3.1 Design principles

Design principles work together to promote usability and improve user experiences. Benyon [43] groups the principles according to *learnability, ease of use, safety,* and *accommodation.*

4.3.1.1 Learnability

Learnability concerns users accessing, comprehending, and remembering the system. It considers how easily users can accomplish specific tasks when they first use an interface, how many attempts are required to become proficient at the tasks, and becoming proficient and familiar with the overall system. Learnability comprises the principles of *visibility, consistency, familiarity*, and *affordance.*

Visibility exemplifies that when an element is visible, users are more likely to be informed and know what steps to take next. Consequently, if an element is hidden, it becomes demanding to discover and comprehend, ultimately impacting usability. Thus, ensuring elements, information, and actions are visible is fundamental to assist

users in staying informed about options available within a system. While visibility seems simple, applying the principles can be challenging as everything within an interface cannot be made visible. Therefore, visibility requires considering trade-offs and prioritising elements according to user needs.

Consistency refers to being uniform in design choices among similar tasks. Similar tasks should be performed with similar operations and elements. Consistency in similar elements or actions, both in terms of industry standards and throughout an interface, impacts users' ability to rely on their assumptions about an interface. Thus, ensuring consistency in features, interactions, language, and visual design is imperative to promote usability and learnability. Furthermore, it ensures a sense of coherency throughout different screens within an interface.

Familiarity concerns speaking the user's language throughout a design. Designers must use terminology that users are familiar with to promote comprehension and learnability. However, familiarity cannot be assumed. Thus, it must be uncovered through user research to understand experiences and users' mental models. Moreover, maintaining iconography and similar to commonly known metaphors and standards is important to promote familiarity further.

Affordance refers to designing elements in a way that clearly communicates their purpose. In turn, considering affordances makes designs more intuitive. Affordances further support interaction by giving hints about object properties so that users know how to interact with them. Strong affordances ensure clarity in intended use without extra information or context. Thus, it is important to make interactable elements perceptually comprehensible.

4.3.1.2 Ease of use

Ease of use is a central concept within usability and concentrates on ensuring users feel in control and confident. Interfaces should promote users' ability to know what to do and how it should be done. Ease of use is closely related to efficiency and comprises the principles of *navigation*, *control,* and *feedback*.

Navigation emphasises the importance of providing users with a logical path and ensuring navigation is aligned with design goals, user needs and expertise. It is important to support users in moving around a system, including providing information about location and directional indications. Users should be able to easily predict how to navigate the system to avoid feeling lost or confused. It should also be easy for them to retrace their steps if needed.

Control emphasises that users should feel in control of their experience; it allows them to use the system according to preferences and needs and reduces confusion. Users should be able to utilise the interface according to their goals. Thus, they should never feel forced into specific actions or feel that their intentions do not match the outcome. Providing a clear and logical relationship between controls and effects further encourages control. The mapping should be intuitive to ensure it is easy to understand the impact different actions may have. Furthermore, there must

be a clear link between how actions within the system affect things outside of the system.

Feedback refers to the communication between the system and the user. It must be clear what effect actions had and inform users about what is happening within the system. Users must always be aware of whether their action was performed and what result it had. Feedback is another way to promote users' feeling of control, and it must be continuous, consistent, and transparent. Furthermore, feedback must be instant to clearly communicate what action it refers to.

4.3.1.3 Safety

Safety extends ease of use and efficiency, highlighting the importance of safety and security. While a system must be easy to use, it must also support users from accidentally performing actions with problematic consequences. Safety comprises the principles of *constraints* and *recovery*.

Constraints can be used to prevent users from making mistakes or performing unsuitable actions. Constraints can be utilised to limit the number of permissible actions to guide users while effectively avoiding problematic consequences appropriately. Ignoring constraints within designs can result in undetected issues or confuse users, as all options are available to them regardless of context. However, when applying constraints, designers must ensure that enough information is provided to avoid ambiguity. While systems should prevent users from potential errors, errors and mistakes are still inevitable. Thus, it is imperative to ensure that users can quickly and efficiently recover from potential mistakes.

4.3.1.4 Accommodation

Designers must always design to accommodate users by considering their needs, experience, and expertise. Accommodation emphasises the importance of designing to suit the users, support them, and align with several different use cases. Accommodation comprises the principles of *flexibility, style*, and *conviviality*.

Flexibility implies allowing users to tailor their experiences. A system can be used by several different users with different intent. Thus, designers must consider different use cases and cater to users with different levels of expertise and knowledge about the system. Users should be provided with different ways of performing tasks to tailor their experience according to their preferences. However, providing an excessive number of different ways of performing the same tasks may result in confusion. Thus, implementing flexibility requires knowing the users and understanding their needs and expectations.

Style refers to the need for designs to be aesthetically pleasing. However, what is considered aesthetically pleasing can be highly subjective. Thus, the principle focuses on utilising visual design to support users' primary goal rather than distract from it. Decisions about aesthetics should always stem from a design's purpose and align with the overall goals. Nevertheless, a general guideline is to focus on minimal-

istic design. It is important to focus on essentials to ensure users are not distracted or hindered from obtaining information and performing tasks. Furthermore, it requires prioritisation to omit irrelevant content without losing context and confusing users.

Conviviality emphasises that interactive systems must be considerable and friendly. However, it does not necessarily mean the language must be overly polite and friendly. Instead, it refers to how well a system supports its users. A convivial interface is intuitive and predictable. It is at its service to the user by staying in the background without interrupting. Instead, it supports users in solving problems or concerns while helping them understand by giving hints and explanations.

4.3.2 Psychological principles and interface design

Design is significantly impacted by psychology as it affects overall user experiences. Therefore, considering psychology and human limitations is imperative in empathising with users and understanding how it may influence design decisions. Psychological principles supplements design principles with their explicit focus on designing through the lens of human behaviour, considering cognition and perception. The core psychological principles are *memory and attention*, and *perception* [43].

4.3.2.1 Memory and attention

UX design is limited to human abilities. Designers must consider users ability to remain attention and remember. Memory and attention focus on *recall and recognition*, *working memory*, and *grouping information*.

The concepts of recall and recognition are closely linked to visibility, highlighting the significance of promoting recognition in interfaces. It is easier for humans to remember things through recognition rather than having to recall them from memory. Designing to reduce memory load requires making actions and information visible, ensuring users do not have to remember them from one part of the system to another. Thus, relevant information should be easily accessible so users can recognise it when needed. Similarly, in-context help allows users to be supported while doing instead of being required to memorise information or instructions.

People's working memory, or short-term memory, is limited. Thus, designers must consider the capabilities of the working memory and refrain from providing too much information at once. Furthermore, options and possibilities should be limited to ensure users are not overwhelmed. Grouping information into larger purposeful entities reduces memory load, making information less demanding on the working memory. However, it is essential to consider what information is suitable for grouping. Groups must be meaningful to ensure users understand their context and purpose. Grouping information also allows strategically hiding information without losing context or requiring users to recall. It further promotes recognition by providing users with just enough information to remember it without putting additional demands on working memory.

4.3.2.2 Perception

The principle of perception builds upon Gestalt theory and the Gestalt principles of design, focusing on *proximity*, *similarity*, and *continuity*. Considering the principles of perception allows designers to structure content to align with human tendencies that will impact most users.

Elements placed in proximity to each other are perceived as belonging to a whole. Thus, related items should be placed in proximity to each other while separating unrelated items. Proximity is closely linked to grouping information, giving users visual cues about relationships between actions or information. It makes it easier for users to scan and comprehend an interface, finding the required content. The human eye perceives relationships in the same way through visual groupings. Similar elements are perceived as belonging together. Similarity can be characterised by visually distinguishing unrelated elements from each other and designing related items visually uniformly, using size, shapes, or colours.

While grouping content and considering proximity is imperative for visibility, designing for a continuous flow of elements rather than an interface compiled of isolated objects is important. The human eye tends to follow lines and paths. Thus, elements that are placed within a distinct path or line are perceived as sequential and connected.

4.4 Atomic design

Design systems are standards for organising and managing design at scale through components and patterns [44]. It is beneficial for creating a shared understanding between designers and achieving visual consistency across screens. Furthermore, design systems make the design process more efficient and reduce redundancy. As a result, design systems are commonly used in organisations to streamline design work.

Atomic design is a method for building design systems, founded by Brad Frost [45]. It adopts a modular approach to design, providing a structured way of creating UI designs. The concept is a metaphor for chemistry and the study of matter and its composition. The universe is made of matter, and atoms make up everything. Frost proposes that the same can be said about interfaces. Atomic design comprises five stages: *atoms, molecules, organisms, templates*, and *pages* (Figure 4.2). While the approach is structured and created to design systems hierarchically, it is not intended to be a linear process. Instead, Frost describes it as a mental model, a way to view interfaces in their entirety, and a collection of elements.

Atomic design can facilitate prototyping. There are several benefits to the reusability aspect, such as reducing time and effort, as elements can be reused across an entire system. Furthermore, adjusting and updating components is easy as changes are transferred to all instances. It simplifies making more minor changes after, for example, user testing. The approach further simplifies scaling a design and makes

it easier for developers to transfer designs to code. Atomic design further improves consistency, making atomically constructed interfaces more understandable.

Atoms are the primary foundational building blocks of an interface. Atoms include, for example, buttons, icons, and inputs. Molecules are groups of atoms working together as a unit. A molecule can be, for example, a search form comprising a label, an input, and a button. While atoms can be abstract, they gain purpose when combined into a molecule. Organisms are more complex UI components. They can consist of molecules, atoms, and other organisms. Headers are an example of organisms composed of the search form molecule, a logo atom, and a navigation molecule. Templates are created with different organisms, molecules, and atoms. In summary, templates are wireframes focusing on content structure and layout. Templates give context to the different elements. Finally, pages are visual representations of an interface, a mockup comprised of all its parts. While templates are solely focused on structure, pages represent the final UI. Templates can be reused and altered for different pages.

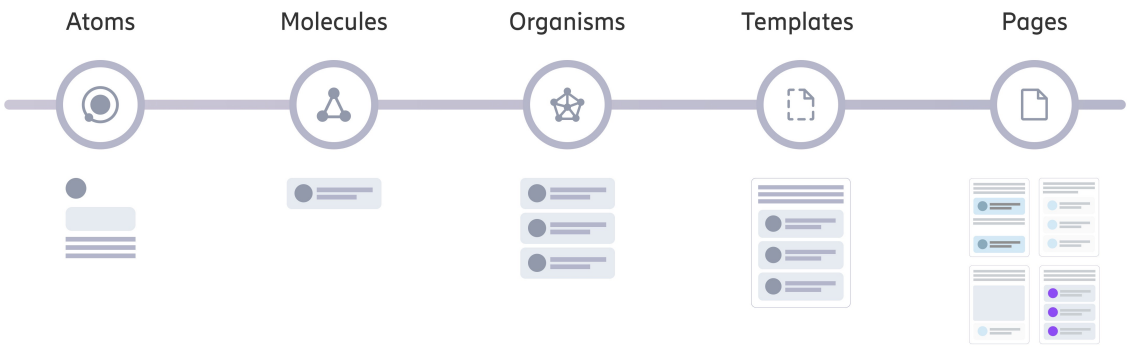


Figure 4.2: Components of Atomic design.

5

Methodology

This chapter details the methodological approach used throughout the studies, structured according to the design thinking process: Empathise, Define, Ideate, Prototype, and Evaluate.

5.1 Empathise

Empathy is the foundation of a UCD process [46]. The goals include understanding users' needs, thoughts, behaviours, beliefs, and emotions. Empathising with users and what they perceive as important is crucial to design solutions tailored to them. Empathising is made possible through observing, engaging, watching, and listening to the users.

5.1.1 Semi-structured interviews

Qualitative interviews facilitate a first step towards gaining empathy by gathering data about users' experiences, opinions, attitudes, and perceptions [47]. While there are four primary forms of qualitative interviews: structured, unstructured, semi-structured, and group interviews [39], the study's purpose determines their applicability.

The purpose of conducting interviews within this study includes identifying pain points within current text management processes. Furthermore, it is necessary to gain a deeper understanding of the nature of collaboration and communication, to understand the context in which the pain points arise. Thus, a semi-structured approach is most appropriate for this study, as it allows for addressing specific topics, for participants to elaborate their answers [39], and the opportunity of exploring new topics revealed from unexpected answers [43].

Expert sampling is a common strategy for qualitative interviews [20]. Experts are knowledgeable within the subject area and hold information, experience, and insights critical to the problem. Furthermore, expert interviews are suitable, for example, when working with an internally facing problem.

5.1.2 Competitor analysis

A competitor analysis compares a product under development with current competitors. The goal is to identify how a product is unique while understanding how other products are better or worse [48]. The competitor analysis aims to identify similar software, analyse their functionality and investigate their advantages and disadvantages. Researching and observing existing systems or products can also be a source to inform requirements [43, 39]. In this thesis, competitor analysis is focused on drawing inspiration to solve the internal problems identified and identifying gaps in current software rather than using the analysis to gain a market advantage. Two types of competitors can be analysed: direct and indirect. Direct refers to products that are the same as your product, and in-direct refers to similar products [48].

5.2 Define

Define involves establishing the design scope and focus. A problem space definition [46] or question [14] can be a guiding statement. Nevertheless, the goal is to establish a shared understanding, gain clarity by determining a viewpoint, and identify design requirements. Most importantly, it should be exclusively derived from the insights gained from empathising. However, a defined scope and focus requires analysing, digesting, synthesising, and presenting the gathered data in an understandable way [43]. Making sense of information is essential before framing a problem and identifying requirements [46].

5.2.1 Affinity diagram

Once data is collected, it must be analysed to identify themes and categories, leading to a better understanding of the domain [43]. Semi-structured interviews allow for a defined scope while simultaneously being exploratory and giving room for elaborate answers. Thus, affinity diagramming is used as an analysis technique, as it is a highly adaptive method for clustering large amounts of unstructured, qualitative data [40], facilitating revealing patterns and relationships [49]. It is suitable in multiple scenarios for collaboratively organising research findings or design ideas [50].

The process includes stating observations and insights on sticky notes to identify patterns and cluster them together [47, 49]. In addition, aligning a clustering logic through a sample sorting [49], often based on insights sharing similar intents or issues [47], is beneficial. Patterns can be defined once all data has been clustered as it reveals relationships, which in turn facilitates identifying themes. Analysing patterns enables a better understanding of the problem space while paving the way for concept generation in parallel [49].

Affinity diagramming is an inductive approach where inferences are made based on data rather than an established framework [47]. Thus, it enables exploring insights and patterns that may have been otherwise overlooked or shaped to fit a pre-determined theme.

5.2.2 Empathy maps

While affinity diagramming promotes discovering general themes in data, empathy maps visualise these findings related to a specific user or segment [51]. Empathy maps conceptualise what users say, do, think, and feel. Typically, these categories are visualised in different quadrants, with the user placed in the centre. Such externalisation facilitates synthesising data and acquiring a deep, shared understanding of the users, which is critical to the DT process. Empathy maps can further protect against bias and assumptions if continuously referred to throughout development. Accordingly, it can facilitate decision-making and prioritisation of user needs.

5.2.3 Defining and prioritising requirements

The knowledge gained from research is valuable in informing requirements which define and communicate expectations regarding a product [39]. Requirements specify expectations on functionality and performance. They can be informed through different methods but must be directly connected to user research. Thus, the requirements reflect user expectations and needs and facilitate keeping user needs at the centre throughout development processes. Acquiring a complete understanding of requirements demands evaluating design work [43]. Thus, establishing requirements is an iterative process; they should evolve throughout the process. Requirements are ideally coupled with success criteria, making it apparent when a requirement is fulfilled [39].

As projects have limited resources, requirements should be prioritised. MoSCoW is a technique for prioritising requirements, and it stands for *Must have, Should have, Could have*, and *Would like to have* [43]. All requirements are valuable but should be prioritised according to what will generate the most benefit [52]. Must have covers everything that must be present in the final solution. These are necessary for the project to succeed. Should have implies requirements that are not crucial for the system to work and potentially not as time-critical but are still beneficial. Could have are "nice to have" and may benefit the user experience but are not considered essential. These are desired but will not be considered if they are too time-consuming. Finally, Would like to have are requirements that are agreed to be unrealistic.

5.3 Ideate

Ideation facilitates transitioning from identifying problems to finding solutions [46]. Ideation draws from previous phases and enables practitioners to ideate within the scope and based on user needs. Thus, brainstorming should focus on finding solutions explicitly to the identified problem [14]. The main goal of ideation is exploring as many ideas as possible based on previous phases [46].

5.3.1 Design workshop

Workshops intend to involve stakeholders throughout a design process through collaborative sessions. There are several different UX workshops, including discovery, empathy, design, prioritisation, and critique [53]. Design workshops encourage rapid ideation while obtaining different perspectives. Prioritisation workshops inform decisions and build a consensus by allowing stakeholders to prioritise features according to their needs. This study leverages ideas from design and prioritisation workshops as they are valuable in understanding and informing requirements [43]. However, it is referred to as design workshop as it is the primary purpose.

Design workshops allow designers to involve stakeholders in the ideation process through activity-based research [47]. They are especially beneficial for broadening perspectives and reducing designer bias. They are most valuable when involving members within different disciplines.

Design workshops are highly flexible and can take many forms depending on the goal. This study utilises design workshops combining the KJ technique and Crazy 8. Workshop settings can be unbalanced depending on the participants. The KJ technique is quiet and, thus, allows all participants to express thoughts and ideas without group pressures [47]. The KJ technique facilitates gaining a shared understanding while grouping and prioritising ideas and challenges. It is a valuable tool, especially for meeting and workshop settings. It can help synchronise team efforts by focusing on one question at a time. A focus question must be determined before a session, ensuring that all participants are ideating around the same question.

The KJ technique involves gathering members from different teams within an organisation to obtain different perspectives [47]. Each participant brainstorms ideas and writes them down on sticky notes. Participants can write, sketch, or do anything that helps convey their ideas. Then, participants organise the sticky notes into groups. Finally, participants prioritise the groups by voting. Insights from the session are organised in an affinity diagram.

Crazy 8 is a rapid ideation method involving sketching eight ideas in eight minutes [54]. While people may find it intimidating, it is highly beneficial as it leaves people with little time to overthink or second guess. It pushes people to think beyond their current ideas and biases. Furthermore, it encourages exploring as many ideas as possible, good or bad, promoting innovative ideas. It is a method intended to help people silence their inner critique. The sketches should not be overly detailed as it may result in the opposite effect, and people get stuck in thinking an idea must be well thought out or perfectly visualised. Instead, each sketch should communicate individual ideas in a way that suits the person.

5.3.2 Brainstorming and sketching

Transitioning from problems to solutions can be challenging. Brainstorming techniques can facilitate the transition by generating, exploring, and refining many ideas [47]. The sessions may include mind mapping, sketching, brainwriting, and

rapid ideation. Furthermore, thinking of quantity over quality is important. Initial ideation sessions should be open without judgement; it is easier to build on bad ideas than no ideas.

Sketching is one of the primary methods for low-fidelity prototyping, focused on exploring ideas with low investment [39]. Sketches are quick to produce, cheap, and allow designers to generate ideas as they sketch. In addition, sketches facilitate a shared understanding and gain a consensus on design direction. Design thinking highlights the importance of exploring several ideas and not letting cognitive biases lead the design. Thus, sketching is beneficial as it allows exploring multiple ideas without risking early attachment. However, similar to the quantity over quality brainstorming mindset, sketching should not be focused on perfection; they are made to be discarded [39]. Thus, both brainstorming and sketching are good approaches within the DT mindset. Sketching should be focused on exploring as many ideas as possible, visualising user research, and externalising thoughts [55]. Solution sketching, in turn, focuses on expanding on one idea or a combination of ideas and producing more detailed sketches [56].

5.3.3 User flows

User flows are flowcharts utilised to model interactions, visualising how parts of a system work together and how users move through the system to reach their goals [57]. While task flows are generally linear, modelling a high-level path and actions for specific tasks, user flows model all available paths to reach a specific goal. Thus, they include user actions, decisions, and optional paths.

User flows are beneficial for ideating and exploring different interaction possibilities [57]. Modelling user flows is essential in understanding how users will interact with a product, especially when designing a complex system accommodating several users and use cases. They are beneficial as they account for all possible interactions for a specific user journey and goal. Furthermore, they permit identifying potential issues before prototyping solutions to ensure users can complete their goals intuitively and efficiently. Lastly, it is an important communication tool as the intended interaction, actions, and flow can be presented to stakeholders before having a semi-functional prototype.

5.4 Prototype

Prototyping is about realising ideas into design artefacts [46]. Prototypes can take many forms; however, users should be able to interact in some way with them to base evaluation. Early prototypes should be cheap and quick to make to gather early user feedback. Early prototypes are essential and will address broader questions, whereas higher fidelity prototypes address specific questions. Accordingly, it is essential to determine what is being evaluated clearly. Prototyping is a powerful tool for starting conversations, communicating ideas, and testing possibilities.

5.4.1 Wireframes

Wireframes act as a blueprint of the final product and focus on information architecture [40]. They include structural layouts and navigational flows of each screen. Wireframes are a form of prototypes and an important asset for interaction designers. They are high-level conceptual designs without any visual content. While wireframes are far from the final prototype, they are a means of communication and evaluation, allowing designers to evaluate conceptually before visually. Thus, there is room for continuous iterative improvements [39].

5.4.2 High-fidelity prototyping

High-fidelity prototypes closely resemble the final product's appearance and functionality [43]. High-fidelity mockups are static representations of the final product [39]. They resemble the final product visually without any interactions. High-fidelity prototypes are, in turn, interactive mockups of the final interface [40]. While low-fidelity prototypes are beneficial in evaluating concepts and structure, high-fidelity prototypes are refined and allow several design factors to be evaluated, such as aesthetics, usability, interactions, flow, etcetera. They may result in a more natural reaction from users than low-fidelity prototypes [39]. However, they are only beneficial in the later stages of a design process once low-fidelity prototypes have been evaluated and iteratively improved.

5.5 Evaluate

The evaluation phase includes evaluating concepts and prototypes with users. Evaluation can and should be conducted throughout the design process [46]. Furthermore, evaluations inform the subsequent iterations of prototypes. Evaluation aiming to improve and refine a design iteratively is called formative evaluation. Formative evaluation encourages a deeper understanding of the user by testing what works or does not work with a design [58], how users interact with it, and most importantly, why [46]. Gathering qualitative data to identify problems, in turn, inform refinements [40].

5.5.1 Expert review

Expert reviews are effective and important in the early stages of design processes [40]. Expert reviews are also referred to as UX inspections. Hartson and Pyla [40] emphasise that while a user experience cannot be inspected, designers should aim to inspect a design to uncover potential user experience issues. Thus, expert reviews are beneficial as reviewers can detect factors that may impact the design before presenting a solution to the users.

Furthermore, expert reviews are efficient as they generally require fewer participants, typically two or three. However, experts may still find different issues and have different perspectives. Thus, considering diversity in experts are more important than the number [40].

Lastly, experts should not replace users in evaluation [43]. Expert evaluation and usability testing should be combined as it results in the best overall design. However, while participants throughout this study are experts within the field, they are also the ones who will use the solution. Thus, they can evaluate the solution based on their expertise and prior experience in the field while providing insights about how it aligns with their needs and solves experienced challenges.

5.5.2 Design walkthrough

Expert reviews can be conducted in several different ways. This study utilises design walkthroughs as they are flexible and can uncover potential problems by addressing user flows, intended features, and design guidelines.

Design walkthrough is a rapid evaluation technique that can be applied at almost any stage of the design process [40]. Rapid evaluation is imperative in the early stages of a design process when concepts and solutions are constantly changing. Thus, there is no need to invest resources into a comprehensive evaluation. Design walkthroughs allow design teams, UX analysts, user representatives, and potential users to explore the design on behalf of the users. It is advantageous as participants walk through the design with the design team, encouraging continuous discussion. Thus, several forms of prototypes can be utilised in design walkthroughs, such as sketches, wireframes, user flows, scenarios, etcetera. Simulating critical scenarios and user flows can facilitate predicting usability concerns. Furthermore, it is an opportunity for questions to arise regarding the emotional impact of a design. Compliance with design guidelines and style guides can also be addressed in this session [40].

6

Process

This chapter presents the entire process and details each of the studies sequentially according to the process. The sections are structured according to the methods applied and discussed in the methodology chapter and describe the process and findings of each method. Furthermore, the final iteration of requirements is presented. Lastly, the prototyping phase is described, including models, low-fidelity prototypes and evaluation, followed by a presentation of the semi-functional prototype.

6.1 Interviews

Semi-structured interviews initiated the empathy phase. The interviews were held with representatives from different teams at Dedicated Networks and covered responsibilities for the NMP, collaboration and communication in general, and current ways of working with text. See Appendix B for the interview protocol. The main purpose was to understand the nature of collaboration and communication, gain a deeper understanding of current text management processes, and explore experiences, perceptions, attitudes, and opinions. Thus, the goal was to answer the first research question and act as a foundation for establishing requirements.

6.1.1 Pilot study

Pilot studies can be conducted to test the suitability of a chosen data collection method or the execution and arrangement of it [59]. Thus, a pilot study was held prior to the interviews to examine whether the interview protocol aligned with the interview purpose, if the questions were relevant and understandable, and establish whether the estimated time frame was realistic. Pilot studies should be conducted with a smaller scale of participants corresponding to the real target group [59]. Thus, the pilot study was conducted with a participant from the UX team who did not participate during the semi-structured interviews. The pilot study resulted in some rearrangements and reformulations of the questions, whereas some were removed as they were redundant or did not support the interview purpose.

6.1.2 Participants

Experts can provide valuable insight into how things work within an organisation; problem causes, what solutions have been tried, and what challenges they brought [20]. All of these are important to inform requirements. This thesis investigates an internally facing problem and the interviews included experts from Ericsson Dedicated Networks. People from UX, CPI, FE, and BE were contacted with a short description of the study, and they were all willing to participate. Eight people were interviewed in total (Table 6.1), including four people from the UX team, one person from the BE team, one person from the CPI team, and two people from the FE team. Participants included team leaders, scrum masters, and UX designers.

Table 6.1: Interview participants.

ID	Team
P1	UX
P2	UX
P3	UX
P4	FE
P5	UX
P6	CPI
P7	BE
P8	FE

6.1.3 Procedure

Understanding challenges from different perspectives is necessary as teams have different established processes. Nevertheless, this study is delimited to support the UX process primarily; thus, the interviews focused on the areas where team processes intertwine with UX. All interviews were held in person at the Ericsson office, ranging from 30 to 50 minutes. All interview audio was recorded to enable transcription, facilitating analysis. Before each interview, participants signed consent forms, scanned and emailed to them afterwards; see Appendix C for the consent form.

The interviews were held either in English or Swedish due to language barriers and to ensure participants felt confident to elaborate and discuss. All interviews were based on the same interview protocol, in English or translated into Swedish. However, it became apparent throughout that different participants had different perceptions about text and current processes. Thus, some interviews were more unstructured than others. Nevertheless, the study scope guided all eight interviews and semi-structured interviews allowed room for insights not previously considered.

6.1.4 Analysis

First, each interview was transcribed according to intelligent verbatim. The interviews were highly discussion and experience based. Thus, they included filler words and repeating sentences as participants thought and reflected while speaking. Intelligent verbatim was the best approach as it permits removing filler words such as "um", "like", etcetera. Furthermore, sentences can be divided into shorter ones to improve readability and understandability [60] if the meaning is not lost. While intelligent verbatim allows for removing irrelevant discussions, they were not omitted as they may prove useful during analysis. All interviews were transcribed manually to promote further data familiarisation.

Once all interviews were transcribed, insights were highlighted and transferred to sticky notes. Then, similar notes were grouped until all insights belonged to a group. Affinity diagramming allows clustering insights to derive themes. The analysis adopted an inductive approach, allowing data to speak for itself [61]. No predetermined groups or requirements were established prior to the analysis. The analysis progressed over several days. Thus, the insights were reviewed and re-structured iteratively to ensure that several possibilities were discussed. Utilising affinity diagramming facilitated the iterative approach to analysis as sticky notes are easily moved and discussed. Initially, insights were clustered into larger groups and then revised to discover sub-themes. The affinity diagram was refined once themes and sub-themes were reviewed and finalised. See Appendix D for the final result.

Table 6.2: An overview of interview themes and sub-themes.

Theme	Sub-theme
Designing for enterprise users	Text impacts on NMP
	Consistency
Cross-functional environment	UX driven design
	Responsibility
	Different perspectives
Shared understanding	Slack for communication
	Guidelines for text
	Single source of truth
Text management processes	Scattered processes
	Manual text management
	Prospects for the future
Tools for text management	Perception of current tools
	The importance of documentation
	Bugs: balancing effort and value
Managing error messages	Unawareness
	Finding edge cases
	Contradicting requirements
Preparing for translation	Lack of structure
	Implementation

6.1.5 Findings

Seven themes with corresponding sub-themes emerged from the data analysis. Themes and sub-themes are presented in table 6.2, followed by a description of each theme and insights that were drawn.

Designing for enterprise users
UX are responsible for NMP and generally in charge of decisions. However, that also requires a lot of responsibility in ensuring that NMP has a certain standard and quality. Ericsson is generally very technical, and one participant highlighted the importance of having a UX perspective in a technical organisation.

> *"For many, it is easy to get caught up in the technical aspects, so you forget this overall perspective."* (P2)

It becomes especially important as NMP is slightly different from other systems developed at Ericsson. The users are not telecom experts and the NMP is, thus, not developed "for engineers by engineers". It is one of few systems accessible for enterprise users. While it became apparent during the interviews that designing for less technical users is a challenge, participants from all different teams showed an understanding and awareness of end users.

Text impacts on NMP
NMP users are different from traditional Ericsson target users. Thus, understanding how NMP impacts the user experience is important. As a result, the UX team have the main responsibility for NMP. One participant specifically stated how important it is to elicit trust between NMP and the users. Furthermore, all participants from the UX team expressed the importance of text in terms of user experience. Specifically, factors such as quality and understandability were mentioned.

> *"If the user doesn't actually understand it, or know how to use it, then there could be problems and it can lead to things like people perceiving the quality of the product as not good."* (P3)

However, as current processes are not working optimally, they also expressed concerns about the understandability and quality of NMP. Several participants mentioned that there are cases where different terms have been used to represent the same thing. One participant explained that using different terms allows interpretation and misunderstanding and can lead to distrust. Furthermore, it can lead to unnecessary contact with support and increased support tickets. In turn, support tickets cost Ericsson money. When writing error messages for NMP, sometimes "contact support" becomes the easy way out. However, UX aims to offload support and ensure users can solve some of their issues independently.

Consistency
All participants from the UX team brought up consistency as a challenge. The text must be consistent on NMP to elicit trust in users. Furthermore, NMP includes a lot of information. As a result, NMP must be understandable as it is an important tool and the main source of managing your network.

Both the UX team and CPI expressed the need to consider consistency when writing for NMP. CPI is currently working towards improving consistency by creating a terminology document including accepted, prohibited, and deprecated terms. From a UX perspective, consistency is easier to adhere to in event logs because they are all stored in the same place. However, several other areas are more challenging because of current text management processes.

Cross-functional environment
Dedicated Networks is a highly cross-functional environment. UX, CPI, FE, and BE all mention they work closely together in different stages of the pipeline; to build and implement new features, review and improve existing features, and manage bugs. FE and BE further collaborate closely with other more technical teams, to a greater extent than UX and CPI. There is a culture that encourages asking questions regardless of the topic. Teams expect input from each other and value getting different perspectives on concerns or ideas.

> *"When everyone gets to speak, and everyone's viewpoints are taken into account, that is when we get the best results."* (P2)

Thus, a general insight from the interviews was that the relationship between teams was described positively, with a lot of close collaboration. However, teams need to be more aligned regarding what to do and what decisions were made.

It became apparent that participants valued good collaboration and showed an understanding of other teams with their specialisations and work processes. While several challenges regarding different perspectives were brought up, the openness to perspectives and collaboration was apparent. One participant summarised this insight well.

> *"We can have an ideal picture of how it should work, a perfect user flow, or a perfect way to do it from a user perspective. But then there are a thousand other technical things that make it not work, it will have to be done this way instead. And there it is very important that you have that collaboration to make it work."* (P2)

UX driven design
The design process is iterative, and the UX team works closely with other teams.

> *"We don't just deliver once: here is everything now it's 100 per cent ready, go ahead. Instead, we talk to them during the process"* (P1)

While the interviews involved participants with different specialisations and responsibilities for NMP, all participants expressed that they have a UX driven process. Meaning that there is an awareness that the project is UX driven, regardless of people's understanding of UX as a practice. It circles back to UX having the main responsibility for NMP. While all participants from the UX team expressed that they have a close collaboration with, for example, FE, BE, and CPI, they were more concerned about the technical streams where collaboration is not as close. As a result, technical teams may not have the same understanding of UX practices.

Responsibility
All interviews made it clear that the UX team has many different responsibilities. They are involved in almost every aspect of text management and have a lot of aspects to keep track of. Thus, their own processes and other teams' processes affect them in several ways.

In earlier phases, the UX team discusses a lot with CPI regarding terminology, grammar, etcetera. Furthermore, the UX team reviews almost everything CPI writes to ensure consistency with their text. UX has a strong connection to the other teams, including FE, CPI, and BE. They often act as an intermediator between FE and CPI, and CPI and BE. Some other teams' communication goes through the UX team, as they are involved in many different areas.

The UX team writes email templates that BE implements; they must also walk through and review emails and event logs before release. The UX team must also review the snack bar messages from BE. Furthermore, both FE and UX participants stated that FE develops what UX delivers. So, FE is highly dependent on UX for wireframes. However, as the UX team is in charge of updating and reviewing NMP content, they must also keep continuous communication to inform other teams about potential updates.

Different perspectives
Several teams are involved in developing NMP beyond the ones addressed in this thesis. It became apparent that different teams may see different things as important, including perceptions of the importance of text. While the UX and CPI team expressed the importance of text and how text affects user experiences, other teams, not working as closely with NMP end users, can be passive to the subject. The UX team are aware of other teams' technical focus and is not expecting other teams to be copywriting experts. However, one participant mentioned that lack of communication could lead to consequences such as information that needs to be exchanged between teams going unnoticed.

> *"Some other streams that have less contact with NMP and end users do not have the same insight that they need to explain such things to us."* (P1)

As a result, the text is automatically put on UX.

> *"We're trying to keep this together and no one else is really doing it, because it's not really in anyone else's interest. Yes, it is if you look at it as we as a group should create a good product with good quality, then it is in everyone's interest, but it ends up a bit automatically on us."* (P2)

While other teams automatically put things such as text on UX, all participants from the UX team expressed they find UX copy difficult and that they do not want to put too much effort into it.

> *"We don't have a UX copywriter, maybe that is missing"* (P5)

Telecommunication terms must be simplified to ensure that end users understand the information NMP provides. Responsibility for text can be challenging as the UX team has limited technical and domain knowledge. Participants from the UX team mentioned that they sometimes get too much power without being questioned. One participant expressed that it is not always a good thing, as they can get stuck in details without checking with the end users or other teams. Another participant mentioned that discussions can arise if technical terms are being used incorrectly.

UX is not working as closely with technical teams but highlights the importance of collaboration as they have different knowledge and perspectives. However, different perspectives can also lead to teams questioning one another's initiatives. One participant mentioned that it could be difficult to get feedback from teams other than UX and that it is generally difficult to distinguish between valuable feedback and opinions. To solve collaboration issues, one participant highlights the importance of continuously reviewing and improving ways of working.

> *"Since we are growing and getting bigger, our ways of working must be more etched in the culture. Otherwise, it will be difficult to keep us together."* (P8)

To strive towards this, UX and FE arrange workshops every release to improve their working methods.

Shared understanding

Achieving a shared understanding is key in a cross-functional environment. At Ericsson Dedicated Networks, this is being practised in different ways. Firstly, Slack is used as the main communication tool for discussion, asking questions, and decision-making. Secondly, guidelines are established on different organisational levels covering different aspects of Ericsson and its products. However, working cross-functionally has its challenges. It was expressed that there is a lack of a single source of truth, especially regarding text.

Slack for communication

Slack is the main source of communication between teams. It is used for private conversations as much as team-based and feature-based channels, within and between teams. Participants agreed it allows for a good balance between private and public chats, is an efficient and simple tool, and is quick to contact people.

However, its flexibility also implies challenges. As it is the main communication tool, much information is exchanged through Slack. It allows for getting others' perspectives on diverse subjects and can evolve into long, engaging discussions. However, reaching nearly 40 thousand messages a month can easily cause information overload, leading to lost things and challenges in decision-making.

> *"I think when it comes to harder discussions, or these things where, you know, how do you come to a conclusion, then slack might not be the best method to use in those situations"* (P3)

Some participants mentioned that decisions made in Slack are easy to link to from other tools. A disadvantage brought up was that not everyone can access the information, depending on the channel where the information has been exchanged. On the other hand, when reaching out in more general channels, messages can go unnoticed. Furthermore, one participant mentioned that some people do not see Slack as an official tool for decisions, which can cause misunderstandings. Regardless, most of the participants are satisfied with Slack as a communication tool. One participant mentions that it is rather the volume of messages that is problematic than Slack as a tool.

Guidelines for text
There are some style guides when it comes to text. However, the text is being handled in different tools; some are provided with guidelines, and others are not. Based on general guidelines established by Ericsson, the UX team have developed UX writing style guides which can be found in Figma. They mainly refer to the tone of voice and grammar. One participant mentioned that these guides are seldom referred to. However, some participants from the UX team mean that it is unnecessary, as they are supposed to write text. One participant mentioned that even though these guidelines exist, it is currently very individual when it comes to writing text.

It happens that other teams write text. One participant mentioned that there are not really any guides for developers and that it could be beneficial to introduce general guidelines. Such guidelines would have to be short and simple.

> *"[...] it cannot be too extensive, because they are not so interested [in UX copy], so they won't sit down and read everything"* (P2)

Another participant expressed that too many guidelines or "checks" can cause annoyance since there is an eagerness to publish some documents. CPI have many and more extensive guidelines for terms and substitutions for technical terms. CPI plans to put all terminology guides on Confluence, which is accessible to everyone at Ericsson.

Single source of truth
Six out of eight participants expressed the lack of overview as the primary concern regarding current text management processes. There is a lot of text to keep track of. Several different tools and unawareness of what text exists result in difficulty getting an overview. Currently, there is no universal place to find all text for the NMP.

> *"The problem is rather that there are a lot of different texts in a lot of different places, and it is difficult to have an overview of all that exists."* (P1)

> *"I don't think we have one place today where we can say, if you want to find all the texts related to x, y, you go here."* (P3)

> *"It would have made it easier for everyone to look at the same thing and that's the one in charge"* (P8)

Information can get lost between teams, and synchronising is especially important as the organisation has grown significantly over the past few years. The entire project is still relatively new and keeps growing. Thus, it becomes crucial that everyone has access to the same information. Furthermore, UX is responsible for NMP but lacks an overview of its content. As a result, the UX team must deal with many consequences, impacting both their amount of work and the quality of NMP. One participant encapsulated the consequences of not having an overview of all different texts and not knowing what is being implemented in NMP.

> *"We own how we express ourselves in the portal, but we cannot verify that we actually do it. That's probably the worst thing."* (P1)

Text management processes
Teams at Dedicated Networks have no established processes for managing text; the text is managed differently depending on the situation. Furthermore, the text is stored in different tools and requires much manual work. While current ways of working allow for flexibility, there are different opinions and expectations regarding a future way of working with text.

Scattered processes
Participants from the UX team expressed that there is currently no established process for handling text. There are a lot of different components that need to match. Communication is important as FE needs to be informed about potential updates so they can be implemented. FE must copy and paste from Figma to their JSON files. BE must copy and paste emails. UX must discuss terminology with CPI. The process currently depends a lot on the context. In some contexts, the way of working is more established. For example, emails and event logs are stored and updated in Excel. However, the process becomes scattered because of the different tools and ways of working. As a result, text management becomes time-consuming and can lead to unnecessary or redundant work.

> *"Because there's no process, or like a defined one, it kind of just depends on you kind of making it up depending on whatever the situation is. Sometimes it makes sense. You know, maybe one process doesn't fit 100% of the things. But I feel like, in general, it's quite scattered."* (P3)

Current ways of working require close collaboration. However, it can be challenging and time-consuming to align teams. Another issue that was brought up in several interviews was that it is unclear what team oversees certain texts. For example, UX is responsible for UI text which is handled in Figma. However, in cases such as snack bar messages, they do not have pronounced ownership.

There are issues in text management processes both from UX and FE perspectives. FE highlighted a lack of structure in their files. In turn, updating and adding text becomes time-consuming, and they lack an overall grip of all the text for NMP.

> *"We have like 40 files. And it's a bit of a pain, and it's just going to grow. There's just going to be more and more files. So, it is sometimes difficult to know... Where is this text written? When we get something new, where should you write this text, in which file... We don't have a good tool to get an overview of what types of texts we have."* (P8)

Manual text management
Currently, text is handled manually. The UX team must manually update Figma files, excel sheets and remember to document decisions and changes. FE is required to copy and paste text to their JSON files from the UX team's Figma files. The manual work was brought up as a pain point both from UX and FE perspectives.

From a FE perspective, manual text management becomes a pain point regarding JSON files. They currently have multiple different files aligned with the structure of NMP. Thus, they must manually find where a text should be updated and copy-paste based on UX. Manually adding and editing separate JSON files becomes tedious and results in disconnected files. However, optimally FE does not want to spend a lot of effort on maintaining JSON files.

> *"We never want to have to maintain a JSON file; we want it spat out to us."* (P8)

From a UX perspective, manual text management was considered a primary concern regarding consistency. The UX team must manually find and update terms in their Figma files and across tools. However, it is impossible to find all places where a term has been used and ensure it is being updated in all areas.

> *"Even though I only see one right now, I forget that it's somewhere else, and it doesn't update. Then you discover six months later that different terms are used for the same thing [...]. So, it's a bit this that we don't have a way to handle our text, we have to do in manually in all places. It works with a small product, but now it's larger. So, there may be a mismatch."* (P5)

NMP is large, and the Figma files are extensive. One participant highlighted that it became somewhat easier when Figma implemented its "Search and Replace" function. However, the NMP design files are separated into several files, so you must still go through each. Thus, it is time-consuming, and there is still a high risk of missing certain files. The UX team currently does not have a way to change UI text in all places simultaneously. Forgetting to update terms in certain places results in both inconsistency and users getting the wrong information.

There are multiple emails that need to be kept up to date. However, they are stored in a separate Excel sheet and not directly apparent on NMP. Thus, most participants from the UX team expressed that they are easy to forget. Oftentimes, they are not up to date, and users get the wrong emails.

Another concern was that any changes the UX team wants to make must go through FE. Therefore, regardless of how minor the changes are, everything still needs to go

through FE as they implement them. There is no way for the UX team to change the text directly for NMP.

> *"So, there is a lot of manual work today and then above all if there is something that we discover that damn I just want to change this little thing, we always have to ask the front end for help."* (P2)

Prospects for the future
Most participants shared thoughts and opinions about what they envision for the future. In general, they will keep current collaboration tools such as Slack, Jira and Figma. Moreover, as FE is not working with static text, they will keep the text for NMP in JSON files. However, teams provided several different perspectives about future processes and responsibilities.

One participant explained how they worked with text in another company, exemplifying the importance of finding a balance. On the one hand, updating text should be simple. On the other hand, it should not be so simple that you accidentally publish things on NMP.

> *"If I saw an error message, I could go into, it was a third-party tool, I could go in there, change the text, click enter and then wait five minutes and then check my phone and then it was already changed. There are dangers in that too, because you know in theory, people can make huge mistakes, but I think we can find a balance somewhere where it is not like super hard to change a small text, but you know not to easy either"* (P3)

From a BE perspective, they do not want to be affected greatly, for example, when a term has been changed. They want it to be simple for them to implement.

> *"I don't want to sit and have meetings and a decision list and stuff like that because we changed a term. I don't want to write ten lines of text for us to change one line of text. It should be pretty easy."* (P7)

From a UX perspective, participants had different perceptions of the best ways to handle responsibility. Some stated that UX should own the error messages and that all responsibility for messaging should lie on UX. Nevertheless, all participants agreed that the most important thing is that the UX team is aware of all text for the NMP and that they must be informed if other teams write messages that users will see. The main insight drawn was the need for an overview rather than full control of all the text. One participant said that it feels like they are spending too much time on having a hundred per cent control and having full control is not necessarily the best in terms of quality. Similarly, some participants stated that the UX team should not be tasked with writing every single piece of text for NMP; it should be a joint effort. However, teams should inform and discuss with UX before writing or implementing text. On the other hand, technical streams must become aware of what messages will be visible to users.

From an FE perspective, they expressed that they are open to changing their file structures. However, it was brought up that they would appreciate it if their processes and file structures could be better aligned and integrated with UX processes.

> *"FE can decide the ID and structure, while UX and docs can fill it with content. Because then we can help each other, we can tell UX that this is our structure, if we are going to have variable names in this then you do this."* (P8)

Any output from a potential tool must also align with FE's current library i18next; otherwise, a shared tool will not be beneficial.

Tools for text management
Tools are very much related to the scattered processes regarding the number of tools used and the different needs of different teams. Some text is not stored anywhere except directly in code. Text is currently being managed in different tools, where tools often handle text for separate purposes. The main tools used are Figma and Excel. Zeplin is used for delivering wireframes, etcetera. However, it is slowly being phased out and replaced by Figma. FE have a requirement that they are not using any static text. Thus, the text for NMP is stored in JSON files. FE also use the text library i18next. Furthermore, Jira is used for project management and bug reports. Teams are also using Confluence to some extent.

> *"There are a lot of texts in different places. That is why UX needs to have some kind of grip on them because all these texts are texts that our end users in one way or another consume and see in different contexts."* (P2)

Perception of current tools
Several insights could be drawn regarding the perception of current tools, having both advantages and disadvantages. The tools themselves were expressed as less of an issue and more the number of tools storing different text. One participant specifically mentioned visual communication as an advantage of Figma.

> *"Oftentimes, I think you can say the same things. But then you mean different things. So it could be good to put something together, it doesn't have to be anything very, high fidelity, but just so that you're making sure you're looking at the same thing."* (P3)

Similarly, the context was brought up as an advantage with Figma; if you can see text context, it becomes easier to understand. It also allows seeing UI components and how the text looks within a snack bar, for example. However, the UX team showed awareness that Figma is known by designers but not necessarily developers. Furthermore, Figma is a closed tool as people need access to it, which not everyone has. Figma also makes it difficult to represent all edge cases.

> *"But there are a lot of edge cases that are more difficult to represent there. So sometimes I would like to have more concrete things with... like OK for up it is like this and for down it is like this."* (P4)

Related both to edge cases may be missing from Figma files and how developers access components and their different states.

Another disadvantage of Figma was the lack of documentation and difficulty structuring. Figma is flexible but requires a lot of effort to apply a structure and a structure that everyone can follow. Excel, in turn, is accessible to all teams and forces the structure that Figma lacks. While Excel may not be the most desirable solution, one participant stated that it is now the best tool.

> *"Excel sheets are the best thing right now because everyone has access and knows how to use them. Then it's not very fun to use Excel, but the functions are there; you can search and comment."* (P5)

However, several participants from the UX team expressed that it is difficult for UX to verify whether what they put in Excel is implemented.

The importance of documentation
Documentation came up in several interviews. Participants highlighted the importance of documentation in order to find and follow decisions. However, they also expressed concern about a current lack of documentation. Tools such as Figma and Slack are not optimal for documentation. While many decisions and discussions happen in Slack, they can be difficult to follow. In contrast, one participant discussed Confluence; it is good for documentation and accessible to everyone. However, it becomes another tool to keep updated and unsuitable for all contexts.

Teams use Jira as their planning tool. Documenting in Jira is important to ensure nothing gets lost. Jira should include a list of all decisions, which can be accompanied by a link to a Slack thread where decisions were made. Thus, it allows documenting what and why. However, Jira is generally not updated, making it difficult to find a concrete list of decisions and know about changes that were made. It is not always clear when a term has been changed for the NMP.

> *"There [in Jira], it must also be documented which decisions were taken. As an organisation, we're not great at that, so it's often very difficult to find at the end of something a list; this is what was actually done. Because it's a bit spread out right now."* (P1)

Furthermore, Jira tickets are used to report improvements or bugs. The general consensus was that they are good regarding documentation, as you can follow decisions and backtrack.

The lack of documentation results in confusion about what was decided and how things should be done. It can lead to redundant work and unnecessary work solving issues that did not need to occur in the first place. Overall, it becomes time-consuming. Furthermore, a lack of documentation affects NMP users as it is deployed with inconsistent use of terms or deprecated terms.

Bugs – balancing effort and value
Sometimes, text issues can be handled directly between teams. However, the most common way of handling issues is by using Jira ticketing. Tickets can be added as bugs or improvements, where bugs are related to something that is wrong according to the specification. Improvements can stem from UX designing something in a way that was not intended, but it still does not pose a huge problem. Bugs must be prioritised and solved before release, whereas improvements have a lower priority and should be implemented when time permits.

Issuing tickets related to text can take some effort and be time-consuming. The ticket must explain what is wrong, how it should be changed, and the context in which the problem exists. Furthermore, teams can be tagged to show it involves specific teams. However, some participants expressed an issue with not knowing who is in charge of what text, leading to teams un-tagging themselves. While tickets generally work well, many tickets can be issued. One participant stated that if everyone knew the context, they would not need to create a ticket.

> *"I don't think the solution is huge like each time you want us to change a small text, you have to go through a 10-step process to do something. It has happened times when it is really easy; just Slack someone and be like hey, can you change this into this text instead? And that actually works quite well, but in that case, it is because everyone already has the context, like I don't have to create a ticket."* (P3)

Adding or updating text rarely requires much effort. People generally take care of text tickets because they are quick and easy to solve. In many cases, people fix a text ticket without questioning why.

> *"Ah, I'm going to update a text. Then people feel good about themselves and finish it quickly. So, I think that the desire to fix things is high, because it's quite easy stuff and it gives instant reward, you can see that things are getting better. Everyone thinks it's nice."* (P2)

However, there are instances where perceptions of Jira tickets can differ and UX can sometimes be questioned. Which again relates to not having decisions related to text documented. There is no list of how to do things, leading to a very individualistic approach regarding value.

> *"But if it's something that is a little bit more effort, and we think the value is high like consistency, but other people may think like should we really open this ticket up to change a small word or capitalization. Things like that I think are very interesting because even there it's not even on a stream level, it can be very individualistic."* (P3)

Managing error messages
Error messages were brought up in almost every interview as a significant challenge and pain point. There is currently no good way of handling error messages and no place to capture and see them all. The UX team experience difficulty knowing ex-

actly what messages exist and how to trigger them. Error messages happen "further back", adding complexity from a UX perspective. Most error messages are currently stored directly in code, resulting in a lack of overview of what error messages exist.

Unawareness
From a UX perspective, unawareness of what messages exist was brought up as a significant pain point However, unawareness relates both to the UX team and other, more technical teams. Technical teams in this instance refer to teams further away from the end users. These teams have not been interviewed in this study. So, the insights are heavily derived from the UX team's perspective.

Some of the error messages are written by more technical teams, and oftentimes they are not aware that the messages they write will be shown to users in NMP. As a result, NMP users are getting messages that are on the first hand intended for internal use, and on the second hand, have not gone through UX. There are times when teams discuss error messages with UX. However, oftentimes they just write something because an error message is required.

> *"And there has been like, developers who made a first draft of all the messages and then came back to me and sat down and fixed each of the messages. But I don't think that stuff always happens either, sometimes it's like oh we just write something there. And then it's not even known that it is there."* (P3)

Because technical teams may not be aware that their messages will be shown in NMP, they seldom inform UX about the messages. As a result, the UX team does not become aware of these messages until a bug has been reported. While the UX team has written several of the error messages and aligned them with BE, there is currently no good way of handling them. Overall, the UX team has very little insight into what error messages have been created and implemented in NMP.

> *"I would guess that we have maybe a 50 per cent idea, if even that, of what error messages might occur."* (P1)

While unawareness is a significant issue, one participant expressed concerns about not having a way to see all the error messages. As the UX team does not have a pronounced ownership of error messages, it may feel impolite to constantly monitor other teams' work. However, the concern is more related to the UX team not being informed of error messages rather than them not having ownership of them.

> *"We are never really informed about them. So, we need get in touch then and say hello, what are all the error messages you have implemented now, could I see them and change the text on them? And that feels a little impolite too. So, we don't have a good process for dealing with that right now."* (P1)

Finding edge cases
Testing and finding edge cases was brought up as a pain point regarding error messages. It is not as simple as gathering all the error messages to solve the issue. Success messages were described as easier to see and monitor, as scenarios for success are clearer. In comparison, things can fail for many different reasons, and there are a lot of different edge cases. Testing error messages requires creating a scenario to see it. However, the UX team cannot test every single scenario, and often they are not aware of specific edge cases. It is also difficult for BE to find all the edge cases, especially when other teams or third parties are involved.

Contradicting requirements
UX and BE have different requirements in terms of error messages.

> *"We have slightly different views, or what to call it, slightly contradictory requirements between UX and back end."* (P7)

While teams are aware of the different requirements, adhering to the separate requirements was a challenge from both UX and BE perspectives.

BE must be able to troubleshoot issues; thus, they need access to an error log with enough information to be able to troubleshoot. UX must create messages aimed at the users. So, if UX only creates error messages, BE cannot troubleshoot and solve issues. If BE only writes error messages, users will get complicated messages intended for internal use, which will cause confusion. As a solution, BE parse error messages to implement messages intended for users and logs intended for troubleshooting. However, it is not the case with every error message.

Preparing for translation
While translation is not currently implemented in NMP, it is a high priority for FE. Thus, it is an aspect that must be considered now rather than in the future. FE has already started preparing for translation by incorporating translation keys in their files. One participant highlighted the importance of having a better text structure and a single source of truth. However, they also emphasised the need to prepare for translation and facilitate its implementation in the future.

> *"We mustn't drop the ball, too, that translation will be a big part of it"* (P8)

However, translation follows several challenges. The translation aspect mostly affects FE, so most insights were drawn from interviews with FE team members. Challenges were related to current file structures and implementation.

Lack of structure
The structure was discussed as a problem with general text management processes and future translation. Not keeping track of all text or having an overview of all the files will make introducing NMP to other markets challenging.

> *"We lack an imminent strategy for the structure of our files and how we manage our files [...]. If we don't have everything under control, it will be really difficult to introduce NMP to other markets."* (P8)

Another issue brought up related to the structure was that translation will most likely be done externally, and if the text is being handed over externally, the structure must be understood.

> *"I don't want to sit and hold their hand; I don't want to spend time on that. I do not have time."* (P8)

Implementation
Implementation was also brought up as a crucial challenge FE is currently facing. They need to solve this problem before translating NMP. The issues were regarding handling Right-to-Left languages and languages that do not use the Latin alphabet. For example, how to handle UI components dynamically for Left-to-Right and Right-to-Left languages.

6.2 Empathy maps

It became apparent during the interviews that UX and FE have the most involvement in text management for NMP, and their processes would be most affected by a text management tool. Thus, the empathy maps (Figure 6.1 and 6.2) illustrate these teams, conceptualising findings from the interviews. Empathy maps can be adapted when needing more details or to express unique needs [51]. While the traditional empathy map consists of what the user says, does, thinks, and feels, this was adapted to tasks, influences, pain points, feelings, and goals to better align with the context of product development and visualise the insights from the interviews.

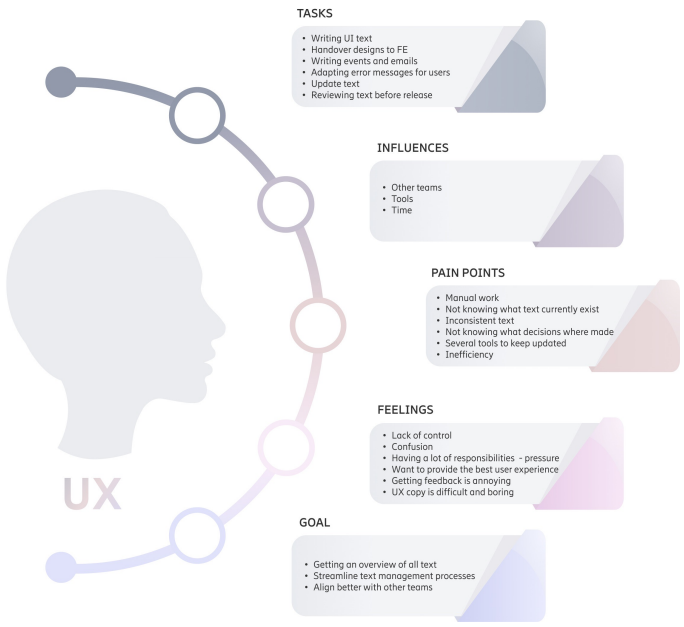


Figure 6.1: Empathy map UX.

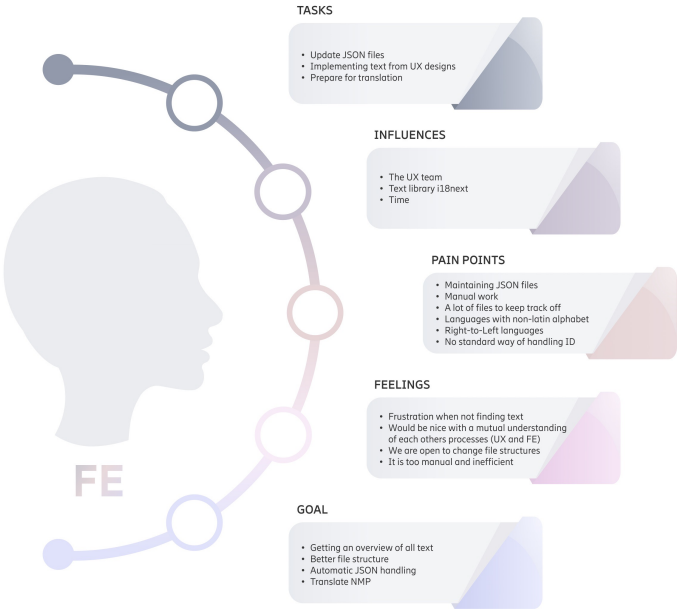


Figure 6.2: Empathy map FE.

6.3 Competitor analysis

The competitor analysis was based on the problems and goals identified in the interviews and aimed to gain market insights. Instead of comparing an existing product against similar products, the competitor analysis focused on understanding what tools exist and how they address Ericsson's needs and pain points discovered in the interviews. While this thesis addresses a specific use case, the problem areas are not limited to Ericsson. Thus, insights can be drawn from how problems have been previously addressed. The purpose was to investigate strengths and weaknesses in existing tools to draw inspiration and understand what current tools are lacking. Furthermore, the aim was to investigate whether an available tool can solve the challenges. The findings are one of the multiple sources that informed the requirements.

6.3.1 Software

A market search allowed for finding relevant tools for analysis. It became apparent that there was a lack of direct competitors as no tools were solely focused on text management or related to the problem scope. Thus, the analysis consisted of indirect competitors. The tools were selected based on popularity and relevance. The selected tools cover functionality beyond managing text but, regardless, including the ability to do so. While no apparent tool directly addresses the requirements regarding text management, existing tools can still provide valuable insights. Thus, the analysis included Localisation platforms, Content Management Systems, and Digital Assets Management.

Localisation platforms facilitate organisations in translating their products and connecting text items to specific IDs, which are used to store translations. The goal is to enable companies to enter new markets by allowing users to use the solution in the desired language [62].

CMS are software for managing web content, allowing users to create, edit and publish content. The main goal is to manage content intuitively and simplify the web development process [63].

Digital Assets Management (DAM) platforms are used to organise, manage, and share companies' digital assets. Digital assets are digital entities owned by a company, including logos, media, brand guidelines, etcetera [64]. While the purpose of DAM differs from the other systems, insights can still be drawn from how the software is built and functions. Thus, one DAM was included in the analysis.

Six different tools were analysed; see table 6.3. The competitor analysis was conducted between 6 March 2023 and 13 March 2023. Thus, data is limited to the version available at the point of analysis.

Table 6.3: An overview of the software included in the competitor analysis.

Name	Type
Lokalise.com	Localisation platform
Transifex.com	Localisation platform
Crowdin.com	Localisation platform
Contentful.com	CMS
Kontent.ai	CMS
Bynder.com	DAM

6.3.2 Procedure

The analysis was based on the framework presented in table 6.4. Each topic is related to one or more themes and sub-themes derived from the interviews. The framework acted as a guide to focus analysis and draw insights based on current needs.

Each tool was analysed by a walkthrough, focusing on each topic within the framework. Moreover, any additional thoughts, functionalities, or other insights were noted. Once all tools had been reviewed, insights were gathered on sticky notes and organised in FigJam based on the framework. The analysis followed deductive reasoning, as the competitor analysis aimed to relate current tools to insights gathered from the interviews. The insights were then divided into strengths and weaknesses. Strengths involved general positive insights and areas where tools aligned with Ericsson's needs. Thus, these insights could be used as inspiration towards the design

process. Weaknesses involved areas where tools did not meet Ericsson's needs or identified issues with how functionalities were implemented. All insights grouped as strengths and weaknesses were based on Ericsson's needs and pain points.

Table 6.4: Competitor analysis framework.

Topic	Focus areas	Sub-theme
Integrations	• JSON integration • Figma integration • Integration functionality	Manual text management Perception of current tools
Structure	• Tool structure • Creating hierarchies • Designer developer balance	Single source of truth Different perspectives Lack of structure
Search and replace	• Search functionality • Searching across files • Find and replace	Perception of current tools Manual text management Consistency
Guides and documentation	• Upload guides • Upload documents • Integrations	Guidelines for text The importance of documentation
Functionality Collaboration	• Fit for purpose • Task management • Tracking updates	Single source of truth Manual text management Unawareness
Context	• Level of context • Visual context	Perception of current tools Bugs: balancing effort and value
Access and roles	• Assign roles • Adaptability	Responsibility Prospects of the future

6.3.3 Findings

The findings are structured according to the analysis framework. Strengths and weaknesses are presented within each category. Figure 6.3 provides an overview of all analysed tools concerning the framework. The tools are coded according to implementation and alignment with Ericsson's needs (green), issues with how it is implemented or not aligning with Ericsson's needs (yellow), and not available (red). An apparent issue with available tools is that they are unfit for purpose. While the

tools offer several useful functionalities, they are lacking in other areas. For example, CMSs concentrate on structuring overall content for a website, while localisation tools concentrate on translation. The latter proved most beneficial concerning this project's problem scope as it relies heavily on text strings. However, currently, other focus areas besides translation must be addressed foremost. Thus, none of the analysed tools can be implemented directly in Ericsson's current processes.

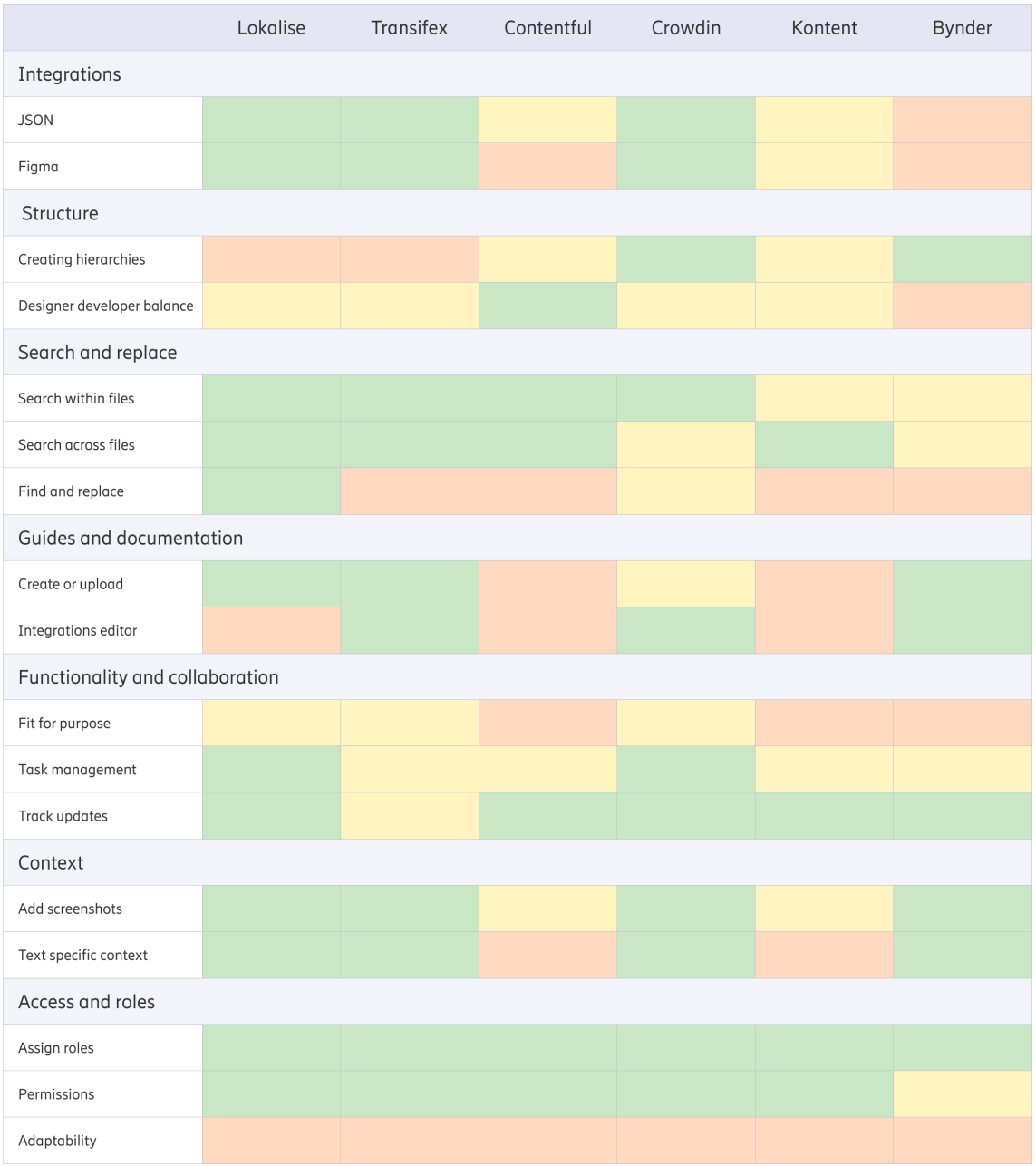


Figure 6.3: Competitor analysis summary.

Integrations

Some teams have specific software and ways of working regarding text management. For example, UX uses Figma to design and communicate wireframes. FE stores all

text for NMP in JSON files. These processes will remain even though a new tool may be introduced. Thus, the analysis focused explicitly on how tools integrate Figma and JSON.

Most tools offered some level of integration with external software. Bynder was the only tool without any integrations with external software or applications. Furthermore, while Contentful offered integrations, they were focused on communication and project planning rather than integrating designer or developer tools. For example, they provided integrations with Slack and Jira but not Figma or similar tools. Content could not be exported to JSON files, and while uploading assets supported JSON files, they could not be integrated into the workspace.

Lokalise, Kontent, Crowdin, and Transifex offer Figma integrations. Lokalise, Transifex and Crowdin were similar in their Figma integrations. In summary, it involves pushing keys and base language from Figma to the in-tool editor, translating in the editor and pulling the translations back to Figma. While it is an intuitive integration with Figma, it is structured so that designers set the keys in Figma files, and developers access them after. Nevertheless, the integration automates connecting keys to screenshots and directly imports the text from Figma frames to the editor.

Kontent did not offer a Figma integration as a specific plugin related to the tool. However, Figma contains plugins allowing data synchronisation with external tools. Thus, Kontent refers to this function when it addresses Figma integration. To connect Kontent with Figma, users must first create an item, a page, for example, within a component in the content model. If the page does not exist in Figma, the UI must first be designed in Figma. Then, the item can be connected from Kontent to the Figma prototype by using data synchronisation through an API generated in the Kontent item. The integration mainly ensures the same text is used in the Figma and the Kontent files and simplifies copy-paste, allowing direct "insert" of text to Figma from the content item in Kontent.

Transifex and Lokalise offered the best JSON integrations. As Transifex and Lokalise are focused on translation, they are automatically heavily focused on text strings. Thus, JSON integration is a comprehensive part of the systems. JSON files could both be downloaded from the editor and uploaded and integrated into the editor. Furthermore, Transifex allows exporting a file limited to reviewed translations, and Lokalise allows exporting bundled to language files or based on file names. Nevertheless, Transifex was highly focused on source files. While the system senses new text strings from the updated source file, it cannot add them directly in the editor. Instead, source files must be updated and uploaded. Users can also set their source files to be automatically updated by providing a URL to the file. However, already existing text strings could be updated in the editor.

While the analysis focused on JSON and Figma integrations, Jira became another area of interest. Some tools offer Jira integrations; Lokalise is a good example, as it can connect Jira tickets directly to the context within the Lokalise editor. Providing context to Jira tickets was an area of discussion throughout the interviews.

Structure
Structure was a commonly voiced concern during the interviews. There needs to be more structure across text files for NMP, which ultimately impacts ways of working. Furthermore, structuring text files according to NMP structure is a requirement to make the text management process more intuitive and efficient. However, an evident issue throughout the competitor analysis was that most tools lacked flexibility in structure. The main issues identified were related to unclear structuring, leading to difficulty getting an overview of the content, inability to create folders, and inability to structure content freely.

The CMSs facilitated more flexibility in how content was grouped and stored and how files were structured. For example, Bynder had a good content hierarchy where assets could be stored within pages related to a website's site map. Teams can structure the files in a way that suits their needs. Contentful provided a structure based on how users create their project's content model. While the content model provides a good overview of the different components, the content they are filled with is scattered. Thus, there is no good way of getting an overview of the existing content.

Crowdin was the only tool focused on localisation that allowed users to create folders and structure files. It was flexible as users could create their own hierarchy with folders and sub-folders. Furthermore, users can view the content through created folders or individual text strings. In comparison, similar tools, Transifex and Lokalise, could not structure content into pages or folders. As a result, all text strings within a project are stored on the same page of the editor and displayed as one long list. While Lokalise had filtering according to tags or files, allowing users to filter what content to be displayed, there was no way of structuring files into separate views or folders. Transifex worked in a similar way; however, compared to Lokalise, the UI structure took more work to understand. The connection between translation keys and text strings was difficult to understand, and where certain text and updates should be written.

Crowdin further had a different structure than the other localisation tools. Instead of displaying all languages related to a translation key in the same view, languages are separated into different folders. While it lacks the ability to view all keys and related languages at once, it could be beneficial as you may not work on all translations at once. Nevertheless, Lokalise allows filtering the languages to view, and the filtering stays even when you log in and log out.

Search and replace
The ability to search and replace terms is essential to improve consistency across content. Finding everywhere a term was used was expressed during the interviews as a challenge. As a result, some terms are not updated before release leading to outdated or inconsistent information throughout NMP. All tools analysed offered the ability to search for specific content to some extent. However, few offered find and replace. Bynder had no way of searching content or utilising find and replace. The tool was primarily focused on uploading assets. Thus, the search function

allowed searching for asset names rather than content. While Transifex, Kontent, and Contentful can search for text strings across files, there was no find and replace function. Thus, updates must be done manually in each place.

Crowdin provided one of the best search functionalities of all the tools analysed. However, there were still issues uncovered. While terms can be searched and replaced throughout an entire project, the downside of Crowdin's find and replace was that it lacked context. When specifying what term to find and what to replace it with, users must select "preview" to see how many terms match the query. If the preview is not selected, strings will be updated regardless. However, while it provides context to where the term exists within a full-text string, it provides no context to where the term exists throughout the project. As Crowdin allows creating files and folders, users can search among files under the "files" tab or through individual text strings under the "strings" tab. Searching within strings will include text from all different files related to the language folder the user is currently in. However, the search does not encompass all languages and only works within the active language file.

Lokalise provided more context in the find and replace function, showing the keys related to the text string and providing possibilities to search and replace within tags or specific keys. Compared to Crowdin, Lokalise find and replace function is not case sensitive or requires exact matches. If a term begins with a capital letter or includes an exclamation point, it must be included in Crowdin's search function to render the results. On the other hand, users can select "match case" in the Lokalise search function. Nevertheless, it is not the default. Furthermore, Lokalise allows searching through all languages at once. A concern brought up during the interviews was that Figma's find and replace function only includes the current file and does not impact all files throughout a project. As all text strings are stored on the same page in the Lokalise editor, it cannot be verified if the same problem would exist in Lokalise.

Guides and documentation

The interviews showed that guidelines exist on both organisational and team levels and address different perspectives. Thus, at this point, guidelines are stored in different places in different formats. As a result, some guidelines are not accessible to specific teams. In other situations, a lack of guidelines was expressed, sometimes resulting in incorrect formatting and terms. Most tools analysed allow for adding style guides or glossaries. Glossaries were primarily offered through localisation platforms, in which glossaries are added by language. Transifex and Crowdin provide relevant suggestions from the glossaries directly to the editor when editing. Lokalise does not support such functionality. A glossary functionality is highly relevant, as using the correct terms within NMP is crucial for understandability.

Bynder and Transifex include style guides. However, they can be utilised differently. For example, in Bynder, it is possible to upload existing guidelines and templates. In Transifex, users must write them manually in the provided editor. As Ericsson already has a lot of different style guides for different purposes, it would be beneficial to upload them directly instead of rewriting them manually in a potential tool.

Kontent and Contentful are CMSs and are both based on content models built up through different elements. Thus, they work as guidelines for creating individual items based on these templates. However, these are content-based guidelines; they do not provide any additional functionality, such as a glossary or other text guidelines.

Functionality and collaboration
Currently, teams at Ericsson manage text for the NMP differently, and multiple tools are used to update and share text. Thus, this part of the analysis focuses on what functionality is provided in the selected tools and in what ways they support collaboration.

As different types of systems have been analysed, they serve different purposes and, thus, provide different functionality. While Transifex, Lokalise, and Crowdin are tools for translation and localisation, Bynder, Kontent, and Contentful are systems for managing content or assets. The latter is managed by building content models, of which individual content items can be created. Both Kontent and Contentful depend on direct and manual management within the provided editor. This means that if content and text are to be added, it must be done through the editor. Thus, these systems might work better if starting from scratch. However, as Ericsson already has a lot of text in action, it would imply extensive manual work.

Bynder focuses on uploading assets, visual design, and ensuring that people follow brand guidelines. Bynder mainly addresses media, graphics and documents rather than individual text strings. This is problematic in terms of Ericsson as they, in many ways, must consider content as individual text strings, to align error messages across teams, ID handling, translation, and many other things.

The localisation tools focus not on the content but on text strings. The users are provided with an editor where text strings from the original language can be translated into a number of desired languages. As previously mentioned, Ericsson needs to consider individual text strings, which makes the functionality of localisation tools a better fit for the purpose, even though the translation is not on the radar yet. As these tools focus heavily on text strings, they provide functionality beneficial when working with text.

As all tools enable collaboration, functionality is provided to support collaborative work. This includes adding tasks or issues, comments, and reviewing or approving updates. Most tools provide system feedback connected to these functions, which is important for Ericsson to enable teams to follow updates. The majority of tools also provide editing history, making it possible to track the evolution of updates. Further, a comment functionality is a possibility to follow decisions, as they are connected to specific content or text strings. Some of the tools also provide collaborative work through announcements, discussion forums, and messages. Kontent offers “content type snippets”; reusable groups of content elements, making modifying elements from one place possible.

Context
This part of the analysis focuses on contextualising content or text strings. The context issue was discovered in the interviews, as some participants struggled with understanding where text will be present in the system, including what different text states are available. All localisation tools provide a function called screenshots. In Lokalise, Transifex, and Crowdin, screenshots of an application can be added to provide context to text. The user can connect specific areas of the screenshot to a text string or translation key, enabling context during translation of exactly where the string will be shown on the screen. If a screenshot is added, it will be visible in editor mode. Bynder also allows context and information to be added for each asset. They can be connected to personas, goals, screenshots, etcetera.

While Contentful and Kontent do not have functionality for uploading screenshots, Kontent can be combined with an additional plugin called Web Spotlight. With Web Spotlight, it is possible to update content directly in a preview website mode. In addition, content can further be arranged with a tree structure, which allows additional structuring. Other contextualising possibilities found were connected to specific text strings or components. For example, when editing a text string in Crowdin, the user is provided with the translation key and to which file it belongs. Furthermore, the user can add a description to the text string. If uninterested in such context, the user can hide it, which will be set as default.

Access and roles
Teams at Ericsson have different areas of focus, and thus, the requirements for managing text differ. For example, FE must assign ID and translation keys to text strings, factors that do not have to be considered by UX. Thus, all text must be accessible to all teams. However, in some cases, certain content may only be necessary for certain teams.

All tools provide the possibility to assign different roles to users. Commonly, these are assigned on an organisational level or team level. The primary difference between the restriction possibilities is based on purpose. For example, CMSs systems enable assigning users to specific content types, whereas localisation tools allow assigning specific languages. All tools further allow restricting access and permissions to different roles so that only permitted users can modify the content. It is difficult to conclude whether restricted access adapts the interface and "hide" restricted actions.

6.4 Design workshop

The problem space was well understood due to the studies from the empathy phase. However, users were kept involved in subsequent stages as well. Continuing the DT process, a design workshop initiated the ideation phase. The main purpose was to foster rapid ideation and prioritise ideas based on different perspectives, including UX and FE, as stakeholder involvement facilitates broadening perspectives and reduces designer bias [47]. Thus, the goal was to include external perspectives to inform decisions, design direction, and iterate the requirements.

6.4.1 Participants

Five people participated in the session, including two participants from FE and three from UX. For relevancy, the workshop was limited to FE and UX. It became apparent during the interviews that UX and FE are most involved in text management for NMP, and a text management tool would most affect their processes.

6.4.2 Procedure

The workshop was held physically at the Ericsson office, with one participant attending online through Teams. The session lasted approximately two hours, including a break, and was a combination of digital and physical ideation. Before the session, a FigJam file was prepared and shared with the participants. FigJam is an online collaborative whiteboard for teams to ideate and brainstorm [65]. It was the best option as it is easier to manage, organise, and group digital sticky notes. Furthermore, it has an integrated timer and voting system. However, to facilitate rapid ideation, the sketching was physical. The workshop had no designated moderator throughout. Instead, the moderator role was shared, responsible for different parts of the workshop. The person not moderating oversaw the timer and took notes. The workshop was divided into three different parts. First, it focused on generating ideas on sticky notes, organising and voting. Then, it focused on rapid sketching based on ideas from the first exercise. Finally, participants presented and discussed their ideas.

Before beginning, each participant signed consent forms and was briefly introduced to the thesis and scope. Furthermore, the main findings from the interviews were presented to provide context. The first part of the session utilised an altered KJ technique. Participants had 10 minutes to individually generate as many sticky notes as possible related to the question *How might we streamline text management processes?* For reference, the main question was supported by ideation areas of functionalities, ways of working, and collaboration. The areas acted as a guide for participants and to focus ideation around the areas most relevant to the research questions. However, participants were allowed to ideate freely if other areas came to mind. The main instruction participants received was to think of solutions and ideas rather than listing problems.

After the ten minutes were up, participants had another ten minutes to organise the sticky notes into groups and name each group. Generally, KJ advocates for grouping sticky notes in silence; however, participants were allowed to speak during this workshop. Interestingly, participants did not speak much during the grouping, yet everyone stated that they were happy with the grouping after the time was up. In addition, participants discussed more during the naming of groups. The session resulted in around 50 sticky notes in which ten themes emerged: *usability, consolidate tools, one place for all text types, automation and integration, functionalities, translation and languages, ownership, author and transparency, process*, and *presentation.*

Once all sticky notes belonged to a named group, participants voted on the groups they considered most important in terms of streamlining the text management process. Participants had five votes each to distribute and were only allowed to vote once per group. Allowing participants to vote provided an understanding of how users prioritise requirements and established direction for the workshop's sketching phase. An advantage of using FigJam is its integrated voting system. Once the voting starts, each participant will have a previously determined number of votes. Then, after ending the vote, FigJam renders an ordered list of the items depending on votes. Thus, the four top groups were apparent directly upon ending the vote. Furthermore, participants cannot see each other's votes while the voting is active, removing potential biases in responses.

Two groups received five votes: one place for all text types and functionalities, and two groups received three: author and transparency, and process. Thus, the four groups with the most votes determined the direction for the sketching. Participants were only available for a limited amount of time. Furthermore, the first part of the workshop allowed narrowing down ideas and areas. Thus, the sketching was an altered version of Crazy 8 to explore ideas within as many areas as possible. Participants had 4 minutes, 1 minute per idea, to ideate on the two highest voted groups, and 2 minutes, 1 minute per idea, for the two second highest voted groups. Altogether, participants produced 12 sketches each.

The session ended with a 30 min discussion, where participants presented and discussed their ideas and thoughts related to the prioritised groups. The discussion was recorded through Teams, including both video and audio. The recording enabled transcription and provided context as to which participant was speaking and what sketch they were referring to. After the session, the sketches were collected and numbered according to the four groups.

6.4.3 Analysis

The analysis included multiple data sources: sticky notes, sketches, discussions, and notes taken during the workshops. As affinity diagramming is suitable in multiple scenarios for collaboratively organising research findings or design ideas [50], it was used as an analysis technique to include multiple data sources.

The recording of the discussion was transcribed similarly to the interviews. The approach utilised intelligent verbatim to improve readability and understandability by removing filler words [60]. Once the discussion had been transcribed, insights and ideas were highlighted from all data sources. Thus, the themes grouped beforehand by the participants were omitted so that all critical insights and patterns were noticed. Utilising an inductive approach enables the data to speak for itself [61]. As insights were highlighted from all data sources, they were clustered into themes. This was done iteratively by reviewing and restructuring sticky notes to discuss multiple possibilities. Once themes were reviewed and finalised, the affinity diagram was refined. See Appendix D for the final affinity diagram.

6.4.4 Findings

Six themes emerged from the data analysis: Transparency, Context, Structure, Non-technical, Translation, and Automation; see table 6.5 for a description of each theme, followed by a detailed presentation of each theme and the insights that were drawn.

Table 6.5: Themes emerged from workshop analysis.

Theme	Description
Transparency	Follow edit history of text strings and what actors have been involved.
Context	Give context to how text relates to the system, the code, and other text strings.
Structure	Anchor structure with current system or structure based on text types.
Non-technical	Easy to use and no programming skills required.
Translation	Support different languages.
Automation	Decrease maintenance through increased automation.

Transparency
It became apparent that transparency is vital for all participants. Aligning perceptions and promoting a shared understanding is essential. Thus, everyone must have access to the tool and be able to understand it. Ideas related to transparency were closely connected to improving collaboration. For example, participants suggested assigning responsibilities to know who owns a text from an organisational perspective and who maintains specific text. Furthermore, users should be able to see contributors when a text has been edited. Regarding transparency and context, participants suggested a comment function when editing a text to give context to why a text was changed.

One participant had the idea of tagging reviewers to a text. It received support from the UX team as it is a way of allowing UX to impact text despite a more technical stream having pronounced ownership. Participants highlighted that the tool should include editing history for transparency and to streamline the review process. Users should be able to see when text has been updated and access all the text that has been updated within a specific period. Editing history should also include decision-making to facilitate cross-functional collaboration. FE highlighted that it is imperative to see when someone has updated an ID as it greatly impacts files.

Participants expressed that the tool should include possibilities to specify owners and assign permissions. For example, some people can only read, while others can suggest and edit. While concerns regarding who decides access and roles were discussed, the main takeaway was that the tool should include the functionality.

Context
Ideas emerged connected to contextualising the relationship between different text strings or between text strings and NMP. For example, the former was conceptualised through an idea where it is possible to link different text strings to one another. This could help users to understand if an event triggered an email and similar scenarios. Another idea was to present this through a node view to get an overview of the different NMP parts it affects. Additionally, the context was brought up regarding error messages to enable users to understand why particular messages are triggered. Such context was predominantly expressed from a UX perspective, as they need to translate errors into user-friendly text for the user to understand; what has happened, why it happened, and how to approach the problem.

Other ideas included the importance of visualising metadata and how text is connected to certain UI elements or code; for example, connecting validation messages to input fields. Somewhat related to this, discussions arose about live preview functionality to enable previewing text in their intended context. In addition, existing CMSs were discussed as a possible solution as they facilitate managing content and building websites, including previews. While all participants agreed on the importance of different kinds of contexts, one participant expressed concern about who should oversee adding this to a potential solution.

Structure
The importance of aligning a future solution's structure with the current system's (NMP) structure became apparent. To anchor this with the existing structure, participants sketched solutions with a construction based on NMP pages or components. Some ideas included a construction based on text types rather than NMP pages, i.e., a division based on UI text, snack messages, event logs, and emails. A well-established information architecture is needed to support this. One participant specifically addressed the importance of categorisation systems.

Another recurring subject was to include functionality for a release or time-based version management. The benefits of version management referred to the possibilities of tracking editing history based on versions, freezing deployed versions, and facilitating release-based text reviews. Furthermore, FE had ideas related to variable structure, which is important for their translation library for future translations.

Non-technical
Several participants highlighted the importance of catering to non-technical users. For example, as the current text is stored in JSON files or code, a potential tool could act as a bridge between developers or other technical teams and other less technical teams. Furthermore, the solution must be easy to use and not require programming experience. Similarly, error messages must be supported with their user-friendly counterparts, including potential descriptions of why an error occurred. While NMP mainly consists of static text, dynamic text must be present in some situations. Thus, ideas emerged that a solution must support the usage of variable injection. However, FE expressed that it is only necessary with an indication rather than any technical execution.

Translation
Ideas exploring future translation were mainly addressed by FE. Ideas referred to supporting different language variations and UI preview functionalities, enabling text editors to understand how different languages will look within the system. Furthermore, one participant expressed the importance of context for future translations. It is essential, especially if translations will be outsourced, as a third party must understand what a text is expressing and in what situations to localise and translate correctly.

Automation
Several ideas and insights were related to automation. Participants expressed a general desire for more automation alongside specific areas where automation can improve the process—for example, having integrated spellcheck in the system. Several issues regarding text are related to being made aware of who writes text and what is written. However, integrated spellcheck prevents unnecessary extra errors such as misspellings.

Furthermore, search and replace was a topic of discussion, and something UX requires, as Figma only allows search and replace in individual files. However, while it is necessary for the UX team, it was highlighted as a potential risk during the discussion. Thus, participants from FE and UX had different perceptions of the importance.

Participants discussed the importance of low maintenance and specifically ideated around reusability. The tool should store text that is used in several areas and contexts. For example, if the same validation message is used in several inputs, that should be accessed and reused instead of writing a new string for each input. Similarly, standard messages that appear for several triggers should be reused. Reusability was highlighted as important for low maintenance and to improve consistency.

CMS was discussed in relation to automation, where one participant had previous experience working with CMS. However, while CMS can automate several parts of the process, three significant concerns were discussed. First, a CMS would require an API to access data, which raises concerns regarding load time. An API would further complicate the process as FE store text locally in JSON files, and it is a requirement that they continue to do so. Thus, the idea was to support exporting text files from the tool. The idea relates to low file maintenance, as FE does not want to maintain their JSON files. Second, the solution is not intended to push content to NMP directly. Instead, the main goal is to improve the process of managing text, promoting a shared understanding between teams and a better overview of the text. Lastly, CMS security was brought up as a concern. Moreover, participants were unaware of any CMS specifically built for managing text.

6.5 Defining requirements

Requirements were defined to determine the expectations on functionality and performance. The requirements result from all the data gathered, including data from the interviews, competitor analysis, and workshop. The requirements were framed based on the aim of streamlining text management processes and synchronising teams to improve cross-functional collaboration.

6.5.1 Procedure

A first draft of the requirements was drawn after the interviews. At this point, the requirements addressed high-level problems in current text management processes and desires for the future. However, as requirements can be informed through different methods and should evolve throughout the process [39], the first draft was iterated after gaining more knowledge through the competitor analysis and design workshop. Therefore, all requirements were kept in a document that was continuously iterated.

Researching and observing existing systems or products can be a source to inform requirements [43] [39]. Accordingly, the competitor analysis provided inspiration based on how existing software solved some of the identified problems. While the design workshop focused on generating ideas from different perspectives, it became apparent that many of the defined requirements could be validated. In some cases, new requirements were defined as participants from different teams participating, enabling rich discussions that were not covered earlier in the process.

Simultaneously, requirements were prioritised according to the MoSCoW model: Must have, Should have, Could have, and Would like to have. However, they were limited to the first three. The prioritisation reflected the insights gathered throughout the process and was based on the following parameters: how critical the problem is that it aims to address and how crucial it is for a successful implementation. Hence, the requirements addressing critical problems that are crucial for implementation were defined as Must have. Accordingly, if the requirements were defined by insights derived from desires that are not crucial, they were defined as Should have or Could have. The prioritisation was iterated as the requirements evolved.

Requirements are ideally coupled with success criteria, making it apparent when a requirement is fulfilled [39]. As this thesis presents a semi-functional prototype, no quantitative measures are set to evaluate the success of the requirements.

6.5.2 Requirements list

Figure 6.4 and 6.5 present the final iteration of the requirements list. Each requirement describes a more overarching expectation of the solution, supported by sub-requirements detailing specific measures required to meet the intended result. Requirements are prioritised according to the MoSCoW model.

Furthermore, each requirement is supported by a description, providing a rationale based on the identified challenges. Finally, the checkmarks indicate how the requirements were informed; through interviews (I), workshop (W), and competitor analysis (C).

Requirement	Description	I	W	C
1. The solution **must** align with current ways of working	The solution will not replace current processes. Instead, it will be a supplement to streamline text management and, thus, must align with established practices.	✓	✓	
1.1. The system structure **must** align with NMP structure	NMP text strings are connected to different pages and components. The system must reflect the same structure for users to understand the context in which the text exists. The system must also provide intuitive ways of defining and changing structure according to NMP changes.		✓	✓
1.2. The solution **must** enable version management	The solution must reflect Ericsson's way of working with releases and allow users to differentiate text strings between versions.		✓	
1.3. The solution **must** enable integrations with current tools	Ericsson utilises various development and design tools within the pipeline. Therefore, the solution must consider these tools to facilitate current ways of working.	✓	✓	
1.3.1. **Must** support JSON integration	NMP text strings are stored in JSON files, updated and maintained by FE. Integration with JSON allows FE to access the files from the tool instead of relying on copy and paste like current processes.	✓	✓	
1.3.2. **Could** support Figma integration	UI text strings are currently accessed from UX Figma files. Figma integration would allow UX to import text strings from their prototypes and provide previews within the tool when text changes.			✓
1.4. The solution **must** coincide with current communication tools	As Ericsson works cross-functionally, robust communication is essential for making decisions and aligning processes. Some tools are used extensively for communication and documentation, regardless of team or project. Therefore, the solution should support rather than replace them.	✓	✓	✓
1.4.1. **Must** support Slack	Almost 40000 Slack messages are sent each month. Slack is the primary place for communication and sometimes decision-making. Thus, the solution must provide a way to access Slack threads to support users in following discussions and decisions.	✓		
1.4.2. **Must** support Jira	Ericsson utilises Jira to create tickets for bugs and improvements. All tickets should provide documentation of the decision made. Thus, the solution must provide a way to access Jira tickets with context to bugs and improvements.	✓		
2. The solution **must** enable managing text in one place	Ericsson has multiple tools and files for storing text, causing difficulty managing text throughout the releases. The solution must replace current systems to avoid managing text in multiple places.	✓	✓	
2.1. The solution **must** store all forms of text for the NMP	All types of text related to NMP must be accessible and managed, encompassing all teams. The solution should not only store UI text accessible in the prototypes but all text, such as error messages, events, and emails.	✓	✓	
2.2. Users **must** be able to get an overview of all text	The solution must improve awareness and allow users to get a clear overview of all text within NMP, as current ways of working result in difficulties getting a summary of the existing text.	✓	✓	
2.3. Users **must** be able to search and find specific items	NMP includes thousands of text strings, and storing them all within the same tool requires intuitive ways to search and find items of interest.	✓	✓	
2.3.1. **Must** support terms and text strings	In scenarios where users directly know what to change, they must be able to search specific terms and text strings.	✓	✓	
2.3.2. **Must** support text types	Currently, different text types are stored in different tools. Universally storing all text types requires the ability to differentiate between them.	✓	✓	
2.3.3. **Must** support ID and path	Each text string has an identifier and exists within a JSON path. Primarily FE utilises the path and ID to locate text strings. Searching specific IDs is also essential for future translation, where the same key is used for each language.	✓	✓	
2.3.4. **Should** support status	The solution should support the user with text progression to reduce the risk of releasing unfinished or incorrect text strings.			✓
2.4. The solution **must** streamline the review process	Before each release, text items must be reviewed. Thus, the solution must facilitate the review process, clarify what needs to be reviewed, and remove the need to review text in different tools.	✓	✓	✓
3. The solution **must** promote a shared understanding	Working cross-functionally requires continuously considering multiple perspectives. The solution must be a place for all teams to establish a joint vision of all text and a common ground to lean against.	✓	✓	
3.1. The solution **must** be accessible to all teams	Current tools and files for storing text are not always shared or accessible to everyone. Thus, the solution must be a place where all teams can access text and contribute with different perspectives.	✓	✓	
3.2. The solution **should** facilitate an understanding of ownership	Currently, there are misconceptions and uncertainty about ownership and responsibilities. The solution should mitigate the misconceptions and clarify who owns what text and who has responsibility.	✓	✓	
3.3. The solution **must** enable specifying roles and restricting access	Apart from clarifying responsibilities, users should be assigned roles concerning pre-defined ownership and permissions. Restricting and specifying what users can do within the system should be supported.		✓	
3.4. The solution **should** simplify sharing guidelines and style guides	Guidelines are currently spread out and seldom referred to. Thus, the solution should ensure that guidelines are more accessible and intuitive.	✓		✓

Figure 6.4: Requirements list part 1.

4. The solution **must** be transparent in collaboration	While UX is responsible for the text within NMP, several teams are involved in the text management process. Thus, the solution must ensure transparency in collaboration to mitigate problems related to unawareness.	✓	✓	✓
4.1. Users **must** be able to track updates	Several issues regarding unawareness and unreviewed text strings are being published to NMP, affecting product quality. Thus, the solution must allow users to see any changes made.		✓	
4.2. Users **must** be able to follow decisions	The entire text management process will not be handled within the solution. Discussions and decisions will continue to happen with external tools. Thus, users must be able to access the discussions that led up to an update or change.	✓	✓	
4.3. Users **must** be able to specify why a change was made	Apart from accessing external discussions, users must be able to provide context as to why a change was made and specify the underlying decision.		✓	
4.4. Users **must** be able to see contributors	Seeing what users made specific changes is essential for transparency. If necessary, it also allows users to know who to contact directly.		✓	
5. The solution **must** facilitate consistency in text	Consistency is crucial as NMP users are exposed to a vast amount of information related to their network. However, current ways of managing text, e.g., in multiple tools and files, make it difficult for UX to ensure consistency.	✓	✓	
5.1. Users **must** be able to update terms simultaneously	Current manual ways of working imply a high risk of missing places where text needs to be changed, resulting in inconsistencies. Thus, the solution must enable users to search and update terms in all or several places simultaneously.	✓	✓	
5.2. The solution **must** support reusability	There are several instances in NMP using the same or similar text. Thus, the solution should support reusability instead of users needing to write the same text in several places, reducing redundancy and improving consistency.		✓	
6. The solution **must** provide text context	Text on its own is abstract. Text strings are connected to attributes and context; page, placement, element, etcetera. Therefore, the solution must provide the user with relevant contextual information to ensure the correct text is managed.	✓	✓	
6.1. The solution **should** clarify what will be visible to the users	To mitigate unawareness of what will and will not be published to NMP, it should be specified what items are being displayed to users and what items are aimed for internal use.	✓		
6.2. The solution **could** provide visual representations	Visual representations of text can enhance contextual information, allowing users to see how the text displays in a UI element.	✓		
6.3. The solution **should** provide context to related texts	Text seldom exists independently on NMP; they can be related within a user task flow or different states within an element. Related texts should be accessible to provide context.	✓	✓	✓
6.4. The solution **should** display how texts are interconnected	Several texts are connected to a specific action and are dependent on each other. Therefore, it should be apparent when and what texts exist within an action and provide an intuitive way of editing and ensuring consistency.		✓	
7. The solution **should** prepare for translation	Ericsson Private 5G is expected to enter the global market, and some clients will require the product and portal to be translated into their native language. While it is not currently part of the process, the system should support the groundwork.	✓	✓	
7.1. The solution **should** support I18next integration	FE is currently working with the i18next framework to prepare for NMP localisation. The system should provide ways to align JSON files with i18next structures.	✓	✓	
7.2. The solution **should** enable exporting text for translation	Translations will likely be outsourced. Thus, all NMP text strings should be exported in a way where outsourcing is supported.	✓		✓
8. The solution **must** make the process of managing text more efficient	Manual ways of working with different tools cause many redundancies in managing text. Thus, the solution must make maintaining, updating, adding, and reviewing text more time efficient.	✓	✓	
8.1. Users **must** experience a low perceived workload	The solution should promote a manageable process. Users must be able to maintain text without added workload.	✓		
8.2. Text files **must** be low maintenance	FE maintains the text for NMP directly in JSON files. Currently, many JSON files must be maintained, and the number is increasing. Therefore, the maintenance of the files must be low.	✓		
8.3. The solution **must** enable non-technical teams to update text	All updates must be done in JSON files, implying that UX needs to go through FE for any changes. The solution must enable non-technical teams without access to the code to manage text.	✓	✓	
9. The solution **must** align with the development processes	Text strings will not be solely stored in the system; they must be integrated within current processes to synchronise teams and streamline the text management to release process. Thus, the solution must adhere to how NMP is developed and how it works technically.	✓	✓	
9.1. The solution **must** support storing text locally	Updates made within the system will not be directly published to NMP. The process of editing, reviewing, and releasing will remain. All text strings for NMP are stored locally in JSON files, and the solution must support keeping the current approach.	✓	✓	
9.2. The solution **must** format text according to JSON syntax	Objects and nested objects must be defined, and each text string must follow the JSON name/value pair structure. In addition, it must be clear what identifier is connected to a text string and its hierarchically structured relationship.		✓	
9.3. The solution **must** support the use of variables	Text strings can contain variables used to store specific data values, which are dynamically changed depending on, for example, users. Therefore, the solution must support the use of variables so they can be handled correctly.		✓	
9.3.1. **Must** support variable indication	The solution must support indicating variables when managing texts in the system, as non-technical teams generally write text. Furthermore, differentiating between variables and static change is required.	✓	✓	
9.3.2. **Must** support variable injection	The variable indications within text strings must be converted to the correct format to prevent issues with variables passed in the code.		✓	✓
9.4. The solution **must** support managing text for system errors	Currently, no way of storing and managing messages triggered by system errors exist, resulting in a lack of overview and comprehension of error messages. The solution must support such management to ensure system errors can be solved as efficiently as possible.	✓	✓	✓
9.4.1. **Should** support troubleshooting	Technical teams needs a detailed understanding of what has gone wrong in the system in order for them to solve the issue.	✓	✓	
9.4.2. **Must** support end users	While technical teams troubleshoot errors, UX must provide messages to ensure the user understands what they can do when an error occurs.	✓	✓	

Figure 6.5: Requirements list part 2.

6.6 Low-fidelity prototypes

Once the requirements were established, they were addressed through early prototyping. As the requirements were informed through all previous data collected, the early prototyping phase considered all findings accordingly. This section contains mind maps, sketching, user flows, wireframes, and evaluation.

6.6.1 Mind maps

It was imperative to structure the data and put it into context, as the early stages of the design process resulted in many insights and ideas. Thus, mind maps were created before any further sketching or prototyping. Mind maps are a straightforward form of cognitive maps and are valuable in design processes as they help organise and structure information [66]. Moreover, mind maps are specifically useful during ideation focusing on structure and overall concepts, as they facilitate planning information architecture, categories, and components.

Creating the mindmaps was a form of brainstorming to establish different views, components, and functionality. Furthermore, the mindmaps visualise different levels within features. Mindmaps facilitated the transition to sketching by first establishing the different aspects of the system. Creating mind maps also helped organise insights and promote a shared understanding of the design direction. Moreover, the mind maps guided the sketching phase to ensure the concepts included all necessary components based on the data collected. The goal was visualising the system without focusing on implementation details or correct labelling.

First, one large general mind map was created. However, as it became apparent that the different views would contain a large amount of information and features, they were separated per page within the application: dashboard, editor, repositories, and teams. Each mind map is presented in Figure 6.6, 6.7, 6.8, and 6.9. At this stage, the requirements had been refined through several iterations. Thus, the mind maps were based on the requirements list to ensure focus on user needs and prioritisation.

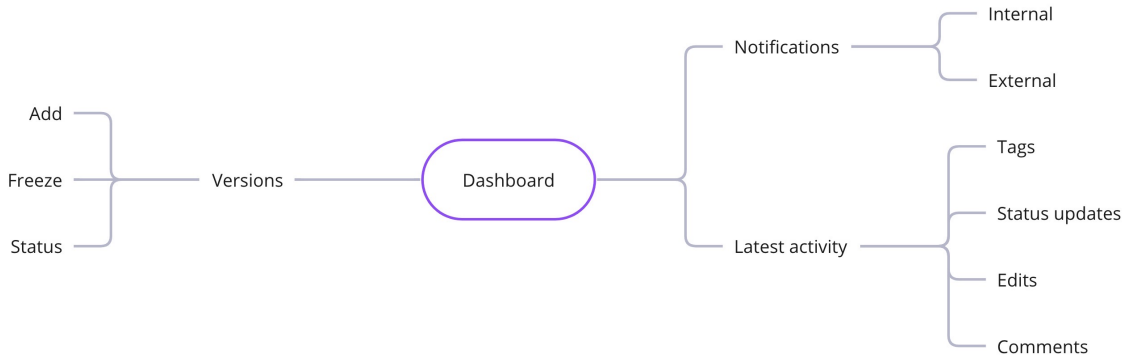


Figure 6.6: Dashboard mind map.

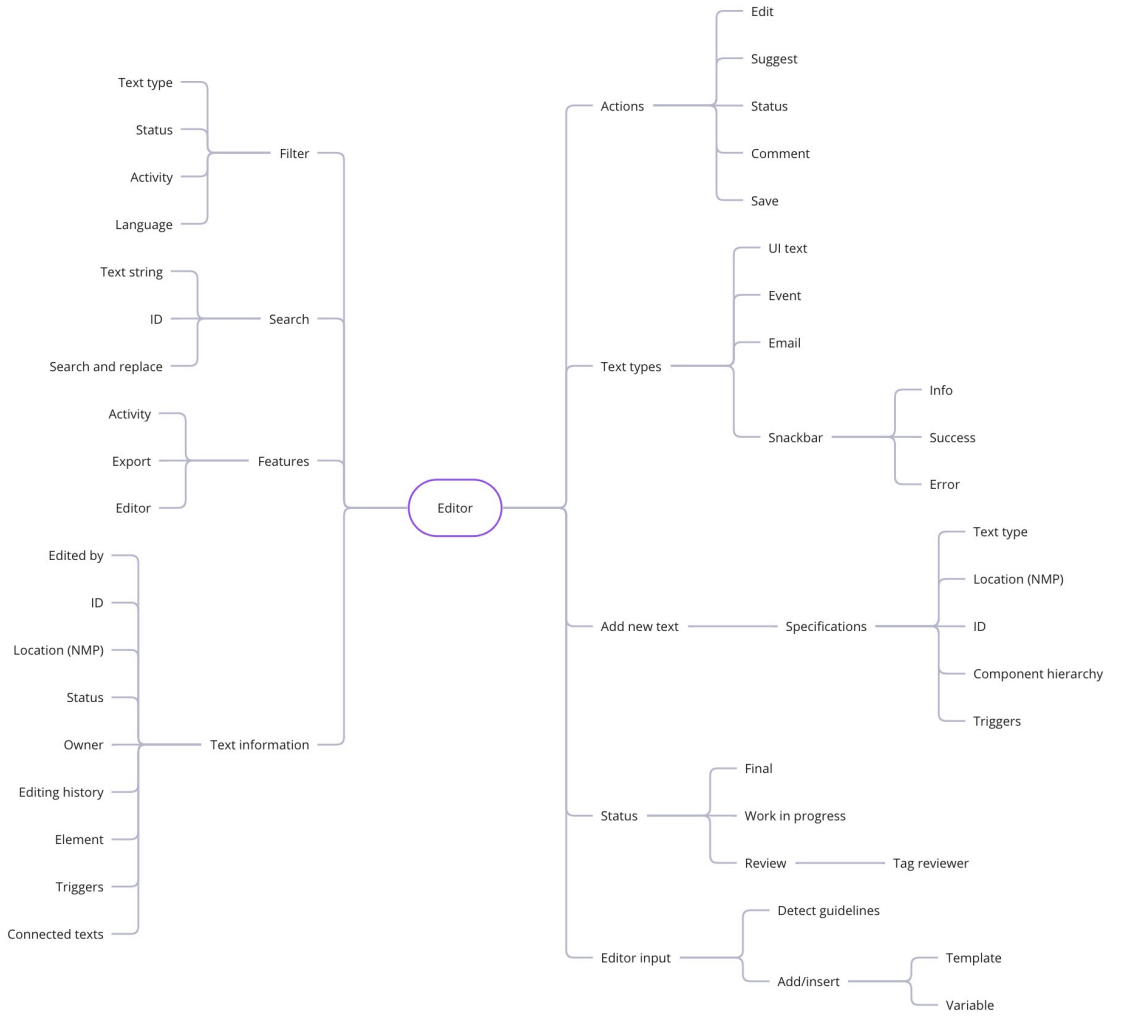


Figure 6.7: Editor mind map.

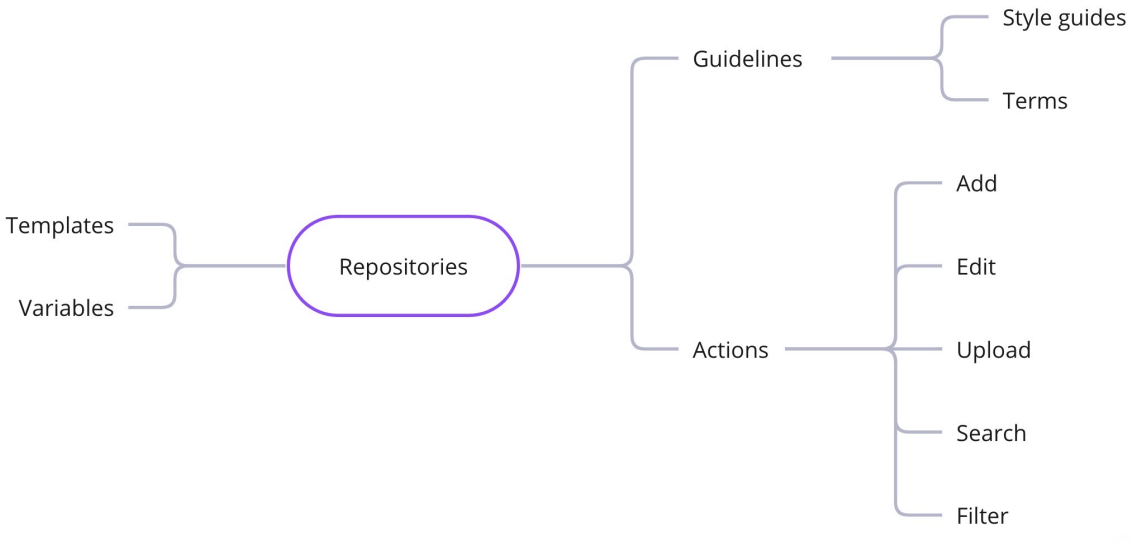


Figure 6.8: Repositories mind map.

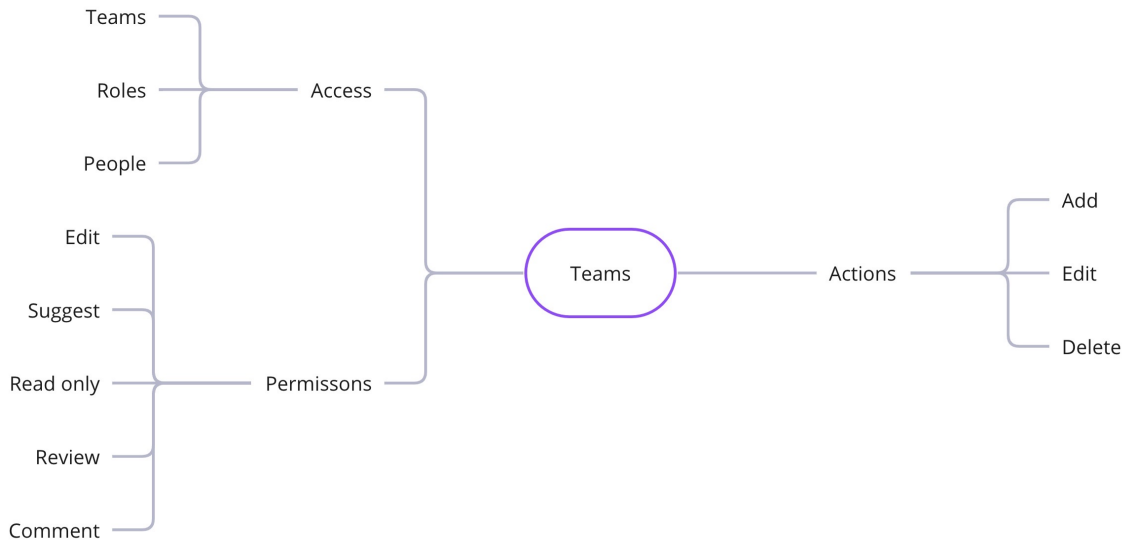


Figure 6.9: Teams mind map.

6.6.2 Sketching

Initial sketching focused on exploring several concepts within the different mind map topics and building on sketches produced during the design workshop. They were simple, focusing on overarching concepts and ideas rather than details. The initial sketching was solely done with pen and paper to encourage the rapid and imperfect mindset.

While the initial sketching session was a free space to explore all sorts of ideas, it was restricted to the topics within the mind map. Based on data and requirements, it was clear that the solution had to include the topics presented within the mind maps. However, the mind maps or requirements do not indicate how the implementation or design should be done. Furthermore, considering design principles such as learnability and perception was imperative in the early stages to ensure the sketches considered the necessary information volume.

Once sketches and ideas had been discussed within the design team, a few concepts were selected or combined. The selected ideas were then refined through digital solution sketching. See figure 6.10, 6.11, and 6.12 for a selection of the final sketches.

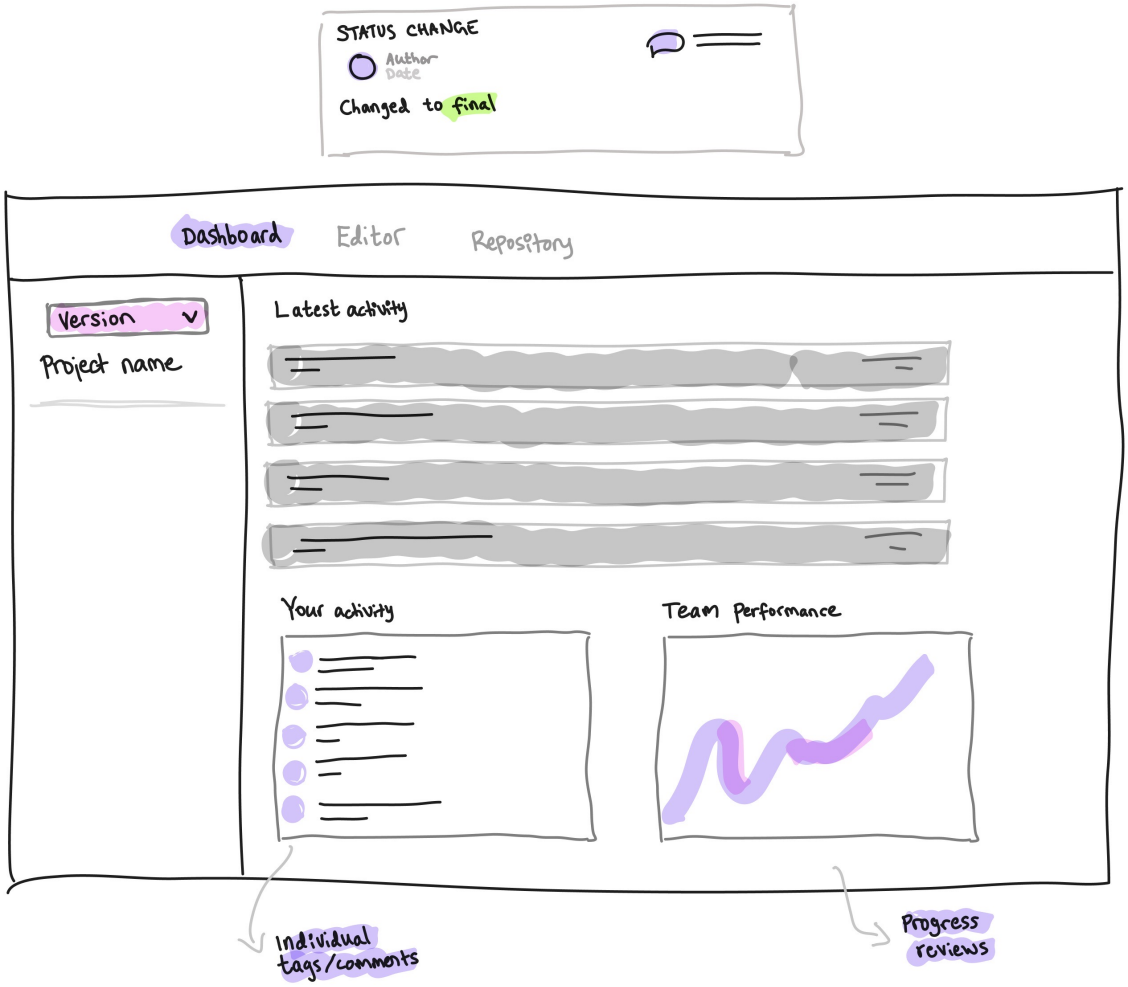


Figure 6.10: Dashboard sketch.

Figure 6.11: Editor side sheet sketches.

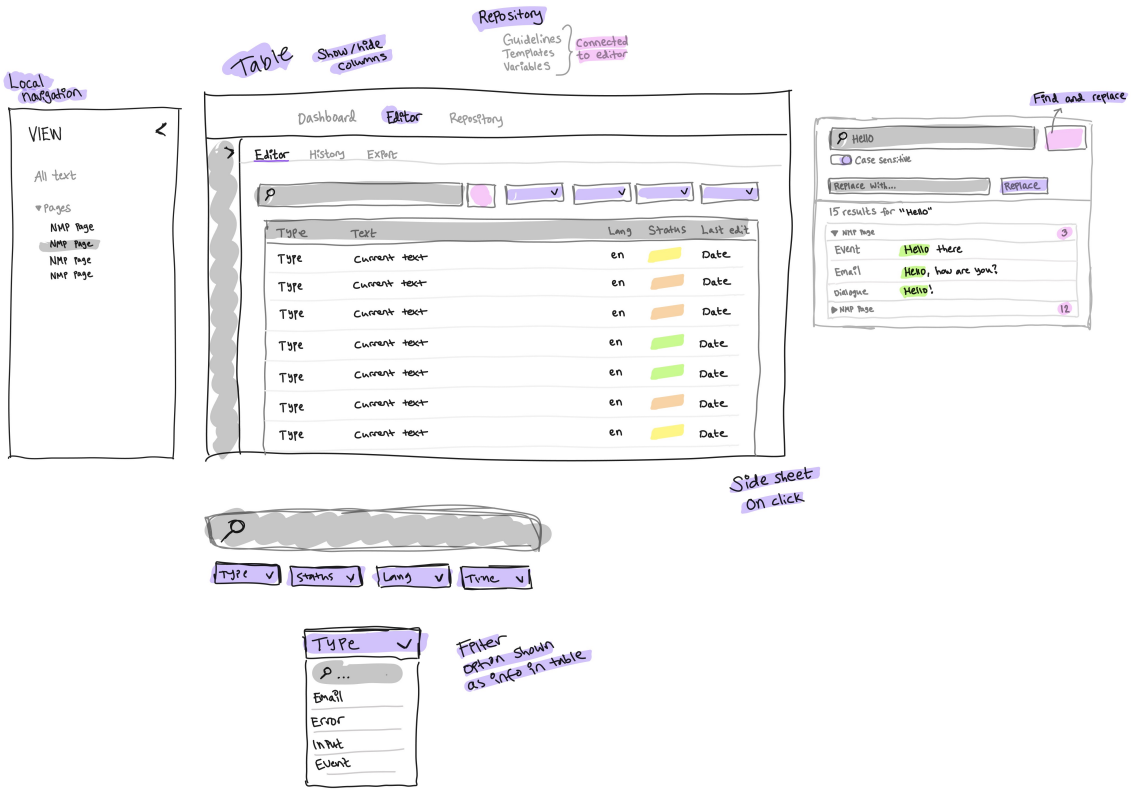


Figure 6.12: Editor sketches.

6.6.3 User flows

Interactions must be designed intuitively to ensure each feature is logically addressed and aligned with user needs. There are several things the system must consider, and ensuring users can access and utilise all features without getting overwhelmed or interrupted is fundamental. Thus, it was important to model how users may move through the system to reach their goals while considering different users and use cases before focusing on the design. It allowed identifying potential issues and preventing them from being uncovered once designing interactive prototypes. Furthermore, design principles should be considered throughout the entire design process, and memory and attention, accommodation, and ease of use depend highly on the flow of actions.

Creating user flows allowed optimising flow and having a solid foundation before creating wireframes. Three user flows were created, visualising the primary flows: locating a text, editing a text, and adding a new text. Each of the three flows needed close consideration as they constitute the main pain points within current processes. The final system closely resembles the flows created in this stage. However, some iterations were made, such as what options to filter on and status names. Furthermore, additional features connecting Slack and Jira were added in later stages. Tagging a user was only considered when assigning a review status in this stage.

Locating a text

Figure 6.13 visualises the flow of locating a text, which can be done in several different ways. First, users navigate to the editor page and select to view all text strings within the system or select specific pages in which the text string exists. Further refinements can be done by filtering on text type, language, and status. Users can also use text search to locate text.

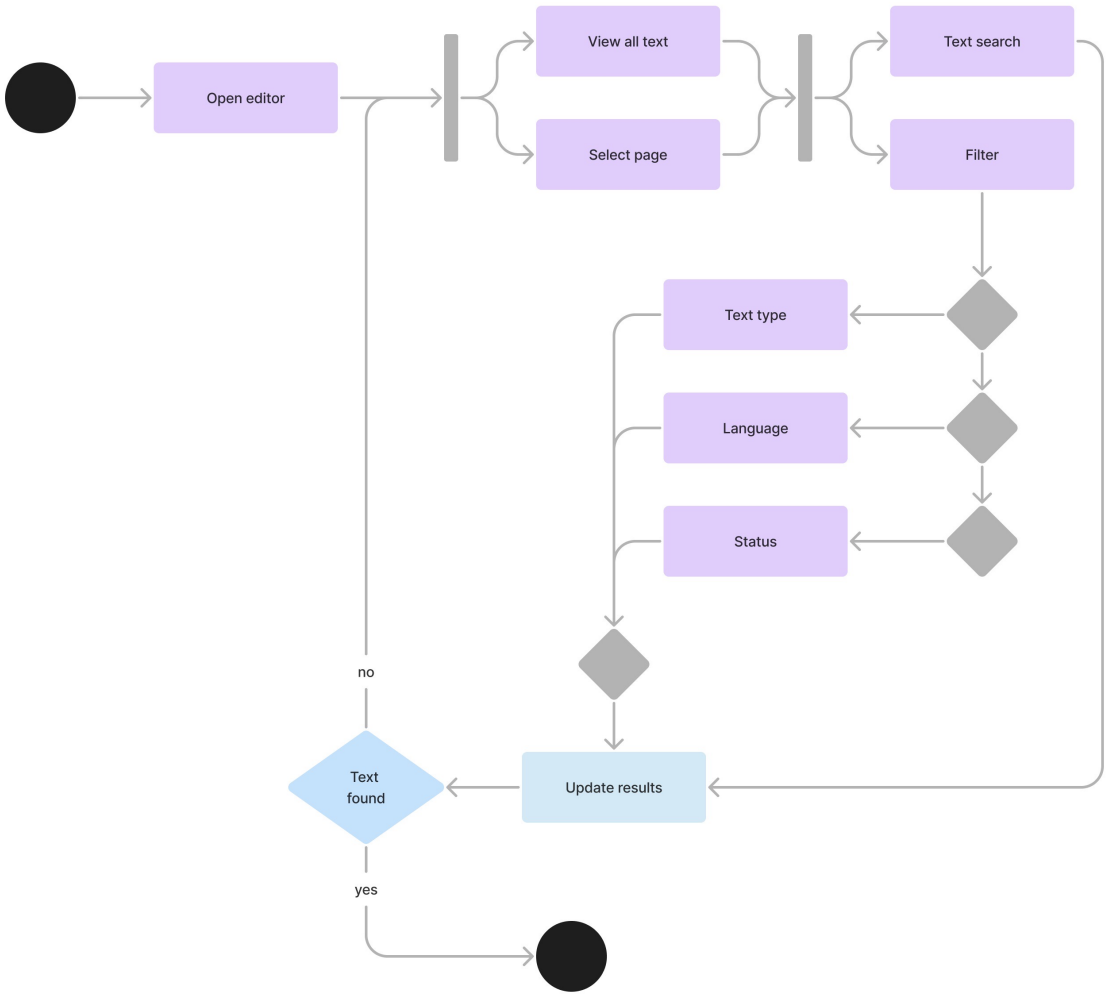


Figure 6.13: Flow of locating a text.

Editing a text

Figure 6.14 visualises the flow of editing a text, which is done by first selecting an item, the text string to edit. The editor appears upon selecting the text string where users can choose to either edit, which changes the text string directly, or suggest, where it will be highlighted that it is simply a suggested improvement that is not connected to any task or change. Other editors then review suggestions. Then, users edit the text and can insert templates or variables when suitable through the editor. Variables are inserted where they exist in code, and templates can be inserted when a text shares a similar structure to other text types or strings. Next, the editor checks for potential issues and alerts the user if problems are detected in connection

to the guides. Once the text is free from issues, users assign status: review, in progress, or final. However, the status is automatically assigned to review if the user edits in suggest mode. When assigning a review status, users can also tag a reviewer responsible for reviewing the changes. Lastly, users can add a comment to the edit that has been made and save it.

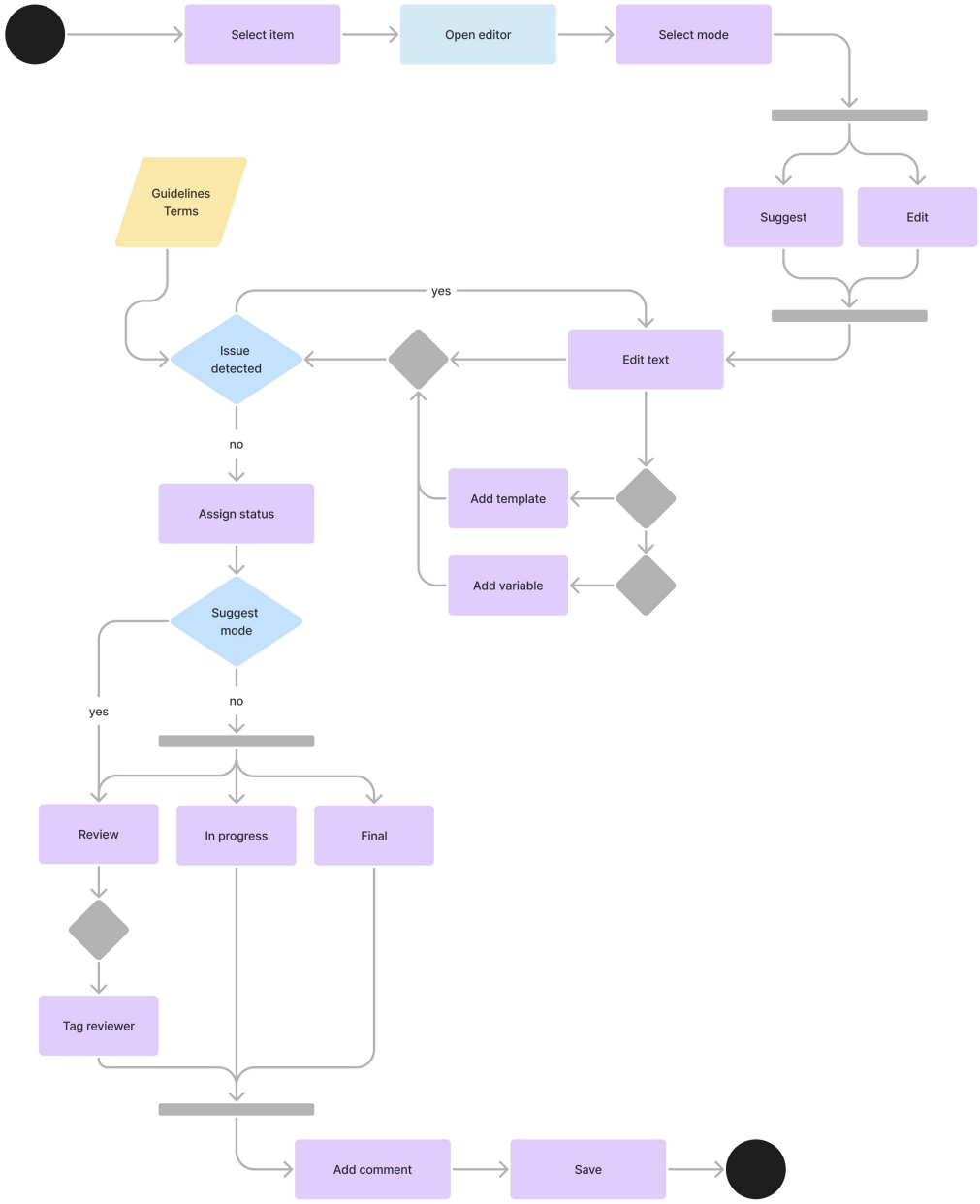


Figure 6.14: Flow of editing a text.

Adding a new text

Figure 6.15 visualises the flow of adding a new text. Adding a text requires some additional steps to align the text with the connected system. First, users navigate to the editor and select add text to open the modal in which text is added. Then,

users define the text type and add the location in the system where the text exists, e.g. the page, the element, e.g. in which component it exists, and the ID, e.g. the connected JSON identifier. If users add an error message, they must also add the connected troubleshooting message. If they select email, they do not need to define location, element, or ID. The critical definition is the connected text which shows how the email is triggered. Text types other than email and error message requires the same details. Then, users add the text in the editor where the same features are available as in the previous flow of editing a text: add a template or variable, and detect issues based on the guidelines within the system. Then, users assign status and comments before saving. The added text string will be visible in the editor and placed according to the defined page and element.

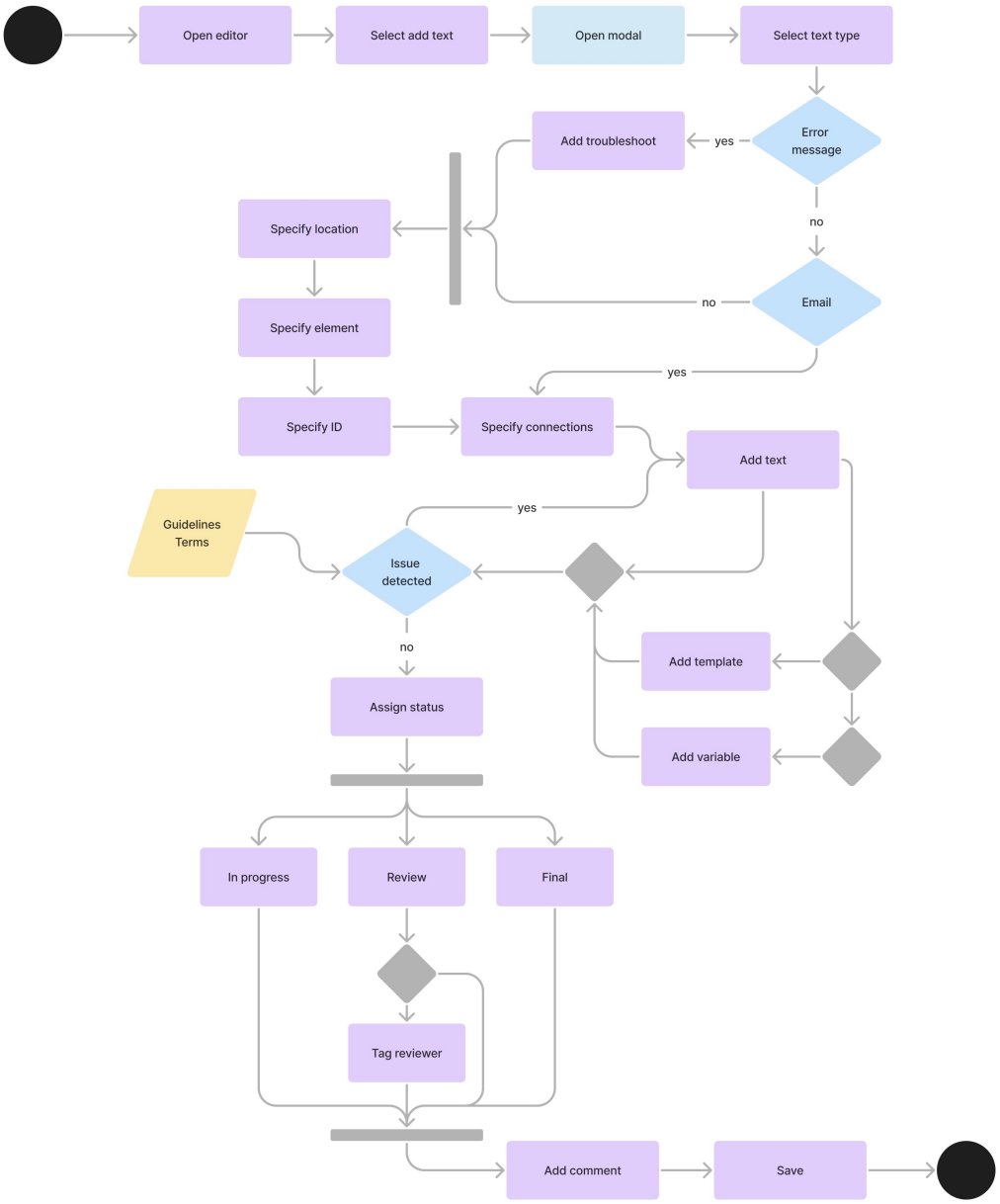


Figure 6.15: Flow of adding a text.

6.6.4 Wireframes

After establishing a solid ground through digital solution sketching and user flows, wireframes were created to map out the structure, features, and overall layout to enable early evaluation and refinements. A text management tool will include much data. Thus, information, content, and hierarchy were prioritised.

Wireframing was approached through atomic design, enabling reusability and simplifying changes, reducing time and effort. Furthermore, it simplifies scaling designs [45]. Thus, it was applied in an early stage of prototyping to facilitate the transition to a semi-functional prototype; see figure 6.16 for an example of using atoms, molecules, and organisms.

Colour or style was not taken into consideration. However, some labels included real data to promote familiarity and allow testing of their understandability in early evaluation. A selection of the initial wireframe screens is presented in section 6.6.5.

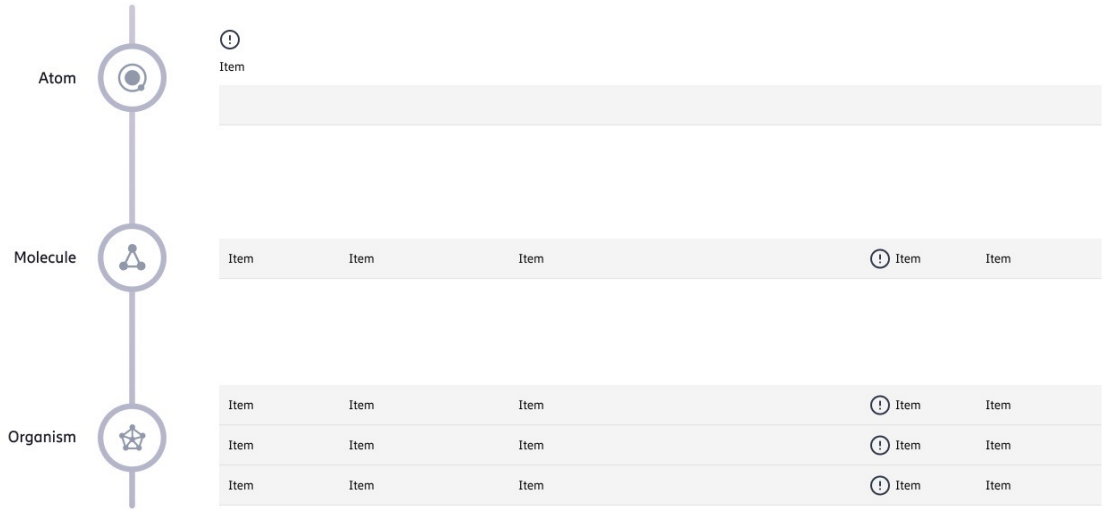


Figure 6.16: Atomic design in wireframes

6.6.5 Evaluation

Wireframes can be used to evaluate and improve navigation and information architecture, especially when these are fundamental to the design [67]. For example, as a text management tool may contain large amounts of data, users must be able to navigate easily and find what they are looking for; the information mechanisms and architecture are fundamental. Furthermore, wireframes focus on a design's structure and general elements [43]. Thus, the main purpose of evaluating the wireframes was to generate feedback about the information architecture, including how the content is organised, structured, and labelled.

While expert reviews can be conducted in several ways, this study utilises design walkthroughs as they are flexible and can uncover potential problems by addressing

user flows, intended features, and design guidelines. Experts can anticipate intermediate states between the wireframes provided [40].

6.6.5.1 Participants

The wireframes were evaluated through expert reviews, including experts from different disciplines. However, while participants in this evaluation are experts within the field, they are also the ones who will use the solution. Thus, they can evaluate the solution based on their expertise and prior experience in the field while providing insights about how it aligns with their needs and solves experienced challenges. The design walkthrough was conducted separately with five participants, including four from UX and one from FE. Throughout the studies, it became apparent that UX will have primary involvement when managing text for NMP. However, one participant from FE was included to ensure the structure would be correctly implemented and aligned with their current ways of working. Accordingly, when conducting expert reviews, diversity is more important than a high number of participants [40].

6.6.5.2 Procedure

As all participants were familiar with the scope of the thesis, no extensive introduction was given. Instead, the participants were given a brief introduction to the wireframes and an overall simulation of the main user flows. Most sessions were held physically at the Ericsson office. However, two sessions were held online through Slack. Each evaluation lasted approximately 30 minutes. The evaluation was limited to the main user flows locating a text and editing a text. The wireframes presented to the participants can be found in figure 6.17, 6.18, and 6.19.

Participants were encouraged to think aloud or ask questions during the walkthrough to uncover potential issues. Furthermore, a few support questions were formulated in advance. Participants were asked to explain their thoughts and potential improvements as issues arose. During the evaluation, the moderator was responsible for continuing the walkthrough, answering questions, and evoking discussion. The person who was not assigned the moderator role was responsible for taking notes.

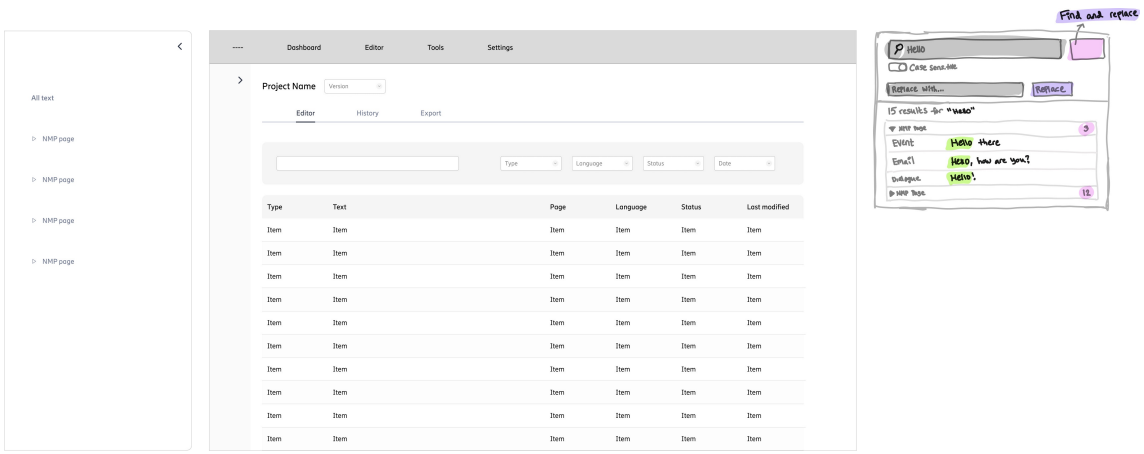


Figure 6.17: Editor

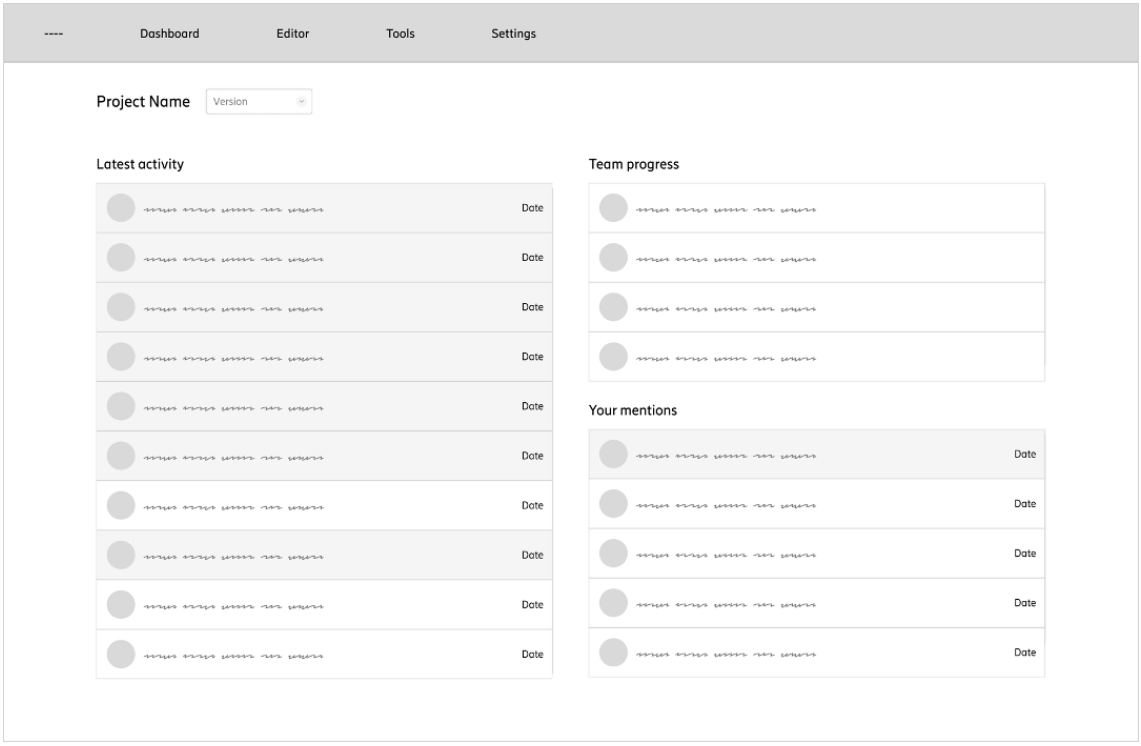


Figure 6.18: Dashboard

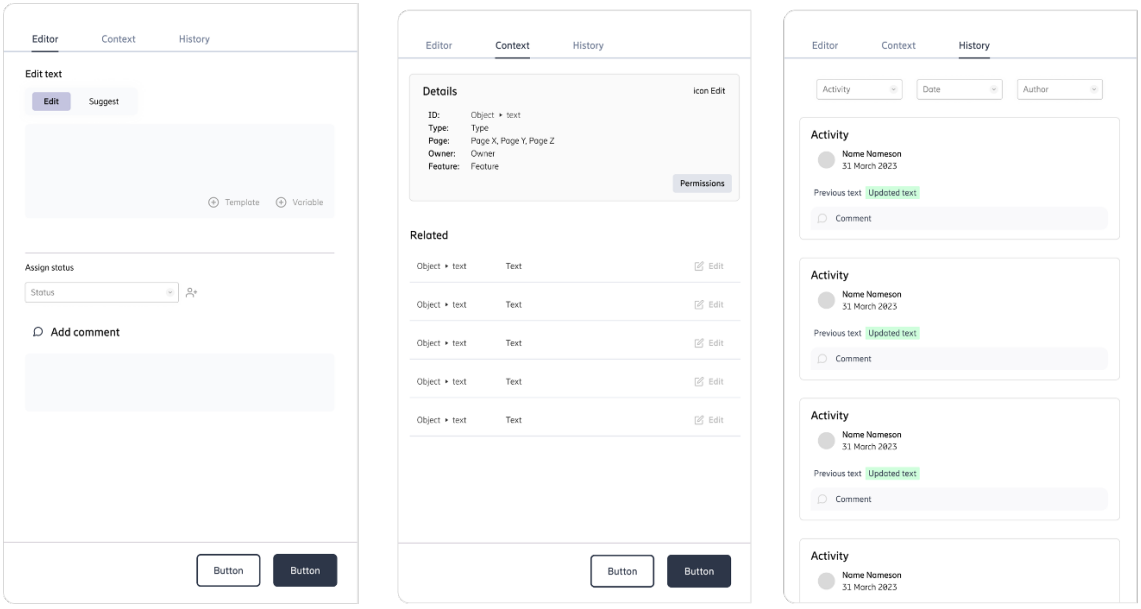


Figure 6.19: Side sheet editor

The sessions were not recorded. Thus, the notes served as the main foundation of the analysis. Affinity diagramming was utilised in this analysis. Insights were highlighted from the notes and clustered into themes. This was done iteratively by reviewing and restructuring sticky notes to discuss multiple possibilities. Once themes were reviewed and finalised, the affinity diagram was refined.

6.6.5.3 Findings

Three themes emerged from the analysis; Structure, Level of information, and Features. The main findings from each theme are presented below, followed by a list of improvements based on the findings. Overall, participants shared positive feedback and expressed that the current state addresses several of their problems well. However, this section focuses on the highlighted concerns and areas that informed iterations and improvements.

Structure
There were some concerns regarding the general structure of the system. For example, one participant questioned the need for two drawers. As the left drawer includes local navigation, another participant was concerned about recognising the current location when the left side sheet was hidden.

Furthermore, one participant asked how the table items would look if the item included an extensive text string. While no issues regarding partly displaying text strings were brought up, some considerations regarding pagination followed.

It became apparent that it is imperative to consider the ID and its present context. The context regards its path, in which objects, such as a modal, form, or card and nested objects, such as enable, disable, or delete, are articulated. This path is used to locate items in code. Closely related to this, one participant from UX pointed out that the ability to navigate between the main file and its child file was beneficial. On the other hand, another participant suggested being able to search by feature rather than by page. However, such functionality would imply someone to tag all features to connect them to items. Further questions arose regarding situations where a page must be removed or if the content is moved from one page to another. While pages are seldom totally removed, feature rearrangements happen occasionally.

Level of information
While participants expressed that the related text information is highly appreciated, there were concerns regarding how writers should know where the newly written text ends up or how to understand where this button, for example, is located in the system. Furthermore, one participant mentioned that knowing what triggers snack messages to write the text would be beneficial. Other discussions arose regarding how details and related text would be specified if new text or a new page was added.

One participant mentioned that it might be beneficial to include ID in the column as it is a unique identifier. Interestingly, a participant from UX brought this up. Other information requested was to see what permissions other teams and individuals have. Regarding future translation, one participant mentioned that such system information is unnecessary. In contrast, another participant was concerned about how the interface would adapt when additional languages entered the system.

Features
There were divided opinions regarding the text status functionality. While two participants expressed that it would be helpful, one participant pointed out that it is a risk of additional work and time if users need to change the status manually. All

participants highly appreciated the integrations of variables and templates. Thus, a suggestion was to enable working with placeholders similarly.

Something that was brought up by the majority of participants was the possibility of connecting text items to Jira tickets. As Jira is the main documentation channel, it could add another context level if such a connection were made possible. Furthermore, one participant expressed that it is equally important for Slack to remain the main communication channel.

List of improvements
A list of improvements was formed based on the insights, including:

- Make the local navigation constant to promote recognition over recall
- A more explicit component hierarchy to align with FE's path structure and to supplement the context
- Editing on a page level to enable adding and deleting pages and moving content from one page to another
- Revise labelling and table columns so it aligns with current expectations rather than future ones
- Enable Jira and Slack connection to reduce redundant documentation and communication
- Add placeholders to the editor
- Add permission transparency to improve collaboration and so users know who is allowed to do what
- Make it possible, but not required, to tag features

6.7 Semi-functional prototype

The wireframes were refined according to the list of improvements after the evaluation. Thus, the wireframes were updated before moving on to a semi-functional prototype. All of these minor changes were easily implemented as the wireframes utilised atomic design. Furthermore, adhering to atomic design facilitated the transition from wireframes to semi-functional prototypes. Some components were added or removed according to the improvements list. However, the majority were refined and combined with additional style guides. See figure 6.20 for an example of updated atoms, molecules, and organisms.

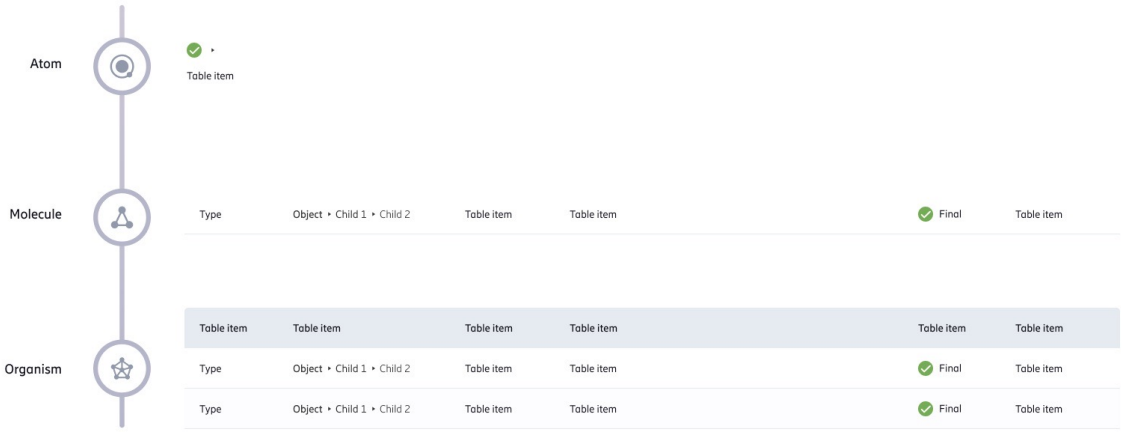


Figure 6.20: Atomic design updated.

While high-fidelity prototypes are interactive mockups of the final interface [40], the following section will present a semi-functional prototype. The semi-functional prototype focuses on the primary user flows presented in Section 6.6.3: locating a text, editing a text, and adding a new text. The prototype includes additional screens that are less detailed or contain limited interaction.

The semi-functional prototype mainly serves as a means of communication to envision how the requirements can be addressed. However, this section briefly describes the different screens and functionality, whereas section 7 presents a more extensive rationale on how the design addresses the requirements. The goal of presenting the semi-functional prototype prior to recommendations is to provide an overview and high-level understanding before detailing all features and rationale. Data are replaced with placeholders to preserve Ericsson's privacy.

The global navigation presented in figure 6.21 shows that the system consists of a dashboard, an editor, and additional tools to manage text management. To the far right, settings are provided to access setup, including team members and uploading files.

Figure 6.21: Global navigation.

Dashboard
Entering the system opens the dashboard. Users can retrieve information about the latest updates and mentions within the dashboard, including project progress statistics (Figure 6.22).

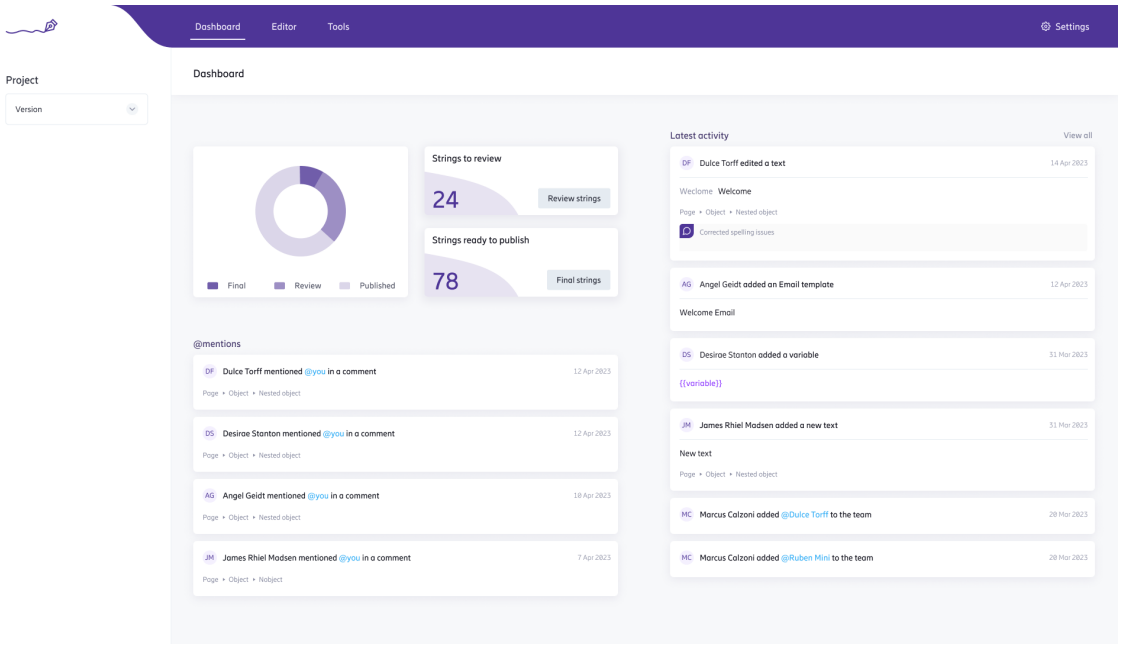


Figure 6.22: Dashboard.

Editor
Figure 6.23 presents the editor, which is where text strings can be located and edited. One of the refinements from the evaluation was to make the left-side navigation constant. Users can navigate through content related to specific pages and components in the side navigation. According to the feedback from the evaluation, the component hierarchy is extended, feeding additional context (paths) to the local navigation and the table with text strings. While the language column was removed, an ID column was added. Path and ID provide context to the text. To locate text strings, users can navigate in the local navigation or by utilising filters in the search bar.

A side sheet slides in from the right by clicking a table item, allowing users to edit or suggest a text with the possibility of inserting templates and variables. The user can further include placeholders while editing, implemented after evaluation. The user can also tag people or comment, including adding links to Slack and Jira, which was questioned in the evaluation. This way, current tools can remain primary for communication and documentation, while the text management tool allows linking relevant tickets or threads. Users can further tab between details to get context to the text string and history, to see previous text string-specific edits, comments, etcetera. See Figure 6.24 for possible actions within the side sheet.

Within the editor, users can add a new text (Figure 6.25) and export text (Figure 6.26). Users can also access the history for all text strings (Figure 6.27).

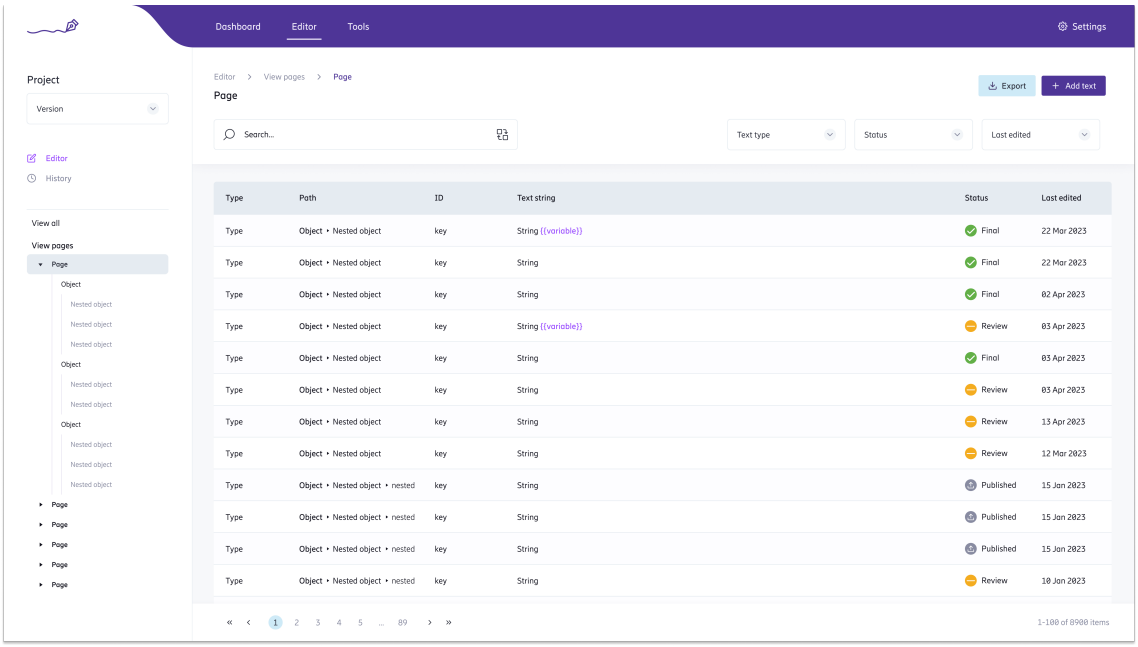


Figure 6.23: Editor.

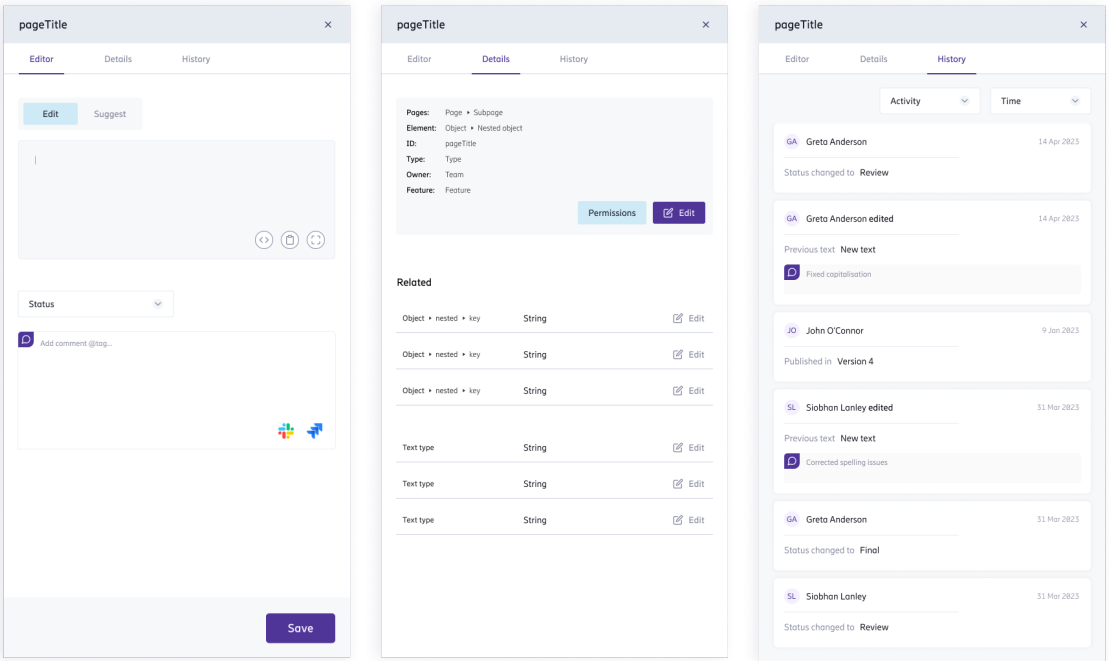


Figure 6.24: Side sheet that slides in from the right where users can edit text, see text details, and text history.

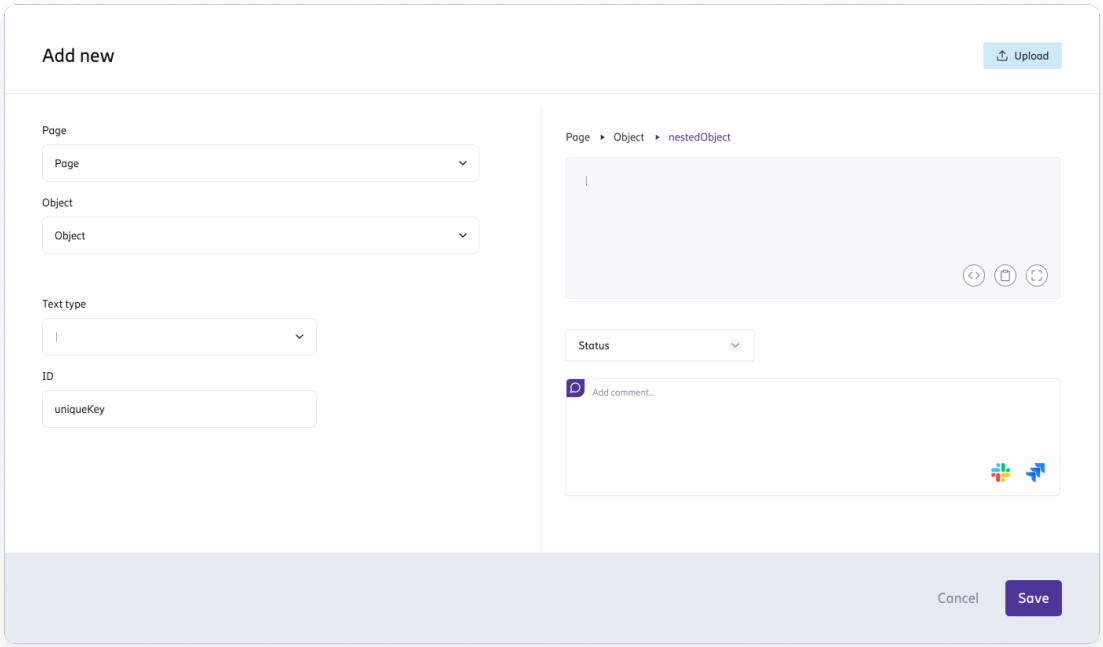


Figure 6.25: Modal for adding new text.

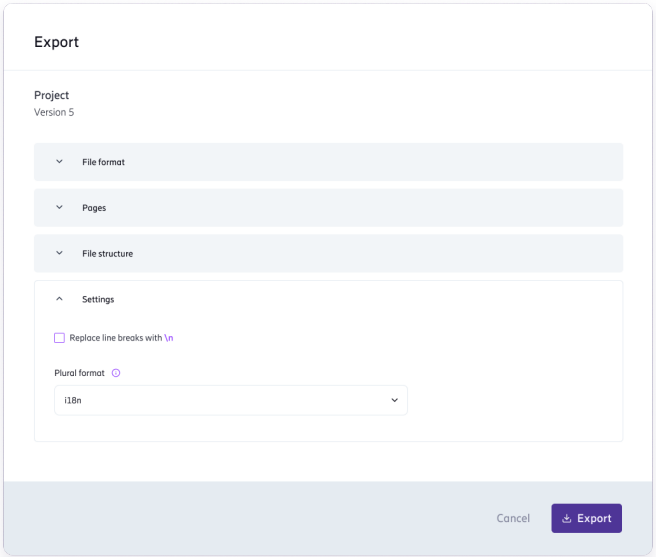


Figure 6.26: Modal for exporting text.

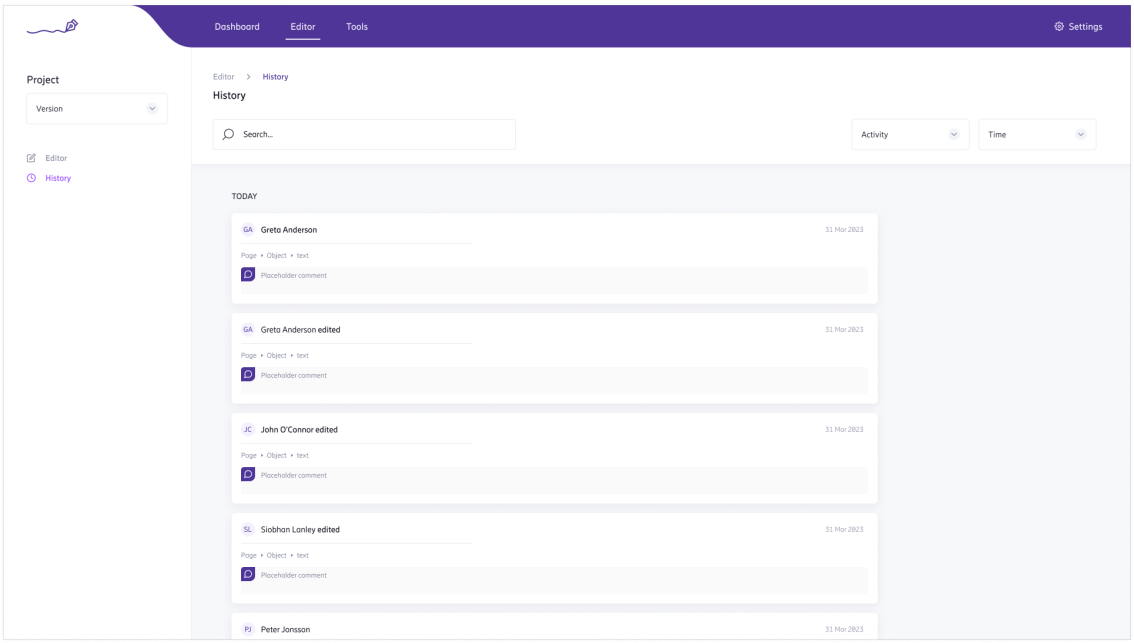


Figure 6.27: Text history for all text strings.

Tools

Tools, formerly named repository, is where users can set up templates, guidelines, and variables, which can later be retrieved from the editor. The example in figure 6.28 illustrates the space for managing variables. Variables can be set up so that UX can easily indicate variables in the text while FE can export them to the correct format.

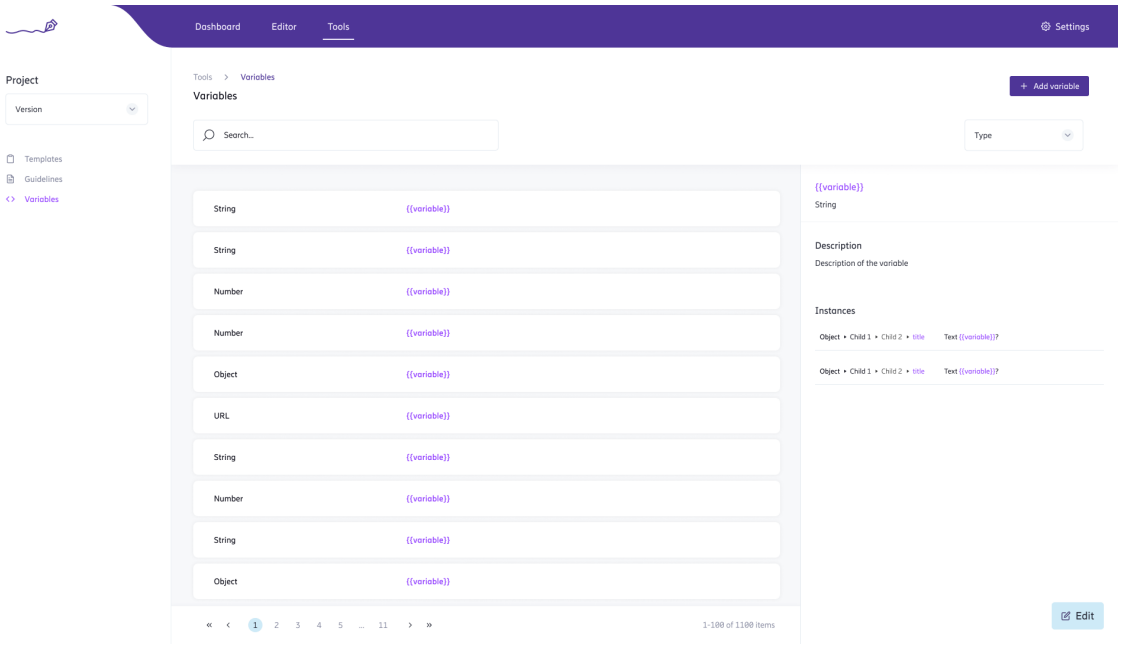


Figure 6.28: Repository for managing variables.

Settings
Within settings (Figure 6.29), users can manage their uploaded files, add new files, delete existing ones, or move content from one page to another. This was a result of the evaluation, as the content of the related website may evolve over time. Users can also access teams and team members, including their roles and permissions in settings. Admin users also manage permissions on the same page. See figure 6.30.

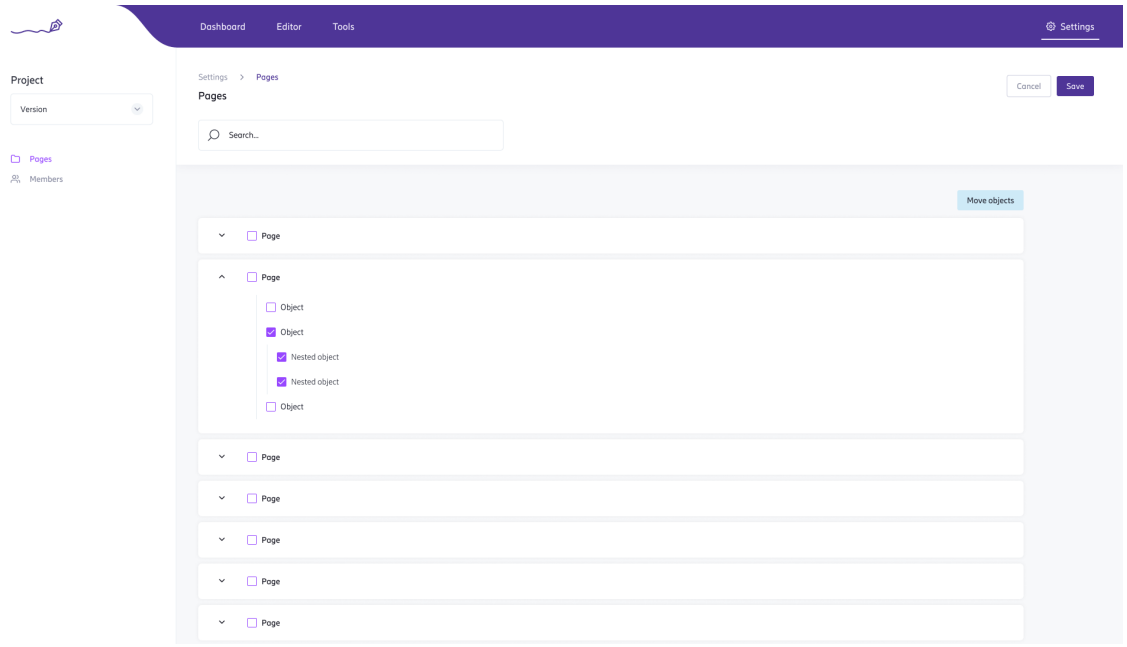


Figure 6.29: Settings for managing files.

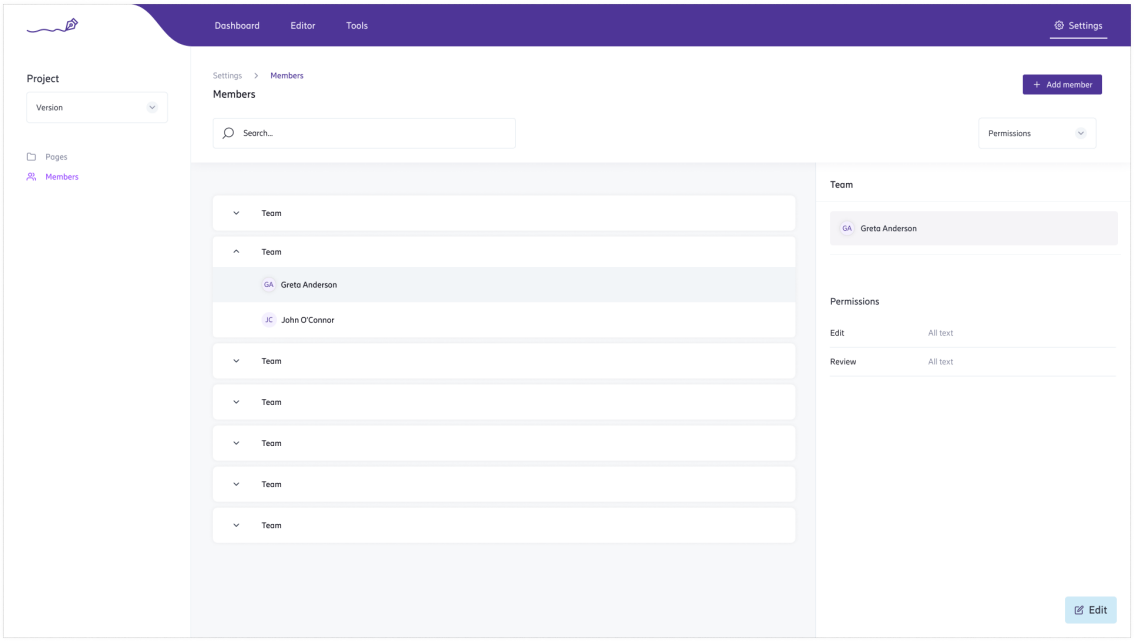


Figure 6.30: Settings for managing team members and permissions.

6.7.1 Practical implementation and current software

The text management tool must accommodate established software, and it is important to consider the practical implications and how established processes may change due to the introduction of a new tool. However, streamlining processes and synchronising teams also depend on all teams using the text management tool. There are areas of development where teams beyond UX and FE can be further included.

The tool is independent of the NMP, which is one of the requirements and something that must be considered in a way that streamlines processes rather than results in additional workload. Furthermore, the tool is not designed to communicate and receive data from other software. Ericsson uses Excel, Figma, Slack, Jira, JSON, and Confluence throughout their current text management process. Excel can be omitted entirely from the process when introducing a text management tool. However, Figma is an integral part of the UX team's processes and cannot be omitted. Nevertheless, the ways of working with Figma will change. The UX team will design the interface as usual in Figma and can test how different text strings will look in their components, etcetera. However, Figma will no longer be the reference for the text in NMP. The solution does not communicate with Figma, and thus the UX team does not need to consider ensuring the correct text in their Figma files. Instead, all final text strings must exist within the text management tool, and when a team needs to access or edit a text, it is done through the tool.

Similarly, despite discussions in Slack, the text management tool should act as a source of truth where people access guidelines, terminology, and the correct texts. Slack will still be used for discussions. However, additional discussions should not be held in the tool to remove redundancy and potentially contradictory information. Instead, users can insert a link to the Slack thread in which the discussion was held. In the same way, Jira tickets will still be used for bugs and improvements; however, they can also be linked to the tool for better context both for the ticket and for the text string.

The ways of working with JSON will change by making it more autonomous. The tool will not communicate with the repository; however, it allows uploading and exporting JSON files to reduce manual work in maintaining the files. Furthermore, maintaining the JSON files becomes more of a joint effort between teams. However, FE must still export the files and upload them to their repository to publish.

Confluence is used in several processes for many different purposes. Thus, it will not be fully omitted. However, ideally, the terminology documents should be uploaded in the tool rather than Confluence to ensure editor integrations are kept up to date.

7

Result

The following section details the study's main findings; a set of design recommendations for a text management tool focusing on streamlining text management and improving cross-functional collaboration. Recommendations were informed by data collected through interviews, competitor analysis, workshop, and evaluation. Each recommendation serves as an actionable plan based on established requirements. Thus, the recommendations are not exclusively UI-centred. Addressing identified challenges and requirements demands considering features, how the solution fits into current processes and technical aspects of implementation.

Each recommendation is provided with an explanation of their rationale, specifying to what requirements it relates and how they align. Furthermore, design proposals are presented, visualising prototypes of suggested features and motivated through design principles that impact design decisions. Design principles such as learnability, ease of use, safety, and accommodation were considered, together with psychological principles of perception and memory and attention. Lastly, further implementation considerations are highlighted, which cannot be visualised through the prototypes. Table 7.1 summarises the design recommendations.

Table 7.1: An overview of the design recommendations.

Number	Recommendation
DR#1	Provide users with direct editing and file export
DR#2	Allow users to customise file structure
DR#3	Integrate JSON syntax
DR#4	Reduce cognitive load by considering workflows and clutter
DR#5	Illustrate links between associated texts and make them accessible
DR#6	Allow users to manage text based on release
DR#7	Provide users with the ability to retrieve text in multiple ways
DR#8	Allow users to make universal changes
DR#9	Integrate guides within the editor
DR#10	Provide users with reusable components
DR#11	Continuously communicate editing history
DR#12	Allow users to communicate text-specific decisions and concerns
DR#13	Allow users to be assigned different roles

DR#1 Provide users with direct editing and file export

Requirements addressed

1.3.1. The solution must enable JSON integration
7.1. The solution should support I18next
7.2. The solution should enable exporting text for translation
8.2. Text files must be low maintenance
8.3. The solution should enable non-technical
9.1. The solution must store text locally

Rationale

A fundamental concern regarding text management in organisations where text is stored in code or other files referred to in code is that only people with access can make changes. As a result, it requires code-literate people. As developers are not necessarily responsible for ensuring text quality, organisations will benefit from allowing non-technical teams to update text.

Text seldom requires extensive changes. Thus, updates should be easy and time efficient. However, there are implications with a direct editing solution as it must align with current development practices. Developers must be able to retrieve correctly formatted files for people to edit text outside the code. Otherwise, developers are faced with an additional workload.

Currently, several tools support similar direct editing possibilities that can be automatically pushed to code, utilising an API to access data. However, this recommendation applies exclusively to organisations requiring local storage. Thus, exporting files edited within the text management tool is imperative. Export requires synchronisation with the current text format (see DR#3 for further details about syntax). Nevertheless, it should not complicate the editing process. Direct editing and export will enable developers to reduce text file maintenance, as non-technical teams can provide the content needed. Thus, it allows organisations to delegate workload based on responsibilities.

Design proposal

Ensure users can edit text regardless of coding experience by providing a regular text editor (Figure 7.1). The editor is triggered by selecting a text string from the table and is where team members, regardless of coding experience, can edit text for a particular product or service.

To efficiently implement the updates made within the editor, provide an export functionality. Furthermore, an export functionality requires customisation, such as relevant file formats based on current development practices. Allow users to specify the file or files to be exported. Consider adding certain constraints to hinder potential problems, such as including the option of only exporting final strings to prevent unapproved text from being published. To prepare for potential future translations, allow users to specify file structure, e.g., bundle structure or directory prefix, determining whether languages should be separated and bundled into files or

all languages within pre-defined files. Furthermore, ensure text files can be exported according to external libraries, such as i18next (Figure 7.2). Once files are exported, developers can easily upload the new file in their project directory, and thus, the updated text can be implemented.

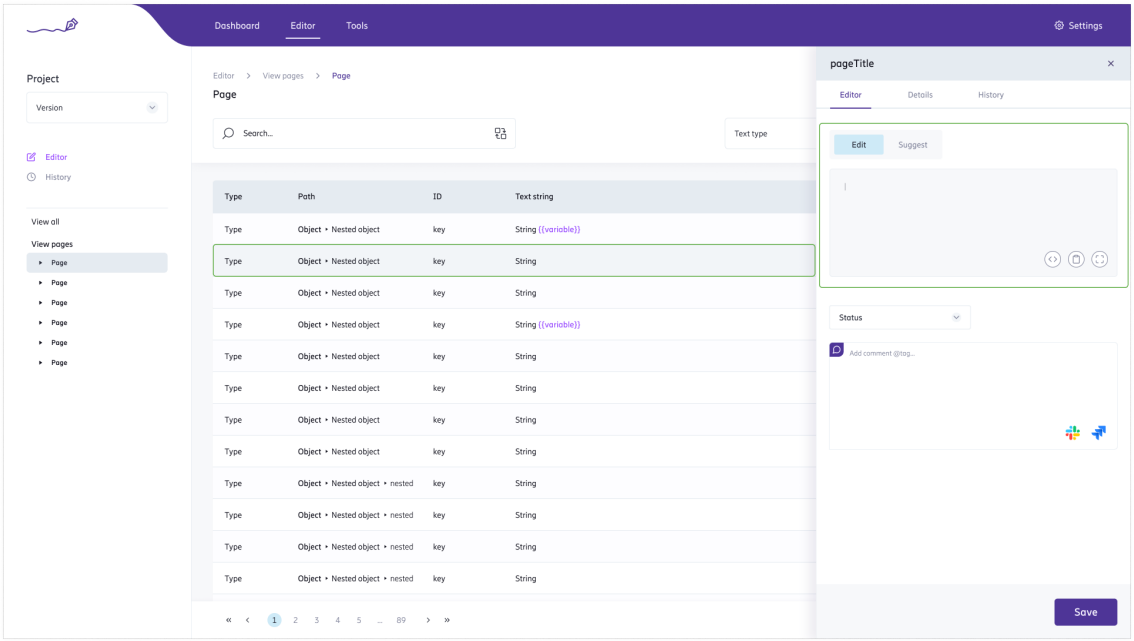


Figure 7.1: Editor triggered by selecting a text string, enabling direct edit.

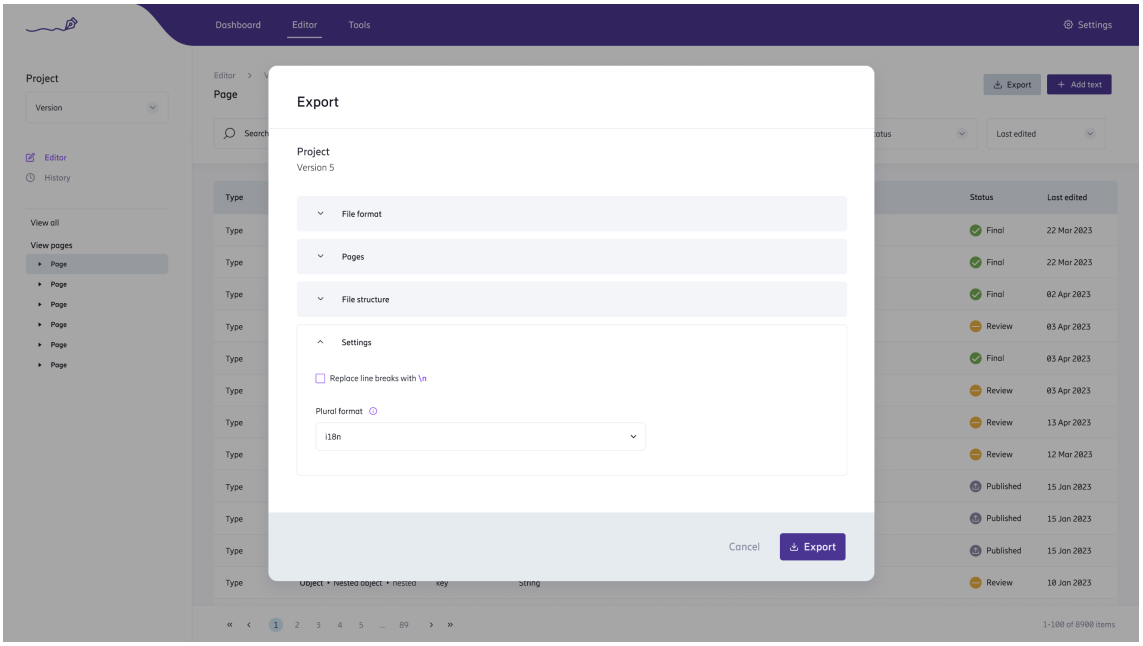


Figure 7.2: Export modal.

DR#2 Allow users to customise file structure

Requirements addressed

1.1. The structure must be aligned with NMP structure

Rationale

Organisations have different ways of developing products and services. This recommendation specifically concerns organisations utilising text files, such as JSON or similar, to retrieve text for their applications.

A product or service's features and content will increase and change over time based on customer needs and organisational goals. As the product or service evolves, the content will be affected; it will be updated, added, removed, or moved. File and hierarchy customisation is necessary to reflect such evolvement, ensuring it aligns with the product or service's content and can be exported and implemented correctly. Thus, it must be possible to expand or remove the number of files, including the possibility to update files ad hoc if any changes have been made that affect their construction or content.

Design proposal

Dedicate a place in the system where files can be rearranged (Figure 7.3). As this feature allows the rearrangement of bigger chunks of content rather than editing text strings, consider separating these actions from the editor.

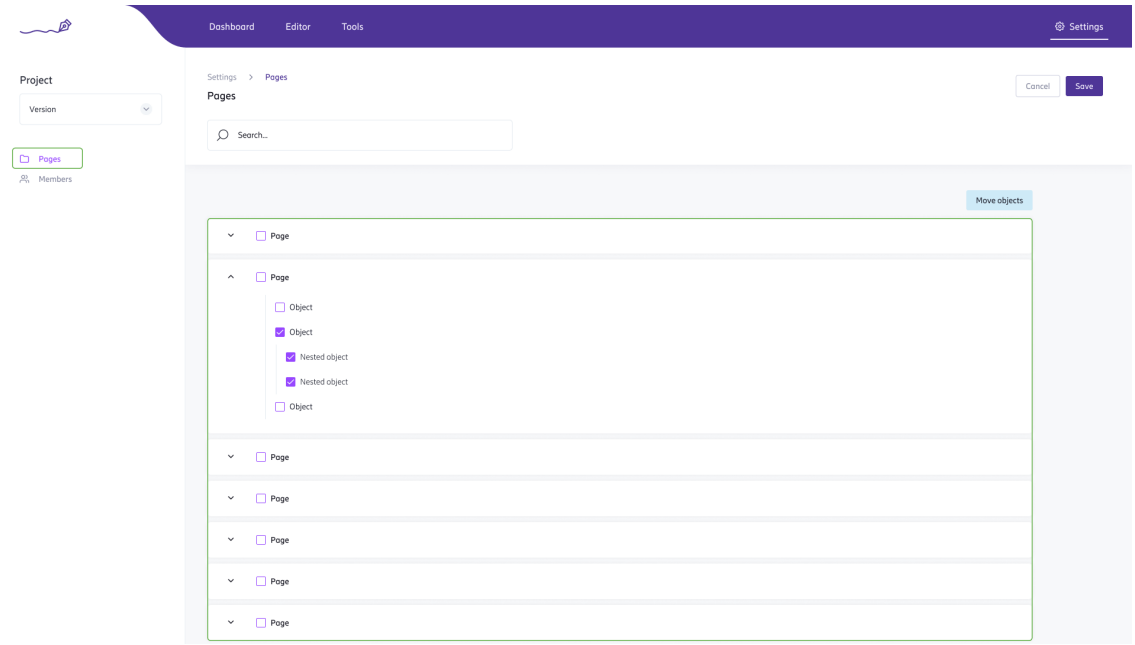


Figure 7.3: Edit pages.

Allow users to add, remove, and rearrange files entirely or partially to align with new features being developed or when content is to be moved from one place to another. Making changes and moving content or pages can result in significant structural changes within the system. Thus, users must be continuously informed by their actions and what content they are moving, promoting recognition rather than recall (Figure 7.4). Organisations are also encouraged to constrain the number of users who have access to such action. Furthermore, connecting the file structure to local navigation in the editor allows users to navigate based on the file structure and its content. For example, Ericsson Dedicated Networks structure their text files based on NMP pages. Thus, users can navigate through these pages and their component hierarchy in the local navigation located in the left sidebar while in the editor. It encourages familiarity as content is arranged in a way that aligns with the product or service being developed, further promoting learnability as it aligns with users' mental models. However, allowing users to navigate based on file structure makes file customisation even more important, as outdated files or content may result in confusion.

This proposal requires that developers have structured and named files and elements in a way that reflects the website or product. If current structuring practices are perceived as illogical, consider allowing users to rename files and components in the text management tool to reflect the real system better.

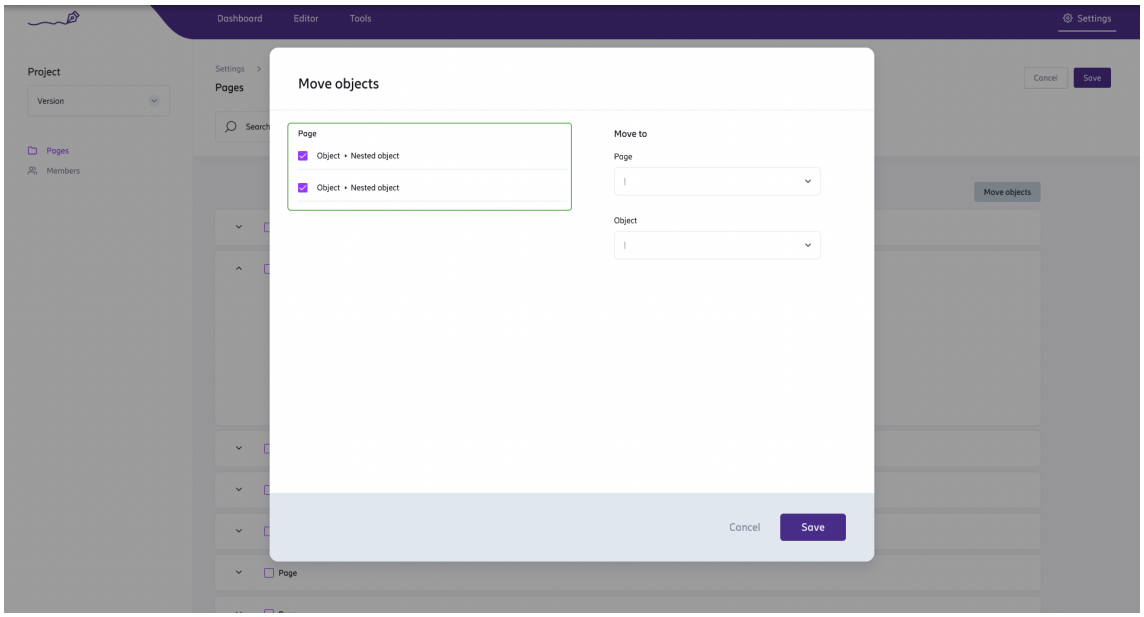


Figure 7.4: Move objects.

DR#3 Integrate JSON syntax

Requirements addressed

1.3. The solution must enable integrations with current tools
1.3.1. Must support JSON integration
9.2. The solution must format text according to JSON syntax

Rationale
The recommendation builds on customising file structure and applies exclusively to organisations that utilise JSON to store text files. Using existing text files to build the tool's foundation was heavily inspired by localisation platforms. Several localisation platforms allow users to upload JSON files. The system converts the files for the in-app editor, separating the keys (identifiers) and values (text strings). However, further consideration of hierarchical structures is required to address identified issues.

Text strings have unique identifiers; however, they also exist within a hierarchical structure of objects and nested objects within the page-based JSON files. Each file contains content for one page, whereas an object within the page can be a modal, a form, a card, etcetera. This path allows users to understand the context in which a text string exists and provides a path to locate them. The text management tool must accommodate all user needs to synchronise teams and align with established processes. Currently, the focus is on aligning FE and UX. First, the tool must follow the same JSON syntax to ensure that exported files are correctly formatted and that the hierarchical structure within the tool mirrors the related system's files. Otherwise, FE must continue relying on copying and pasting and constantly maintaining their files. Secondly, the JSON structure should not complicate the interface; the hierarchy must be clear, and it must be displayed in a way that aligns with non-technical users as they do not have insight into the text files.

Design proposal
The recommendation primarily concerns technical aspects of system features and supporting file formats. However, it requires considering how to represent the underlying structure within the interface. Figure 7.5 presents the connection between the underlying JSON files and their representation in the interface. The hierarchy is clearly visualised in the local navigation, allowing users to navigate through pages, objects and nested objects. Creating a visual hierarchy, mirroring the hierarchy within the text files is imperative to design for perception and allow users to categorise and comprehend the information. However, JSON files can be highly nested and complex. Thus, the visible depth should be limited to accommodate short-term working memory. In this example, the local navigation is limited to pages, objects, and one nested object. Further nesting is represented in the table where each text string is connected to its path and ID. When adding new text directly in the editor (Figure 7.6), ensure that all required specifications are included. Add constraints to prevent users from saving if page, object, and ID are not defined.

Figure 7.5: Connection between JSON syntax and the representation in the interface.

All text strings must be correctly tied to their hierarchical structure for the system to work as intended. The structure is automatically connected if they are added by uploading JSON files. However, manual input is necessary if not added through upload, which in turn requires knowing the names of objects and identifiers. Furthermore, the tool directly reflects the names of objects and IDs. Thus, a naming convention should be established to ensure users can comprehend what a name refers to.

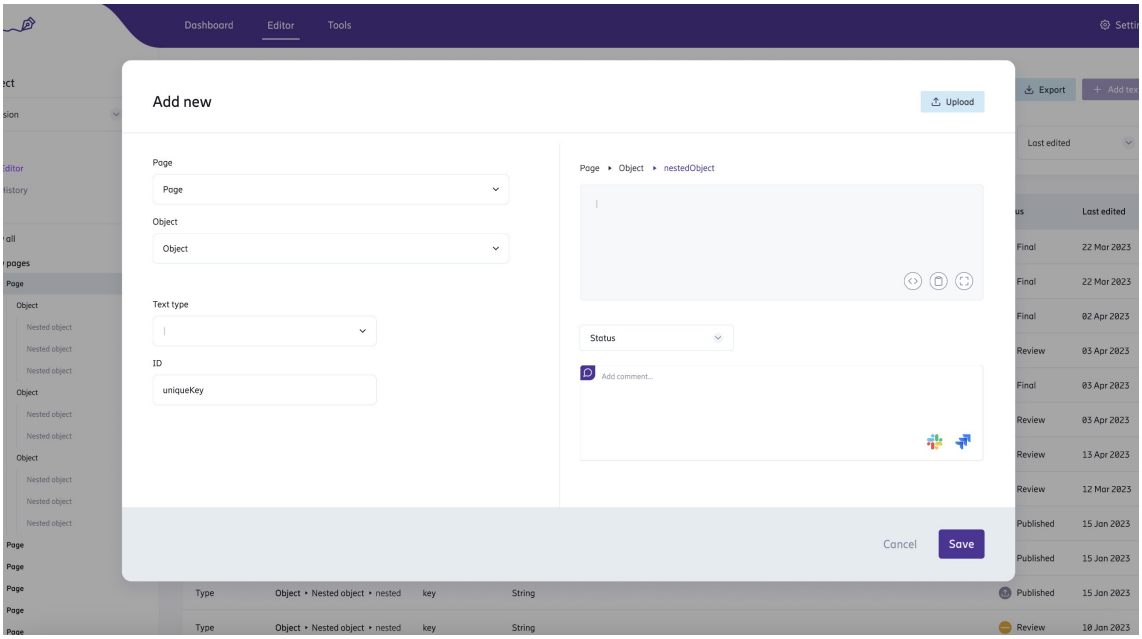


Figure 7.6: Add new text.

DR#4 Reduce cognitive load by considering workflows and clutter

Requirements addressed

8.1. Users must experience a low perceived workload

Rationale

Removing unnecessary noise and clutter allows users to focus on the primary content and tasks. Similarly, the navigation and flow of actions should not complicate the process. Workflows must be considered when designing interactions and tasks instead of designing each action in isolation. Users should be kept from constantly being required to shift attention or navigate between different pages. Designing for desktops allows leveraging screen estate to accommodate workflows and eliminate memory load.

Design proposal

Navigating between pages requires users to shift attention and context, causing potential disruptions in workflow. Thus, minimising navigation and leveraging screen estate allows users to streamline their workflow. In this example, the primary user flow concerns locating and updating text. Thus, all interactions related to the flow are accessible within the primary editor view. In addition, the editor sidebar (Figure 7.7) appears upon selecting a text item from the table; thus, all the contextual information stays visible in the primary view.

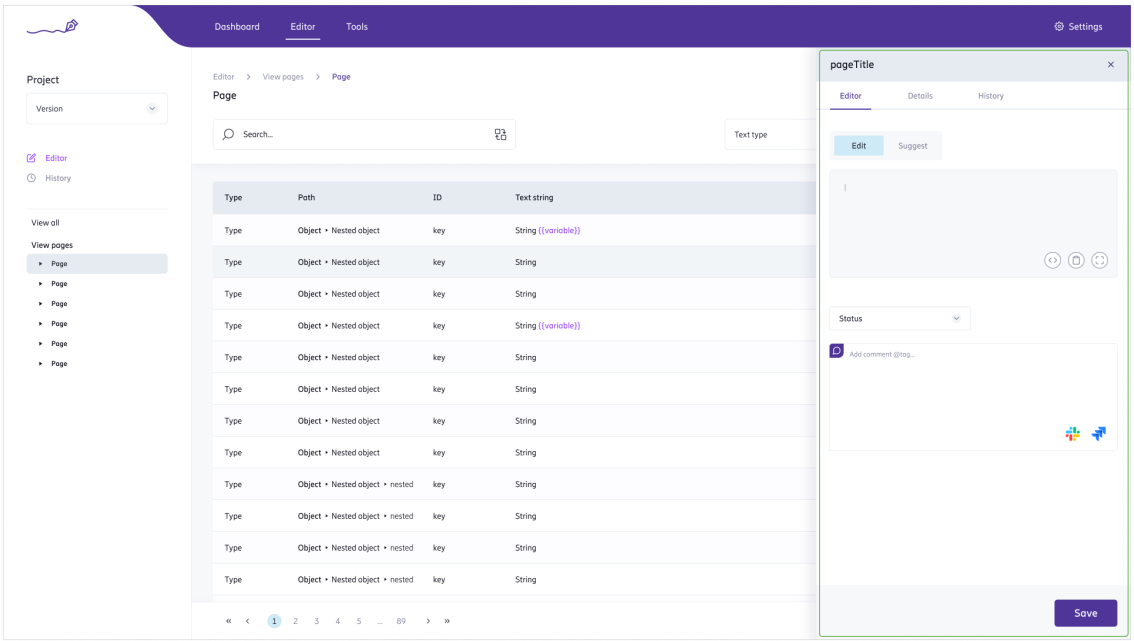


Figure 7.7: Editor as a sidebar allow for contextual information to stay visible.

Grouping information into larger contextual groups can reduce memory load. Groups should ideally be formed based on information and actions related to a specific task.

In this example, the sidebar editor is divided into three areas accessible through tabs. However, information and actions within the tabs should be independent of the same flow. Users should be able to perform specific tasks without navigating between the different tabs. For example, all options necessary to edit a text are grouped within one tab. However, depending on the goal, users may want to access all tabs consecutively throughout a flow, which they can easily do.

Workflows can be further supported by providing contextual navigation when applicable. For example, there are several instances of contextual navigation within the dashboard page (Figure 7.8). Users can click “view all” and navigate to the history page and find all activity logs. Furthermore, clicking the activity items allow users to navigate directly to the location of the concerned item. Users can also access all text strings that need to be reviewed directly from the progress indication. Other examples of contextual navigation include navigating to permissions from details and editing related items (Figure 7.9).

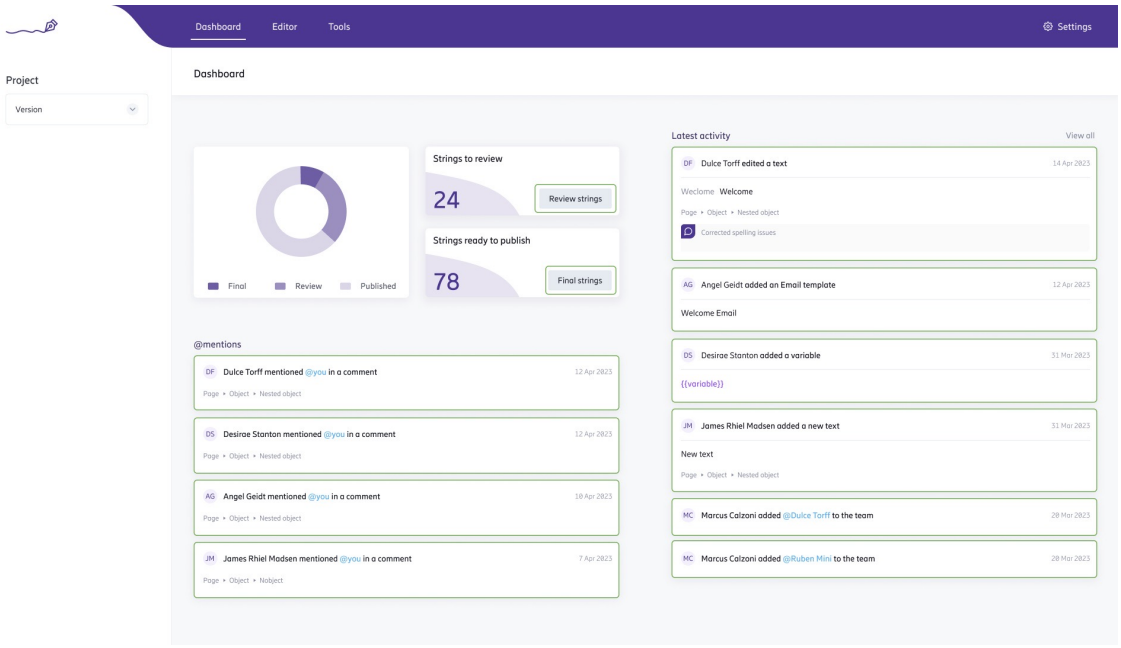


Figure 7.8: Contextual navigation within the dashboard.

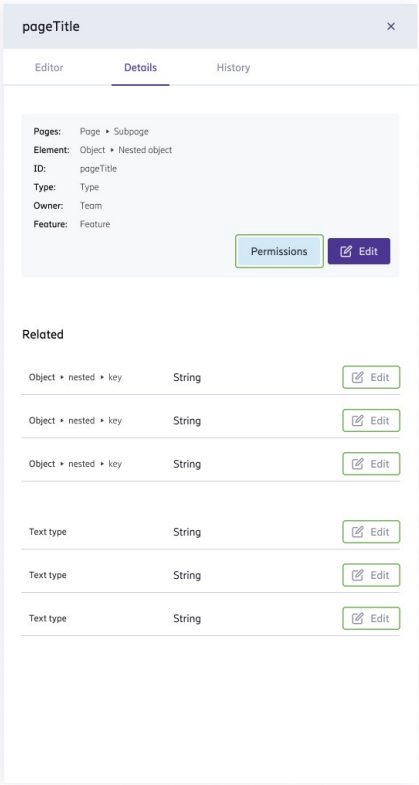


Figure 7.9: Contextual navigation within details.

Users must be able to get an overview of text strings and their contextual information. This example utilises a table (Figure 7.10) to present and structure individual text strings. Tables allow flexibility in use while similarly structuring each item, presenting the contextual information without cluttering the interface. When elements appear within an enclosed area, they are perceived as belonging to a group. Adding borders creates a visual separation between the groups. Thus, each row is separated by borders. However, vertical lines were omitted as text strings closely relate to each column item and present the contextual information. Further groupings may affect perception.

While actions and information should be visible, they can quickly clutter the interface. Thus, they should be made observable rather than completely visible. This example utilises expandable lists throughout the entire interface, allowing users to first get an overview of the actions available before further interaction. For example, the local navigation consists of expandable list items (Figure 7.11) and the settings page discussed in DR#13.

However, the expandable lists should be designed to fit separate flows, allowing users to focus on one task at a time. If an item is expandable, it is indicated by an arrow. Affordances are essential to make sure hidden actions and information are observable.

Type	Path	ID	Text string	Status	Last edited
Type	Object ▸ Nested object	key	String {{variable}}	Final	22 Mar 2023
Type	Object ▸ Nested object	key	String	Final	22 Mar 2023
Type	Object ▸ Nested object	key	String	Final	02 Apr 2023
Type	Object ▸ Nested object	key	String {{variable}}	Review	03 Apr 2023
Type	Object ▸ Nested object	key	String	Final	03 Apr 2023
Type	Object ▸ Nested object	key	String	Review	03 Apr 2023
Type	Object ▸ Nested object	key	String	Review	13 Apr 2023
Type	Object ▸ Nested object	key	String	Review	12 Mar 2023
Type	Object ▸ Nested object ▸ nested	key	String	Published	15 Jan 2023
Type	Object ▸ Nested object ▸ nested	key	String	Published	15 Jan 2023
Type	Object ▸ Nested object ▸ nested	key	String	Published	15 Jan 2023
Type	Object ▸ Nested object ▸ nested	key	String	Review	10 Jan 2023

Figure 7.10: Table with text strings.

Depending on user goals, consider more flexibility by incorporating further tailoring possibilities such as hiding and sorting table columns. Furthermore, in cases where users want to go through numerous text strings, scrolling becomes inevitable. Thus, fix the table header while scrolling to promote recognition over recall. Moreover, every action cannot be hidden in an expandable list. Considering the flow of actions to logically group content into list items is essential to reduce confusion and frustration. Users must be able to sense what actions or information an expandable list contains. Thus, several unrelated actions should not be hidden within the same list, as the label cannot accurately represent its content.

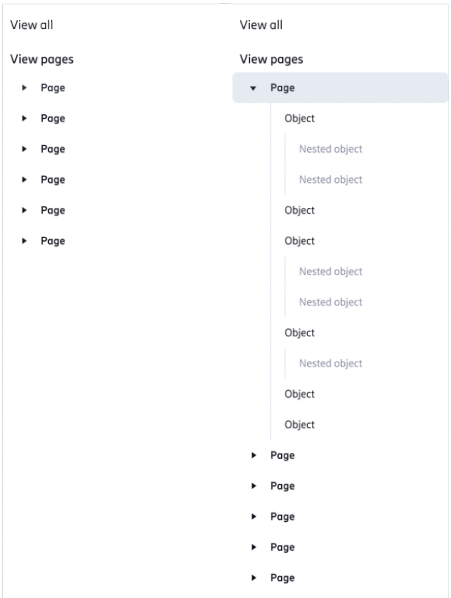


Figure 7.11: Expandable list items in the local navigation.

DR#5 Illustrate links between associated texts and make them accessible

Requirements addressed

6.1. The solution should clarify what will be visible to the users
6.3. The solution must provide context to related texts
6.4. The solution must display how texts are interconnected
9.4. The solution must support managing text for system errors

Rationale

While text strings each have a unique identifier and can be accessed independently, several strings are generally connected throughout an application. Therefore, ensuring that users can quickly locate associated texts is essential to promote consistency and time efficiency.

Text strings can be associated with an element's component, task flow, or states. However, when editing individual text strings, enough information must be provided to understand the context in which the text exists.

Several text strings can be connected to the same action and depend on each other, where one user action triggers several events that include text strings. Generally, the events triggered by the same actions consist of similar text displayed in different mediums or components. Thus, if one must be updated, most likely, all of them must be updated. Displaying interconnected texts within the editor facilitates reviews and ensures consistency.

Similarly, error message functions are composed of an error log used internally to troubleshoot and a message intended for users that the interface will display. One cannot replace the other, as error logs must contain enough information to troubleshoot, whereas user error messages must be understandable and straightforward, unlike error logs. Thus, there must be a clear connection between the two forms of messages generated for the error to synchronise teams and align with technical and non-technical teams' needs.

Design proposal

Users should not be required to recall if a text string has associated items, the number of items, or what items are related to then manually locate them. Instead, promote recognition by providing clear and accessible information and provide a shortcut to editing the associated items. In this example, the editor for each text string includes three different tab options, where details (Figure 7.12) include all the relevant information for each text string.

Associated texts can be identified automatically by mirroring relations from the code. For example, an input is an object containing different properties such as placeholder, label, and hint. Thus, the properties within the input object will automatically become related texts. Providing the error log for an error message removes the need for users to recall it while editing and provides a clear link between them.

However, error logs should ideally be visible from the editor (Figure 7.12) rather than details as it is clearly linked to the task of editing the selected text string. Furthermore, error logs should be omitted as an item in the table. Instead, the table should only represent text visible to the end users. However, users should still be able to use the search bar to search for the contents of an error log.

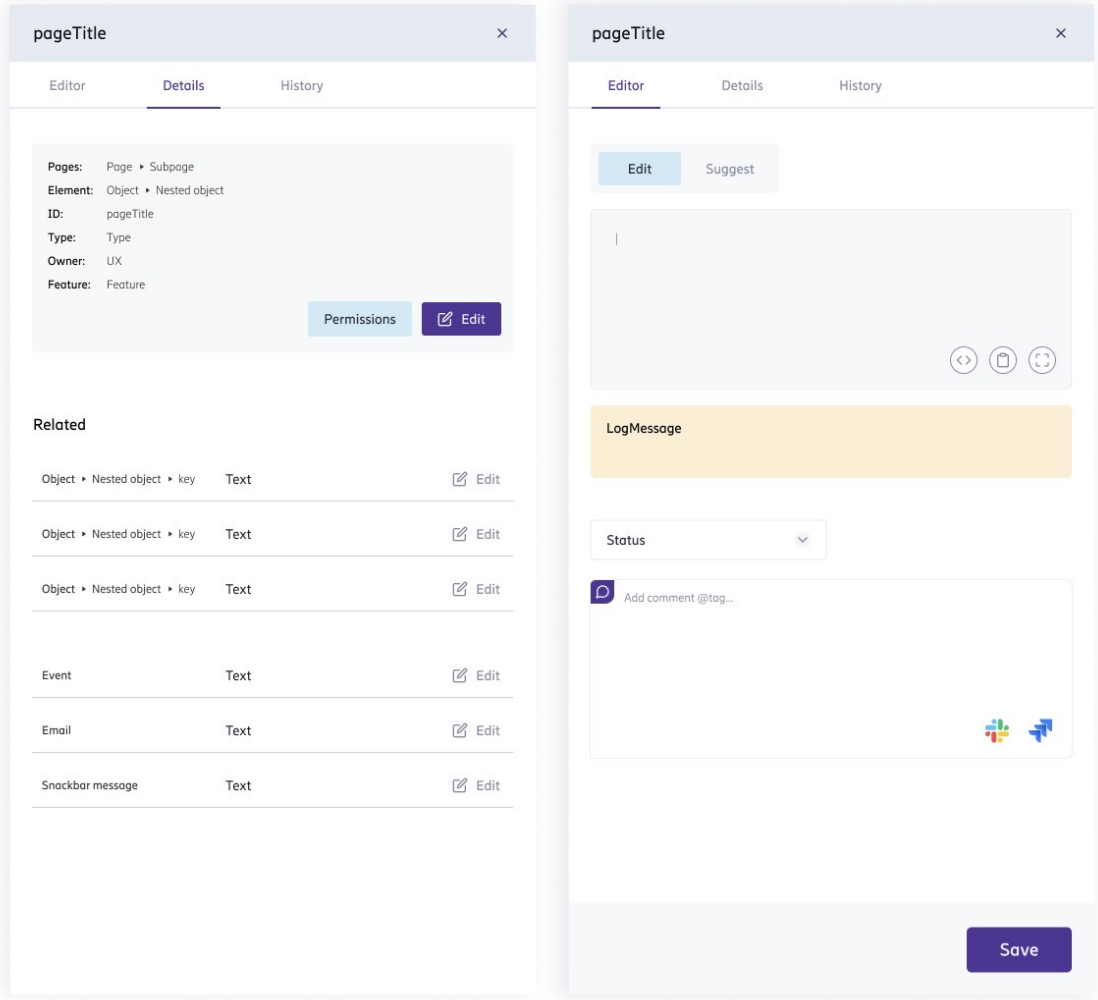


Figure 7.12: Details with related information and editor with error log.

Ideally, richer context should be included for error messages; for example, a description regarding user flow and actions resulting in the error. It would further synchronise non-technical and technical teams, giving non-technical teams a clear context to formulate the error messages intended for users. However, someone would need to review each error message and describe the flows manually. Furthermore, the approach would be unrealistic as thousands of errors may occur, including multiple edge cases which are complex to comprehend fully. Thus, the first action point must be to store them outside of the code and pair troubleshooting messages for internal use with the error messages intended for users to establish a clear link between them.

DR#6 Allow users to manage text based on release

Requirements addressed

1.2. The solution must provide version management
2.4. The solution must streamline the review process

Rationale
Many organisations utilise agile development approaches to develop products and services that meet customer needs and organisational goals. Therefore, text management processes should be shaped to align with sprints and releases, and a text management tool must support versioning.

Tracking and reviewing text within a fast-moving and dynamic environment can result in stress and confusion, especially when text is stored in multiple tools. Storing all text in one place will provide a better overview and reduce the risk of overlooking items during reviews. However, aligning with current ways of working requires working release-based within the text management tool as well, allowing users to identify updates made targeting specific releases, facilitating prioritisation and comparison between releases.

Ultimately, this will promote a shared understanding of the system's current state. In addition, informing users about current and previous releases will enhance user control as they can map how changes within the tool will affect the product or service. Furthermore, storing previous releases will be a safety net, allowing organisations to rely on previous text strings.

Design proposal
Allowing users to create versions within the project enhances control as previous releases are stored and accessible while developing content for the next release. Furthermore, it allows users to locate and recall changes and decisions made in the past. Make it visually clear if the user is located in the current or previous release to ensure no changes are made unintentionally. In this example, it is located in the side navigation, continuously informing users what version they are editing within (Figure 7.13). Users should also be able to freeze versions, constraining users from making changes within a released version.

Inform users about the current state of a text string. Assigning a status provides users with feedback on what items require reviewing, get an overview of recently edited text strings, and informs them of published or ready strings. Furthermore, visually differentiate text statuses so they are easily distinguishable to ensure there are no misconceptions. Text statuses can, for example, include review, final, and published (Figure 7.14). Ideally, items will automatically be assigned review as they are updated, meaning they need to be approved before publishing. However, users still have the option to manually assign status as some edits are minimal and previously discussed. The status will automatically be assigned final when a review item is approved. For an item to be assigned published, it must be exported from the

system. However, consider allowing exporters to specify if exported strings should be changed to published. It increases control and facilitates testing. Furthermore, consider restricting the number of users who can freeze and export so that no inappropriate actions are carried out unintentionally.

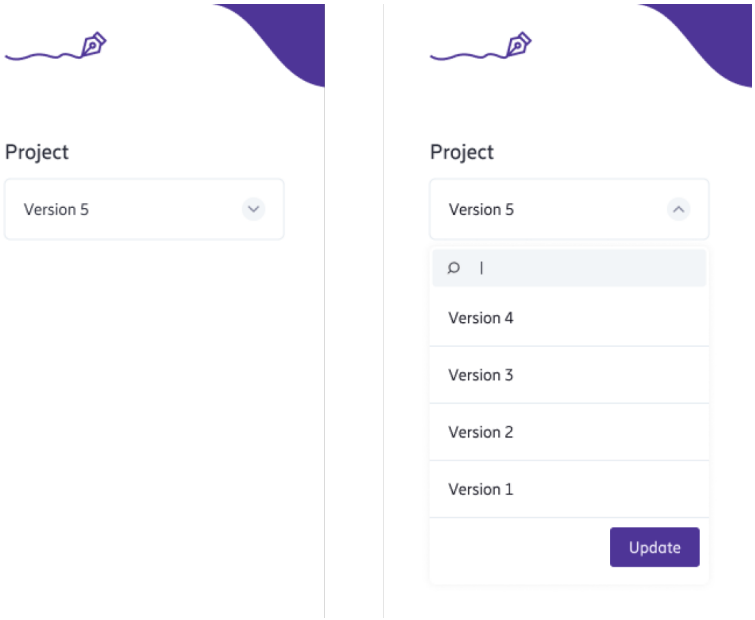


Figure 7.13: Locate and change version in the side navigation bar.

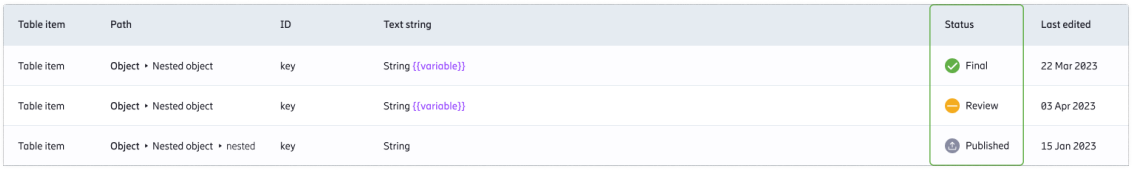


Figure 7.14: Text status.

DR#7 Provide users with the ability to retrieve text in multiple ways

Requirements addressed

1.1. The system structure must align with NMP structure
2.1. The solution must store all forms of text for the NMP
2.2. Users must be able to get an overview of all text
2.3. Users must be able to search and find specific items

Rationale

While the requirement of storing all text depends on how the text management tool is implemented into an organisation, it also implies considering the amount of data. Migrating all text storage from several tools to one universal "single source of truth" can significantly increase productivity. However, as a result, users must manage more extensive data sets. One of the primary purposes of the text management system is to streamline the process of finding and managing text and improve awareness and transparency. Thus, provide an overview of the entire data set and allow users to focus on details by refining and filtering out unrequired data, ultimately tailoring information to users.

Like file structures discussed in DR#2, categorisation systems must align with current practices to encourage familiarity, ultimately promoting learnability as it better aligns with users' mental models. Moreover, findability becomes equally imperative to ensure users can locate the correct texts among tens of thousands of text strings. The purpose is not for users to browse; users will have a specific goal when using the system. There are several different users and use cases for locating text. For example, factors including if the text is known or unknown, user roles and to what team they belong, and current stages of a development process all impact use cases. Thus, providing multiple ways of finding text accommodates the different use cases to ensure users can achieve their goals. Use cases include, but are not limited to, handling bug tickets or improvement tickets, reviewing text strings edited within a specific time frame, reviewing text strings belonging to a specific category, and updating the text to align with organisational decisions.

Design proposal

Flexibility is central in accommodating different use cases and needs, encompassing overall options and functionality within options such as search and filter. In this example, users can locate text through the local side navigation bar, by filtering table items, or through text search (Figure 7.15). Enhance users' feeling of control by providing a clear mapping between actions and effects. The filter options should mirror the information provided in the table. In this example, the table columns detail path, ID, text string, status, and last edited. Thus, users can filter information through text type, status, and last edited.

Users can view all existing text or select different levels of the hierarchy depending on the goal, such as locating all text within a page or objects and nested objects

within a page. Users can further refine their search by filtering all text or within the selected pages. Thus, the system must indicate what page or object the user has selected to promote visibility and ensure the users have enough information about their current actions.

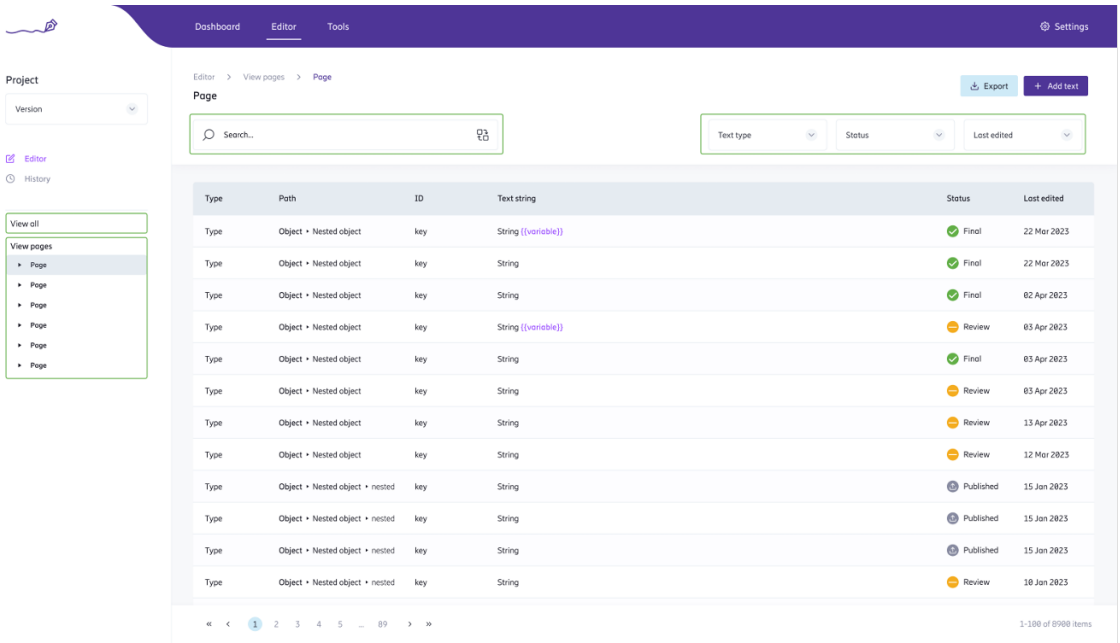


Figure 7.15: Multiple ways to locate text.

The search bar allows users to find specific strings. However, users should also be able to search paths, identifiers, error logs, terms, variables, and partial text strings. Flexibility in search is imperative to align with several different teams' needs. Furthermore, providing sufficient feedback is essential due to the number of possibilities in user actions. For example, indicate what filters are active (Figure 7.16) and what search term currently displays the results. Lastly, the table should be dynamically updated upon applying filters. However, consider designing filters that allow multiple options with an "apply" button to promote user control and reduce the risk of several reloads.

While the text management system is designed to utilise and mirror project JSON files, handling in-system data requires some considerations. The system must handle additional metadata for each text string not included in the exported files. For example, storing editing history, latest edits, status, and text type. JSON files can define the base categorisation system for path, ID, variables, and text strings. However, data such as status are user-generated, while the system automatically stores editing history. Furthermore, consider assigning each text string with a base status value to ensure all text strings can be located through filters.

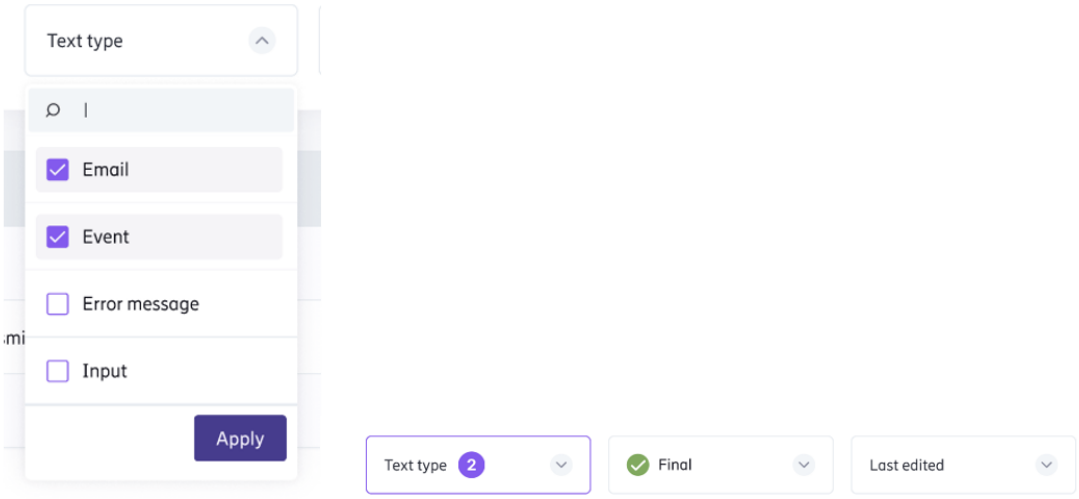


Figure 7.16: Active and inactive filters.

DR#8 Allow users to make universal changes

Requirements addressed

5.1. Users must be able to update terms simultaneously
9.3. The solution must support the use of variables

Rationale
While storing text in multiple tools forces users to manually make changes in multiple places, a "single source of truth" for text enables universal changes. This should be taken advantage of, especially in organisations maintaining systems with large amounts of data. Thus, users must be allowed to update text simultaneously throughout an entire system or in parts of the system. Ensuring that all items are updated accordingly will reinforce user control as it reduces oversights. In addition, universal changes will facilitate consistency in content, impacting user trust.

Similarly, universal changes can be through implementing templates and variables, discussed in DR#9. However, it should be implemented cautiously, removing the risks of users making problematic changes. Risks are especially high when text is directly pushed to the system as it is being updated or approved.

Design proposal
Include a standard text replacement feature. This recommendation is inspired by "find and replace" features available in Word, Figma and Crowdin. Place the feature in the main editor, allowing users to replace terms or text strings throughout the entire system. Furthermore, allow users to replace terms or text strings in parts of the system, such as a specific page, object, text type, etcetera. In this example, the search bar can be expanded and reveal the replace feature. Users type the text to be replaced and press the replace icon (Figure 7.17). Then, they can choose "replace all", referring to replacing all matched items according to the given path, whereas "replace" allow the user to specify constraints for replacement (Figure 7.18). Providing users with several possibilities when replacing text improves system flexibility and accommodates different needs and use cases. Provide the ability to undo changes (Figure 7.19), ensuring users can easily recover from potential mistakes.

Allowing users to make universal changes requires considering system feedback. It must be clear if changes are being made within parts of the system or the entire system. Furthermore, the user must know what parts of the system will implement the change by promoting recognition throughout the process. Replacing content throughout the entire system can be restricted based on roles and permissions if necessary.

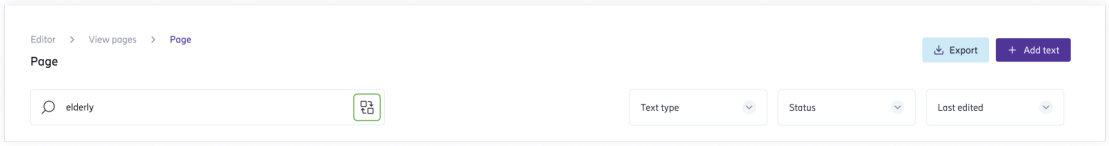


Figure 7.17: Find and replaced collapsed.

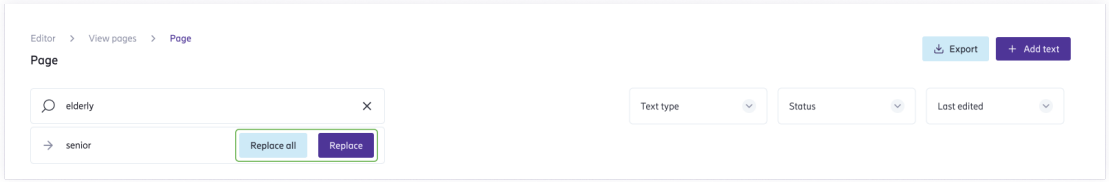


Figure 7.18: Find and replace expanded.

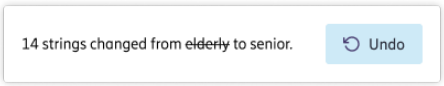


Figure 7.19: Undo text changes.

DR#9 Integrate guides within the editor

Requirements addressed

3.4. The solution should simplify sharing guidelines and style guides

Rationale

The rationale for implementing guides within the system is twofold. First, simplifying sharing guidelines requires that they are more transparent and accessible than current practices. Second, the inclination to use the guides must be addressed. Similar to how the tool operates as a "one source of truth" regarding text for NMP, the same should be applied to guidelines. Guidelines must be stored in the same place and structured to align with organisational goals and practices. However, the guides should also be integrated within the editor to enhance the inclination to follow guidelines and strive towards a shared understanding among contributors and consistency in content.

Design proposal

Store all relevant guides in a shared repository within the system (Figure 7.20), accessible to all users and editable by document owners. It must be clear what guidelines are integrated within the editor to allow users to get an overview and refer to them. Include a search functionality within the shared repository and provide context to the guidelines. Furthermore, allow flexibility in what guidelines to implement. Similar to page settings, provide the ability to structure, name, and group guidelines according to organisational needs.

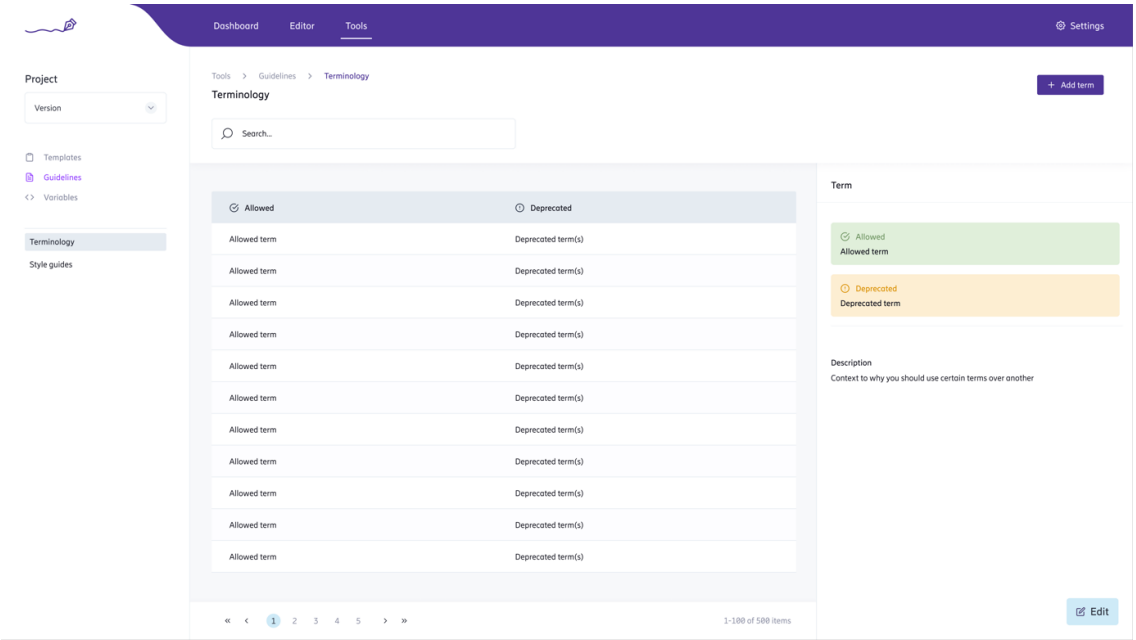


Figure 7.20: An example of a shared repository with allowed and deprecated terms.

Integrating guides within the editor removes the need for users to recall the information in the shared repository. Instead, inform the user of errors by direct feedback and help them diagnose and recover from the issue by offering context and a solution. Integrating guides within the editor requires feedback to ensure clear communication of actions' consequences. Thus, a term should only be automatically changed if it is accompanied by feedback and context as to why the action was performed. Provide feedback by emphasising the incorrect word and allow users to either confirm a suggestion (Figure 7.21) or provide clear feedback if it is changed automatically. Including direct access to the specific guideline's documentation promotes recognition in cases where users want additional information. Furthermore, consider incorporating constraints, hindering users from saving an edit containing deprecated or incorrect terms.

Figure 7.21: Editor detecting deprecated terms.

Implementation requires additional considerations regarding mapping the guidelines to the system. Enabling the system to identify deprecated terms and suggest a substitutional term demand documenting the pairs. However, implementation requires considering instances where users do not use a documented deprecated term but rather another term in place of the correct one. The system must be able to map and identify the intended term to provide feedback and suggestions. Similarly, if a term has been changed within an organisation, updating the term repository within the system should allow users to update any instance of the old term.

Furthermore, organisations generally have guidelines regarding tone of voice and formatting rules. Therefore, integrating writing rules and spelling alongside term identification provides the entire toolkit to ensure text strings are consistent and follow established guidelines. Nevertheless, there are several benefits in leveraging external writing assistants and providing services through third-party vendors, providing the ability to specify writing goals and tone of voice, etcetera.

DR#10 Provide users with reusable components

Requirements addressed

5.2. The solution must support reusability
9.3. The solution must support the use of variables

Rationale

Applications with a large amount of content commonly contain multiple instances containing the same or similar text. Thus, concentrate on reusing content when possible rather than relying on copying and pasting or rewriting. In cases where the same text occurs in different places, users should not be forced to write it several times. Instead, users should be able to set up components that can be reused in different parts of the system. The rationale is heavily influenced by component content management systems and components in programming.

Providing ways to create and import components containing reusable text snippets allow users to work time efficiently, reduces redundant work and improves overall consistency in content. Furthermore, content maintenance becomes more manageable as updating the component changes all instances. Similarly, code files contain variables defined and reused across an application. There must be support to define and insert variables within text strings to align a text management tool with the related application. Without variable indication and injection support, the exported files must be manually reviewed to correct variables.

Design proposal

Set up a library of reusable components aligning with established practices and goals and store them within the shared repository previously discussed in DR#9. This example includes templates, placeholders, and variables. All instances are consequently updated once an item within the library is updated. The feature works similarly to find and replace but utilises component logic.

Placeholders can be inserted if a text item is created prior to establishing the content or if other uncertainties arise. Templates are outlines of text items connected to specific contexts that can be reused wherever fits. Storing variables in the library, formatted correctly, allow users to insert and reuse the variable throughout the text files. However, it also aligns with the code; thus, converting it to external text files requires no extra work.

Differentiate variables from the static text as they serve different purposes, and it must be clear what a variable is for all users. Furthermore, use the same style to indicate variables throughout the text management tools to ensure consistency and improve learnability.

Ensure that actions are visible by directly incorporating access to templates, variables, and placeholders in the editor. Accommodate different users by providing flexibility in how to use the feature. In this example, users can add new items in the library or add them directly in the editor (Figure 7.22).

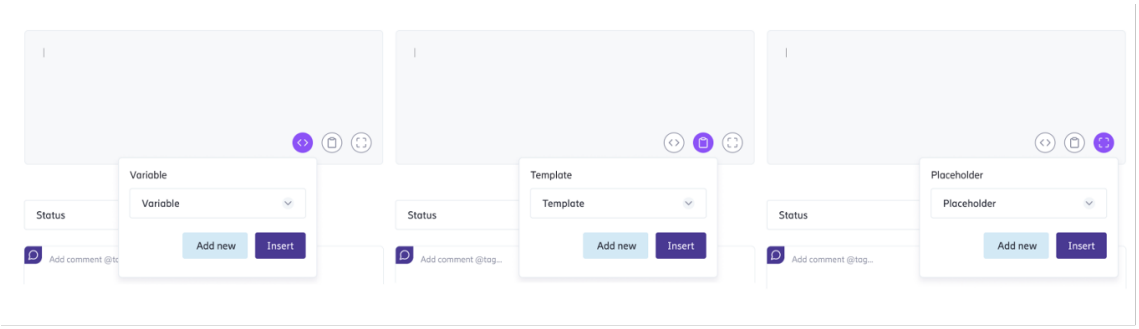


Figure 7.22: Add new items or insert into editor.

Variables can be inserted by pressing the icon to find and insert, removing the need to memorise and recall the variable name. However, the system should also detect a variable if written directly in the editor. Moreover, formats should align with established practices to promote familiarity and allow users to map the features within the text management tool to their processes and systems. In this example, variables are indicated by using double curly braces. When the system automatically detects and inserts a variable, it must be communicated to the user. Clear feedback is necessary so that users can understand whether the variable has been successfully inserted (Figure 7.23).

Templates should not be created without context. Instead, each template should be connected to existing elements and components to prevent several templates for the same purpose. While flexibility is important, it becomes a risk if templates are not utilised. Each feature of the text management system should be documented as part of the general text management process. Furthermore, templates should be integrated within the user flows to ensure they are intuitive to use and remove the need for users to recall what templates exist. Suggest relevant templates when users create new text. For example, if creating a new input, provide suggestions and quick access to templates such as validation messages.

Placeholders need unique identifiers to correctly update each instance when changing the placeholders in the tool repository. Furthermore, consider constraints and feedback to ensure text containing placeholders is not published in the final version. Lastly, utilise variables to align with current ways of working and user preferences. In this example, variable references are equivalent to the code. However, variables can have a name aligning with the code and be referenced as something else if the name is correctly formatted for export.

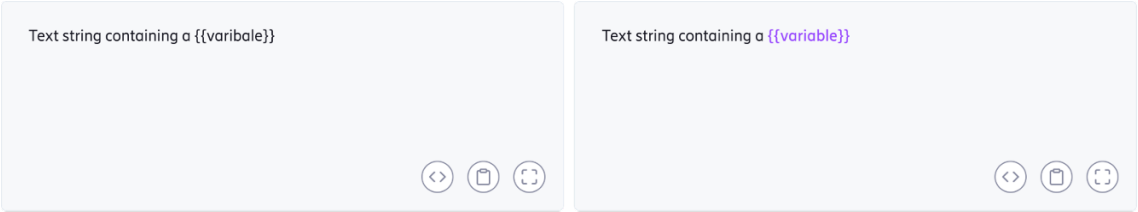


Figure 7.23: Variable indication without (left) and with (right) visual feedback.

DR#11 Continuously communicate editing history

Requirements addressed

4.1. Users must be able to track updates
4.4. Users must be able to see contributors

Rationale

Extensive communication to establish a shared understanding is imperative to cross-functional environments. Otherwise, information can easily get lost, causing confusion and redundant discussions. It is a significant concern regarding text management. Users must be able to stay current about what and why changes were made. However, it can be complicated when non-technical teams oversee text and are unsure whether their updates have been implemented, as they cannot access the code.

Thus, teams must clearly understand when a text has been updated and by whom to increase transparency. In addition, communicating contributors will improve the ease of communication as users will know whom to contact if questions arise, thus, removing uncertainty and avoiding redundant conversations. Editing history will increase awareness of the changes made and facilitate further potential actions to ensure quality before being published.

Design proposal

Promote flexibility and recognition over recall by providing history throughout the entire system. Implement editing history for each string in the editor (Figure 7.24) to enhance transparency and allow users to follow the development of individual items. Overall, history can include many items, making navigating and getting an overview difficult. Thus, users should be able to access text-specific editing history easily. There should also be a dedicated space for finding editing history for the entire project, where users can access a chronological overview of the latest updates concerning all activities in the project (Figure 7.25). Furthermore, promote visibility by providing the latest editing history at the system's landing site or dashboard so that users are immediately exposed to the latest actions, including updates and personal mentions.

Communicate detailed information, including contributors, updates and edits, and comments. Specifying contributors enhances control, as users know whom to contact if questions arise, facilitating collaboration. Furthermore, it enhances transparency by streamlining tracking responsibilities for a change. Second, communicating text updates and displaying the previous text ensures users can follow the evolution of text strings, making it easier to track potential errors and inform future decisions. Lastly, providing users with a commenting function that enables users to comment on specific changes will reduce cognitive workload as no additional context is needed. Include the possibility of tagging other users within the comment so they are informed when a specific task concerns them and are not required to locate the text. Furthermore, include filters so users can easily find the information they seek.

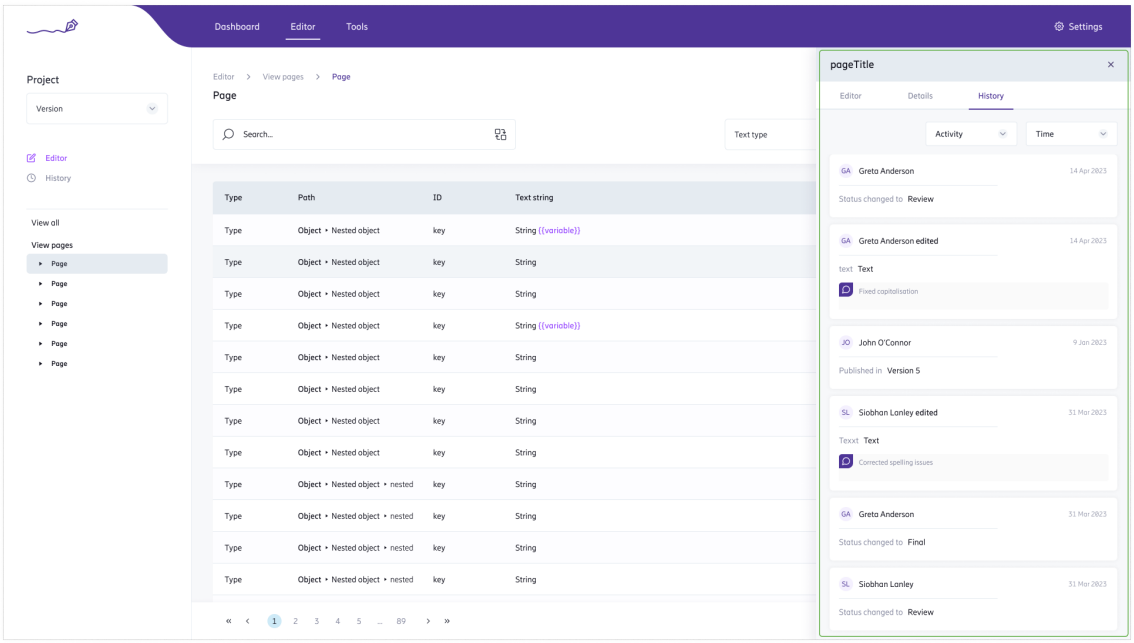


Figure 7.24: Text-specific history integrated into the editor.

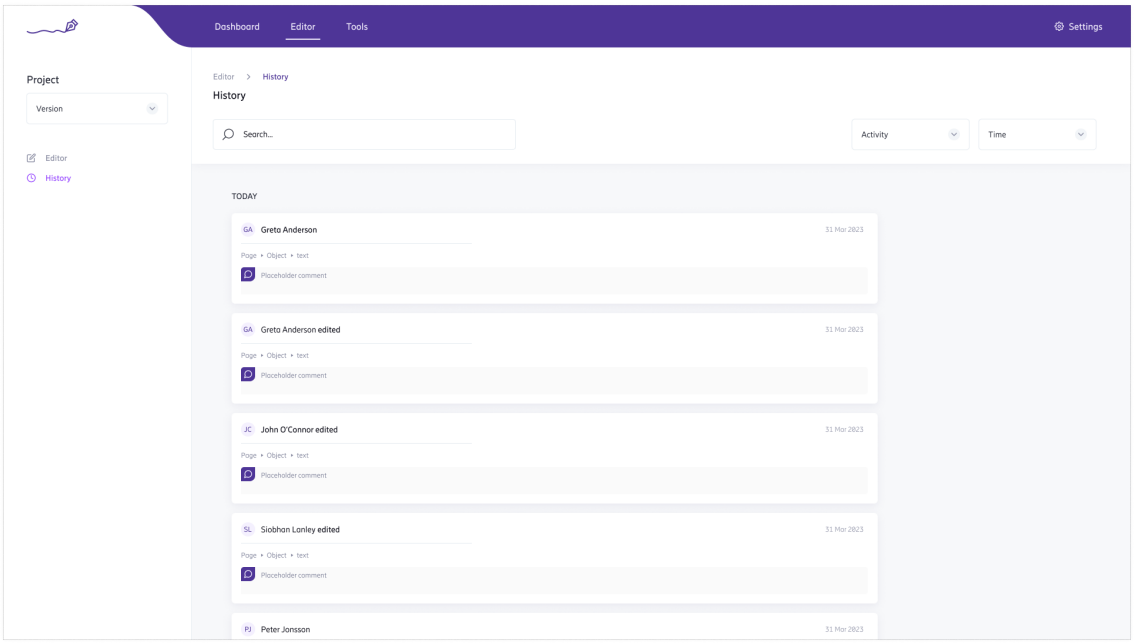


Figure 7.25: History page concerning all text strings.

DR#12 Allow users to communicate text-specific decisions and concerns

Requirements addressed

1.3. The solution must coincide with current communication tools
4.2. Users must be able to follow decisions
4.3. Users must be able to specify why a change was made

Rationale

Documentation is critical in a cross-functional environment to support decisions and promote a shared understanding within the organisation. Text management is no exception; thus, allowing users to acquire relevant documentation is imperative to a text management tool. However, organisations usually have a primary communication tool where decisions are made and discussions or questions arise. Thus, a text management tool should support such communication channels rather than forcing redundant documentation. The text management tool must coincide with external communication tools rather than replace or add an additional workload to established processes.

Design proposal

Allow users to add links to external tools, ensuring enough context and rationale for an update without manually writing it in a comment. Furthermore, it aligns the text management tool with external tools and allows users to access information and decisions easily. Encourage the use of such functionality by promoting recognition and providing shortcuts. In this example, links can be added to the external tools Slack and Jira (Figure 7.26). Furthermore, the icons differ from the internal shortcuts as they have different purposes.

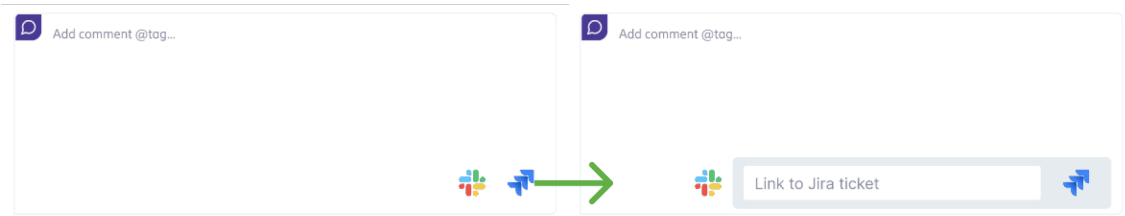


Figure 7.26: Insert links to Jira and Slack.

DR#13 Allow users to be assigned different roles

Requirements addressed

3.1. The solution must be accessible for all teams
3.2. The solution should facilitate an understanding of ownership
3.3. The solution must enable specifying roles and restricting access

Rationale

Designing for different roles and permissions is essential to align with current practices and reflect cross-functional responsibilities. For example, in large organisations, text may be owned and written by different parts of an organisation. Thus, assuring text quality can be challenging. Allowing users to define ownership and be assigned roles and permissions that reflect responsibilities reduces potential ambiguity. Regardless of established permissions and responsibilities, all teams must have access to the text management tool to enable users from different teams to contribute, ultimately facilitating collaboration across functions.

Design proposal

Dedicate a space within the system where permissions are managed (Figure 7.27). The needs may differ depending on organisational structures. However, in this example, all users with access to the system can be assigned roles and permissions.

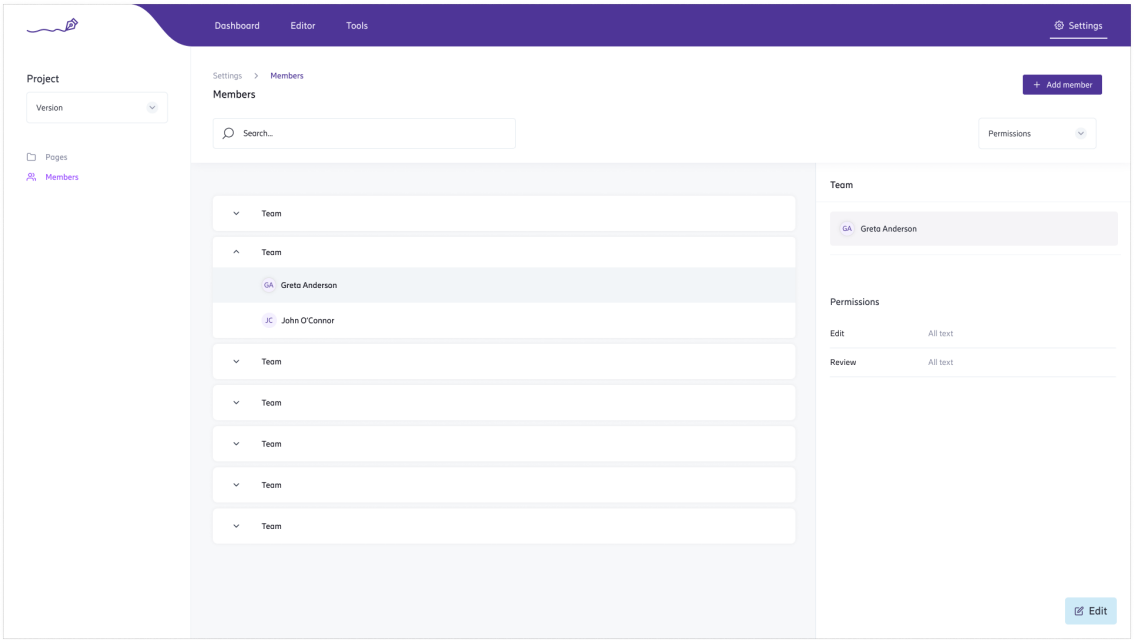


Figure 7.27: Team settings.

Permissions should be able to be defined both on an individual level and a team level. For example, if entire teams have specific responsibilities, they can easily be reflected by assigning team roles. However, if individuals have specific responsibilities, they must also be defined. Promote recognition to ensure that users are continuously

informed and reminded of established roles and responsibilities by providing the information for each text string within the editor.

Consider incorporating permissions such as edit, suggest, review, and read-only. While edit permissions are straightforward and essential to update text, the suggest permission can be an opportunity to promote collaboration by allowing people with different perspectives to give text suggestions, regardless of the editing permission. However, these actions must be easily distinguished to enhance user control, informing them that a suggestion does not equal updates.

Generally, specific teams are people responsible for the final review. Thus, review permissions should be considered and constrained accordingly to avoid accidentally publishing unfinished text. Lastly, providing read-only permission allow all teams to follow updates and decisions despite not being responsible for updating or editing. Furthermore, permissions can be designed in more detail, e.g., based on text types and context. For example, only people with responsibilities and access to code should be able to change the ID.

8

Discussion and Conclusions

This thesis was initiated on behalf of Ericsson and examined the possibilities of supporting text management processes to improve cross-functional collaboration. Much effort was put into the initial studies as previous research on challenges within text management is scarce, ensuring a rich understanding of issues related to current practices. The challenges were later addressed in a set of design recommendations for a text management tool, serving as an actionable plan to mitigate the identified challenges.

Paying closer attention to text is advantageous for organisations managing large products where user trust and understandability are crucial. It impacts usability and customer relationship more than may be apparent at first. For example, the NMP is a system for enterprises to manage and control networks and connected devices [21]. Thus, providing consistent and correct information at the correct time is imperative. Furthermore, the portal is the face of a complex infrastructure where users configure, manage, and monitor their networks; feeling secure and confident using it is essential.

Several teams at Ericsson manage text for the NMP. However, there is currently no intuitive way to align text management processes between teams, resulting in inconsistencies in the text, confusion about standards and potential updates. Accordingly, the lack of a shared space specifically for text management results in difficulties communicating cross-functionally, ultimately pointing toward a need for a joint solution specifically for text. Thus, this thesis identified challenges that impact cross-functional collaboration and text management to design a solution aiming to streamline text management processes and synchronise teams to improve cross-functional collaboration.

The following chapter provides discussions, conclusions, and insights regarding the entire project. First, the main findings are discussed in relation to the research questions and previous research. Then, discussions concerning the process are presented, highlighting specific areas of interest. Lastly, ethical considerations, generalisability, limitations and suggestions for future work are discussed.

8.1 Research questions

What are the main challenges with current approaches in a cross-functional organisation that affect teams' productivity and ability to synchronise their text management processes?

The results indicate that specific teams or specific actions are not causing friction; the challenges stem from issues encompassing the entire text management process. As a result, teams are both affecting and being affected by current processes. Thus, there are apparent issues regarding cross-functional collaboration. While this study differs from existing research by focusing on text management processes, several parallels can be drawn between the identified challenges and those discussed in previous research.

Podmajersky [13] emphasise that processes impact text quality more than anything else. Similarly, it became apparent that issues with text quality, consistency, and a shared understanding stem from current processes and lack of synchronisation between teams rather than a lack of knowledge regarding UX writing. Ultimately, these issues are increased by the lack of tools aligning with current needs and processes. The challenges regarding text management processes can be directly linked to the general challenges regarding cross-functional collaboration discussed in previous research. Participants described several challenges regarding unawareness, different perspectives, and lack of documentation. Literature indicates how not having a shared understanding affects how well teams can synchronise their work [9] [10] [11] [12]. Research further indicates that non-structured communication channels and decisions that are not adequately communicated between teams [23] hinder efficient collaboration [19]. Similarly, participants highlighted that how communication tools such as Jira and Slack are utilised is a problem rather than the tools themselves. Decisions are not documented consistently in the same place, resulting in a lack of concrete decision lists. A common theme throughout cross-functional collaboration research is the need for detailed documentation and reliable communication.

Consistency was identified as an apparent challenge and is similarly discussed in existing research. While research suggests adopting style guides can mitigate consistency issues [19] [10] [11], it is not necessarily sufficient. For example, participants stated they have several established practices and guidelines that are seldom referred to. Furthermore, they are not always commonly known, and only some are accessible to all teams. Similarly, while several parallels can be drawn between identified challenges and challenges regarding cross-functional collaboration, the results highlight that attempts at implementing such strategies have not resulted in any notable improvements. Thus, it became apparent that there must be more practical considerations for implementing solutions.

It is important to highlight that friction within cross-functional collaboration can vary depending on context and process. The results indicate that collaboration can work well in some aspects while lacking in others. While there are no definite answers as to why collaboration is tentative in specific contexts or processes, the

data suggests it can stem from the difference in requirements that products entail. For example, Ericsson usually designs for telecom experts rather than enterprise users. Thus, this change requires adopting the mindset of a different user rather than the traditional "for engineers by engineers" mindset.

Scattered processes, the use of various tools that do not fit current needs, and the lack of one source of truth were identified as the main factors impacting text management processes. Similar challenges are brought up in existing research. While there are existing tools aimed at designer and developer collaboration, these tools often lead to inefficiency rather than an improved process [17]. Leiva et al. [23] conclude that this is due to the different teams' tools operating in isolation, which can be directly connected to the concerns regarding current tools participants expressed during the interviews. In addition, using multiple tools for managing text is causing difficulties in tracking updates and versioning. While this thesis focuses solely on text, it supports previous research where issues related to auditability and tracking content changes arising from software discontinuity [28] were brought up and addressed by implementing a CCMS. However, Podmajersky [13] highlights that it is common practice for UX writers to use various tools throughout a process. Thus, it further strengthens the need to investigate requirements and explore solutions for a tool that can be used as a primary source of truth throughout the text management process, which ultimately is the goal of this thesis.

What design recommendations can be proposed for a text management tool to streamline text management processes and synchronise teams to improve cross-functional collaboration?

UX writing is a relatively new term, stemming from an increased awareness of how text impacts user experiences and customer relationships. Thus, the studies provide perspectives beyond managing content as technical writers. It places text management in a cross-functional context by providing design recommendations considering different practices and perspectives. The recommendations are presented as a strategy for managing text while encompassing user experience and developer implications.

CMS is the most common approach for managing content within an organisation. However, several factors within this study impacted the applicability of a CMS; it became apparent during the interviews, ideation, and evaluation that adopting a CMS cannot satisfy user needs. Thus, this study required building the recommendations from the data rather than applying the needs to existing tools. The identified needs point to a system that provides more than just updating and managing text. As the challenges proved to be complex, it became essential to establish requirements directly connected to the identified challenges. Establishing requirements provided a clear overview of what factors must be addressed to mitigate the experienced friction and how they should be prioritised to provide the most value. Prioritisation impacted the focus throughout design recommendations and was directly informed by user needs. While the studies address one case specifically, it raises awareness of how text management processes and available tools impact qual-

ity and productivity. Furthermore, it highlights additional needs that traditional content management systems inherently lack.

While the requirements answer how identified challenges should be addressed and mitigated, the recommendations provide further contextualisation. Therefore, several implications, factors, and considerations for implementation are also discussed to provide more context to the requirements and situate the findings into a more extensive process. However, requirements should ideally be coupled with success criteria, which were omitted from this project. Instead, the focus was on recommendations intended to be an actionable plan to address the requirements and mitigate the identified challenges. While the requirements were iteratively improved based on the studies and user needs, Ericsson best sets success criteria as they can further prioritise the requirements and define criteria based on their organisational constraints. The requirements are solely formulated by the data gathered throughout the studies and rationalised through technical validity and constraints. Furthermore, the study aimed to identify and understand the challenges, provide a strategy, and visualise recommendations through a semi-functional prototype. Thus, it was beyond the scope of this study to fully evaluate the requirements as that would require a working system.

The design recommendations are related to requirements that are exclusively related to all the data gathered throughout the project. Furthermore, empirical data has been accompanied by design principles and industry standards to form the recommendations. However, while the recommendations provide the necessary context and actions for solving problems related to text management in cross-functional environments, a potential implementation cannot be controlled. As stated in previous research, tools and processes are shaped by organisations, depending on their expertise and working methods [24]. However, interestingly, the findings share similarities with the previously discussed approach presented by Bexiga et al. [17] focusing on the designer-developer relationship and their collaboration. For example, the findings of this study reiterate the need for involving both designer and developer needs, maintaining shared repositories, streamlining design-to-code conventions and simplifying sharing guidelines.

Previous research highlights the importance of a shared understanding in cross-functional environments and contextualisation of content. However, this thesis builds upon previous research by considering practical implications throughout the design recommendations. For example, DR#9 builds on the need for establishing guidelines and proposes an integration of guides into a text management tool, promoting adherence. Moreover, the results support the fundamentals of component content management: reusable components. The results demonstrate the importance of reusability to improve user experiences, both in making processes more efficient and impacting consistency in content. Reusable components will decrease redundant work internally and ensure consistency in text, concluded as one of the main factors impacting user trust, decreasing cognitive load, and time to learn a system [11].

In line with the claims of Moreira [31], the results indicate the importance of providing context to text. Enough information must be provided when managing individual text strings to understand the text's context. While previous research focuses on semantics and metadata, the results of this study demonstrate an additional level of contextualisation necessary through illustrating links between associated texts. Furthermore, the results highlight that text management tools must store interface-related text and all forms of text generated by the system, such as emails. It helps organisations ensure that the correct information is provided so users can act on the information accordingly. Furthermore, it facilitates ensuring that information is consistent and not contradictory.

8.2 Process

A lot of time and effort was devoted to the empathise phase to understand user needs and behaviours, aligning with the design thinking approach described by Butler and Roberto [15], and Brenner and Uebernickel [14]. Involving users throughout the process enabled a deeper understanding of users and uncovering hidden needs. Likewise, participants expressed that their continuous involvement allowed them to realise and express problems and needs that were not directly evident.

The project is in collaboration with Ericsson. However, in addition, Ericsson continuously stayed involved and engaged. Thus, recruiting participants was never an issue as it was easy to get in contact with people across several teams, and people were happy to participate. The willingness to partake has benefited the breadth and depth of the results. The initial scope was to focus on the UX team's perspective alone and focus on other teams in relation to UX. However, throughout the empathy phase, it became apparent that FE is heavily involved and affected by current practices. Understanding their ways of working was, in turn, vital to address issues experienced by UX. Thus, the scope was reframed to include both UX and FE. While it is unsurprising that UX and FE have the most responsibility regarding text management, it is essential to note that other teams are still involved in the process and will be affected by potential changes.

While the competitor analysis was a great asset in finding inspiration and understanding what solutions currently exist, it would have been more beneficial to conduct it later in the process, once requirements had been refined and after some form of ideation. Thus, existing tools could have been analysed with a deeper understanding of user needs based on more data and include more aspects than the current framework. Nevertheless, involving users throughout all design thinking phases proved highly valuable in terms of understanding the problems, contextualising them into ideas and iteratively refining the requirements.

The design workshop was invaluable to the process as focusing on solutions resulted in a deeper understanding of the problems. While participants discussed and exemplified several perceived challenges during the interviews, expressing them fully in words can be challenging. Thus, the workshop confirmed previously defined requirements and raised awareness of problems not discussed during interviews. Allowing

participants to explore ideas and produce a means for communication provided a deeper insight and understanding of the problems uncovered through interviews. As a result, participants expressed that they could easier understand and express the problems they are currently experiencing. In turn, user needs were more thoroughly understood, resulting in requirements being refined and more detailed. Had users not been involved during ideation, several requirements and ideas would have gone unnoticed until potential evaluation or never uncovered.

While the design process only consisted of one evaluation, involving users during ideation and ensuring each idea was discussed and explored proved beneficial in establishing a solid foundation for creating the wireframes. It resulted in unnecessary problems or oversights being prevented during the evaluation. Instead, the evaluation uncovered issues that were not previously fully comprehended. Thus, the design walkthrough yielded valuable insights that largely impacted the design recommendations. While the evaluation was valuable, no usability testing was conducted for the entire interface. It is important to stress that the aim was to identify challenges and find solutions based on requirements rather than presenting a ready-to-deploy system. The semi-functional prototype is mainly an aid for visualising necessary components, features and implications.

Furthermore, while design workshops are flexible and can be organised in many ways, combining individual brainstorming, sketching, and group discussions worked very well concerning this project's aim. However, ideally, the session could have been longer. Nevertheless, the structured approach of KJ followed by sketching allowed participants to ideate according to their own experiences, needs, and priorities rather than a predefined scope. It further could have benefitted from an even narrower scope. While a more limited scope could have facilitated deeper and more extended discussions about each area or idea, it would also have reduced participants' opportunities to ideate freely and explore as many ideas as possible related to their perceived challenges. The discussions were still insightful, and a broader scope allowed participants more freedom during ideation.

Moreover, the discussion proved valuable for the study's results and for participants to understand each other's perspectives. It was apparent that participants from different teams had not discussed some of the experienced issues with each other and had different angles to the problems. In some cases, there were apprehensions about each other's ideas; however, the apprehensions stemmed from not fully understanding the needs of the other team. Thus, the discussion was important to the workshop as a whole and reiterated the importance of communication in cross-functional organisations.

8.3 Ethical considerations

Collecting empirical data implies obligations and ethical responsibilities. Among these are established consent forms designed to ensure transparency in research and inform participants about the study and their rights. Consent forms protect all parties; participants, researchers and organisations [40]. Thus, consent forms have

been used consistently in all studies involving users. Participants were provided with all information related to the session; research intent, plan, anonymity, withdrawal rights, potential recordings, how data will be handled, etcetera. Participants were also verbally briefed on the content of the consent forms, following established practices [68], to ensure that there were no misunderstandings. The consent forms were handed over physically at the beginning of each session, where participants were given time to read through them, ask questions, and sign them. A digital copy of the form was sent to each participant after the session. Sending the documents to the participants ensured they could access the information after the studies if needed.

Data must be handled in accordance with the General Data Protection Regulation (GDPR) [69]. Consequently, participants' anonymity must be preserved, and their identity should not be revealed [40]. Participants agreed to participate with anonymity preserved; thus, demographic data were omitted. All information connected to a participant's identity has been discarded and replaced with codes, e.g., P1, P2, and no unnecessary or sensitive personal data was collected. Furthermore, recordings were stored safely during transcription and deleted as the transcriptions were complete.

Research must be transparent. However, this study was conducted in collaboration with Ericsson. Thus, their privacy and researchers' obligations must be considered. Prior to the project, a non-disclosure agreement (NDA) was signed with Ericsson, ensuring confidential information remains protected. Thus, current practices have been described and detailed accordingly, and data was removed from the prototypes presented in the final thesis.

8.4 Generalisability

Simons [70] highlights that case studies do not make findings invalid; the generalisability and usefulness depend on how inferences are drawn from data. Furthermore, the author emphasises that the most important aspect is not necessarily generalising but showing that findings can be transferred to other contexts and how it relates to existing research.

Requirements were identified based on data regarding internal challenges within Ericsson Dedicated Networks. Thus, design recommendations were tailored to those specific requirements. Furthermore, while every person within the UX team was heavily involved, there are limitations regarding sample size. As a result, the design recommendations and requirements may not be generalisable over other organisations. However, the data obtained through user research has been supplemented with insights from current tools and industry standards to ensure the design recommendations can be applied outside of Ericsson rather than only tailored and valuable to them. Despite the need for more studies similar to this thesis, several of the findings can also be paralleled through existing research, which further strengthens the applicability in other organisations.

While this thesis addresses a specific case, the recommendations are designed to be both general directives and to be customised and adapted. The flexibility allows them to be adapted and implemented in other organisations as well. Presenting design recommendations instead of an entire interface without context facilitates external organisations applying them as it breaks down the solution into components and focuses on specific features, aspects, and flows. Thus, people can relate to specific problems they are facing and act accordingly.

However, to obtain a more generalisable result, similar studies must be carried out within other organisations with similar ways of working to Ericsson to explore the challenges. The identified problems in this thesis cannot be solved by implementing a CMS or through existing tools. The requirements and recommendations are not limited to telecom companies or only network management; the applicability within organisations depends on their current processes. The recommendations must be evaluated in different organisations to understand their applicability fully. While these studies considered cross-functional aspects such as expertise, different cultures were not explicitly addressed. Thus, conclusions regarding global applicability cannot be drawn. Evaluating the recommendations in other countries would be beneficial to understand how different cultures impact text management.

8.5 Limitations

It became apparent that no tool could be adapted and implemented to mitigate the identified challenges. However, the competitor analysis sample is only a small representation of currently available tools. Furthermore, only throughout this study have more tools appeared that are developed for teams to write and collaborate on text. However, they are primarily design-focused and do not fulfil all the requirements. Nevertheless, while they were explored as they were identified, they were not included in the competitor analysis.

As people become more aware of UX writing and the importance of copy, more and more tools will become available. However, it is vital to bring this subject to research before we see too many tools on the market to ensure that they are being developed in synergy with experienced challenges regarding text management. Similarly, this thesis is not concluding that organisations must develop new tools based on the presented recommendations. Instead, it presents a strategy and highlights essential aspects affecting process and quality. If a tool that adequately addresses the requirements emerges, that is viable to solve the identified challenges of this case. However, obtaining a generalisable result is complicated as all tools, regardless of foundation, must be adapted to the organisation in which it will operate.

Finally, the results are not conclusive as they have yet to be applied and tested in practice. Instead, the requirements and recommendations are intended to be a proposed strategy designed to impact the identified issues, streamline text management processes, synchronise teams, and improve cross-functional collaboration. Furthermore, the general scope is subjected to limitations as there are no notable works to compare to this study. Thus, this thesis aims to address the gaps in current research

and highlight challenges that stem from cross-functional organisations and increased awareness of UX writing.

8.6 Future work

Acknowledging the study's delimitations and limitations reveals directions for future research. The following steps regarding the specific findings of this study include implementing and evaluating the recommendations in practice. In addition, the recommendations should be applied and evaluated within different organisations to draw conclusions about generalisability and applicability. Future studies should consider the level of impact the recommendation has on current processes and how users are experiencing that impact. Furthermore, this study primarily focused on UX and FE and their links. Thus, future work includes more thoroughly addressing teams with other backgrounds.

Similar studies should be conducted in organisations experiencing similar issues to further add to a limited research area and uncover potential correlations. Accordingly, further research is needed to understand how different organisations manage text, accompanied by analyses of existing tools, to establish an understanding of how well current practices work.

Digitisation allows organisations to enter the global market. While this study was limited to monolingual text management, it aimed to lay a solid foundation for globalisation and localisation efforts. However, the general text management process must be addressed before integrating several languages. Thus, further research is needed to supplement the design recommendations regarding additional needs concerning globalisation and localisation. In addition, beyond pure translation, such studies must consider the interface implications of different languages and cultural and linguistic implications.

Bibliography

[1] Gregory Vial. "Understanding digital transformation: A review and a research agenda". In: *The Journal of Strategic Information Systems* 28.2 (2019), pp. 118–144.

[2] Anthony Bolton, Leila Goosen, and Elmarie Kritzinger. "Enterprise Digitization Enablement Through Unified Communication & Collaboration". In: *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*. Association for Computing Machinery, 2016, pp. 1–10. DOI: 10.1145/2987491.2987516.

[3] Prokhin Egor. "Digital Transformation of Industrial Companies: What is Management 4.0?" In: *2020 The 11th International Conference on E-business, Management and Economics*. Association for Computing Machinery, 2020, pp. 131–138. DOI: 10.1145/3414752.3414779.

[4] Marianna Di Gregorio, Giancarlo Nota, Marco Romano, Monica Sebillo, and Giuliana Vitiello. "Designing usable interfaces for the Industry 4.0". In: *Proceedings of the International Conference on Advanced Visual Interfaces*. Association for Computing Machinery, 2020, pp. 1–9. DOI: 10.1145/3399715.3399861.

[5] Rebekka Andersen and Tatiana Batova. "The Current State of Component Content Management: An Integrative Literature Review". In: *IEEE Transactions on Professional Communication* 58.3 (2015), pp. 247–270. DOI: 10.1109/TPC.2016.2516619.

[6] Rebekka Andersen and Tatiana Batova. "Introduction to the Special Issue: Content Management —Perspectives From the Trenches". In: *IEEE Transactions on Professional Communication* 58.3 (2015), pp. 242–246. DOI: 10.1109/TPC.2016.2521921.

[7] Joel H. Helquist, Amit Deokar, Thomas Meservy, and John Kruse. "Dynamic Collaboration: Participant-Driven Agile Processes for Complex Tasks". In: *SIGMIS Database* 42.2 (2011), pp. 95–115. DOI: 10.1145/1989098.1989104.

[8] Rashina Hoda, James Noble, and Stuart Marshall. "Balancing acts: walking the Agile tightrope". In: *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*. Association for Computing Machinery, 2010, pp. 5–12. DOI: 10.1145/1833310.1833312.

[9] Gilbert Cockton, Marta Lárusdóttir, Peggy Gregory, and Åsa Cajander. *Integrating User-Centred Design in Agile Development*. Springer Publishing, 2016.

[10] Suzanne Kieffer, Aissa Ghouti, and Benoit Macq. "The Agile UX Development Lifecycle: Combining Formative Usability and Agile Methods". In: *Proceedings of the 50th Hawaii International Conference on System Sciences*. Association for Computing Machinery, 2017, pp. 577–586. DOI: 10.24251/hicss.2017.070.

[11] Westley Knight. *UX for Developers: How to Integrate User-Centered Design Principles Into Your Day-To-Day Development Work*. Apress, 2018.

[12] Greg Nudelman. "Lean UX communication strategies for success in large organizations". In: *Interactions* 25.5 (2018), pp. 80–82. DOI: 10.1145/3236683.

[13] Torrey Podmajersky. *Strategic Writing for UX*. O'Reilly Media, Inc, 2019.

[14] Walter Brenner and Falk Uebernickel. *Design Thinking for Innovation: Research and Practice*. Springer International Publishing, 2016.

[15] Allison G. Butler and Michael A. Roberto. "When Cognition Interferes with Innovation: Overcoming Cognitive Obstacles to Design Thinking: Design thinking can fail when cognitive obstacles interfere; appropriate cognitive countermeasures can help disarm the traps". In: *Research-Technology Management* 61.4 (2018), pp. 45–51. DOI: 10.1080/08956308.2018.1471276.

[16] Julia Friberg and Therese Johansson. "Development of a Cooperation Framework for Cross-Functional Teams". MA thesis. Chalmers University of Technology, 2019.

[17] Mariana Bexiga, Stoyan Garbatov, and João Costa Seco. "Closing the gap between designers and developers in a low code ecosystem". In: *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. Association for Computing Machinery, 2020, pp. 1–10. DOI: 10.1145/3417990.3420195.

[18] Agnaldo O. Penha-Junior, Carlos E. De Souza, Guibson M. Souza, Jaqueline S. Araujo, José Ivan B. Vilarouca-Filho, Leon M. Barroso, Rodrigo J. B. Fernandes, Paulo Cesar R. Fonseca, and Felipe T. Giuntini. "Challenges in the Development of a Global Software User Interface by Multicultural Teams: An Industrial Experience". In: *2021 2nd Asia Service Sciences and Software*

Engineering Conference. Association for Computing Machinery, 2021, pp. 41–47.

[19] Helena Barke and Lutz Prechelt. "Some reasons why actual cross-fertilization in cross-functional agile teams is difficult". In: *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering*. Association for Computing Machinery, 2018, pp. 97–103. DOI: 10.1145/3195836.3195839.

[20] Michael Patton Quinn. "Expert sampling". In: *The SAGE Encyclopedia of Educational Research, Measurement, and Evaluation*. Ed. by Bruce B. Frey. SAGE Publications, Inc., 2018.

[21] Ericsson. *Ericsson Private 5G*. https://www.ericsson.com/494b57/assets/local/5g/documents/ericsson-private-5g-solution-brief.pdf. 2022.

[22] Ericsson. *Manage massive-data networks in real time*. https://www.ericsson.com/en/network-management. n.d. Retrieved 2023-02-07.

[23] Germán Leiva, Nolwenn Maudet, Wendy Mackay, and Michel Beaudouin-Lafon. "Enact: Reducing Designer & Developer Breakdowns When Prototyping Custom Interactions". In: *ACM Transactions on Computer-Human Interaction* 26.3 (2019), 19:1–19:48. DOI: 10.1145/3310276.

[24] Dimitra Chasanidou, Brian Elvesæter, and Arne-Jørgen Berre. "Enabling team collaboration with task management tools". In: *Proceedings of the 12th International Symposium on Open Collaboration*. Association for Computing Machinery, 2016, pp. 1–9. DOI: 10.1145/2957792.2957799.

[25] International Organization for Standardization. *Systems and software engineering — Content management for product life cycle, user and service management information for users*. ISO/IEC/IEEE 26531:2023(en). International Organization for Standardization, 2023.

[26] JoAnn T. Hackos. "International Standards for Information Development and Content Management". In: *IEEE Transactions on Professional Communication* 59.1 (2016), pp. 24–36. DOI: 10.1109/TPC.2016.2527278.

[27] Guiseppe Getto, Jack Labriola, and Sheryl Ruszkiewicz. "A Practitioner View of Content Strategy Best Practices in Technical Communication: A Meta-Analysis of the Literature". In: *Proceedings of the 37th ACM International Conference on the Design of Communication*. SIGDOC '19. Association for Computing Machinery, 2019. DOI: 10.1145/3328020.3353943.

[28] Grace Leinbach Coggio. "Technical Communicators as Agents and Adopters of Change: A Case Study of the Implementation of an Early Content-Management System". In: *IEEE Transactions on Professional Communication* 58.3 (2015), pp. 271–288. DOI: 10.1109/TPC.2016.2516618.

[29] Jacob E. McCarthy, Jeffrey T. Grabill, William Hart-Davidson, and Michael McLeod. "Content Management in the Workplace: Community, Context, and a New Way to Organize Writing". In: *Journal of Business and Technical Communication* 25.4 (2011), pp. 367–395. DOI: 10.1177/1050651911410943.

[30] Rahel Anne Bailie and Jeffrey Huset. "The Effect of CMS Technology on Writing Styles and Processes: Two Case Studies". In: *IEEE Transactions on Professional Communication* 58.3 (2015), pp. 309–327. DOI: 10.1109/TPC.2016.2516642.

[31] Cátia Moreira, João Taborda, Rosa Del Gaudio, Lara dos Santos, and Paulo Pereira. "Contextualizing Data on a Content Management System". In: *Proceedings of the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval*. Association for Computing Machinery, 2015, pp. 11–19. DOI: 10.1145/2810133.2810134.

[32] Tatiana Batova. "Component Content Management and Quality of Information Products for Global Audiences: An Integrative Literature Review". In: *IEEE Transactions on Professional Communication* 57.4 (2014), pp. 325–339. DOI: 10.1109/TPC.2014.2373911.

[33] Louis Weitzman, Sara Elo Dean, Dikran Meliksetian, Kapil Gupta, Nianjun Zhou, and Jessica Wu. "Transforming the content management process at IBM.com". In: *Case Studies of the CHI2002/AIGA Experience Design FORUM*. Association for Computing Machinery, 2002, pp. 1–15. DOI: 10.1145/507752.507757.

[34] Chris LaRoche and Brian Traynor. "Technical communication on life support: Content strategy and UX are the reclamation". In: *IEEE International Professonal Communication 2013 Conference*. 2013, pp. 1–6. DOI: 10.1109/IPCC.2013.6623941.

[35] Mariana Ribeiro de Azevedo, Fatima Manuela Boia, and Oscar Mealha. "Corporate relevance of UX Writing contributions for a Design Language System". In: *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*. Association for Computing Machinery, 2022, pp. 1–6. DOI: 10.23919/cisti54924.2022.9820407.

[36] Jennifer Schmich. *The problem with the UI text strings we write*. https://uxdesign.cc/working-out-a-workflow-for-ui-content-strings-7b283595e45a. 2019.

[37] The Style of Elements. *Several platforms, thousands of strings ... one writer?* https://thestyleofelements.org/several-platforms-thousands-of-strings-one-writer-8b3f09250674. 2016.

[38] Iddan Aaronsohn. *The Different Types of Content in Applications*. https://medium.com/flycode/the-different-types-of-content-in-applications-962c6a4c3bf7. 2022.

[39] Helen Sharp, Jennifer Preece, and Yvonne Rogers. *Interaction design: beyond human-computer interaction*. Wiley, 2019.

[40] Rex Hartson and Pardha Pyla. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Morgan Kaufmann, 2012.

[41] Phil Turner. *A Psychology of User Experience. Involvement, Affect and Aesthetics*. Springer International Publishing, 2017.

[42] Maria Rosala. *Crafting Product-Specific Design Principles to Support Better Decision Making*. https://www.nngroup.com/articles/design-principles/. 2020.

[43] David Benyon. *Designing User Experience: A guide to HCI, UX and interaction design*. Pearson Education Limited, 2019.

[44] Therese Fessenden. *Design Systems 101*. https://www.nngroup.com/articles/design-systems-101/. 2021.

[45] Brad Frost. *Atomic Design*. Brad Frost, 2016.

[46] Hasso Plattner. *An Introduction to Design Thinking, a Process Guide*. https://web.stanford.edu/~mshanks/MichaelShanks/files/509554.pdf. Institute of Design at Stanford, 2010.

[47] Bruce Hanington and Bella Martin. *Universal methods of design: 125 ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport Publishers, 2019.

[48] Richard Neves. *Competitor analysis*. http://www.pineeka.com/pages/researchmethods/ResearchMethods_CompetitorAnalysis.pdf. 2020.

[49] Vijay Kumar. *101 Design Methods: A Structured Approach for Driving Innovation in Your Organization*. Wiley, 2012.

[50] Kara Pernice. *Affinity Diagramming for Collaboratively Sorting UX Findings and Design Ideas*. https://www.nngroup.com/articles/affinity-diagram/. 2018.

[51] Sarah Gibbons. *Empathy Mapping: The First Step in Design Thinking*. https://www.nngroup.com/articles/empathy-mapping/. NN group, 2018.

[52] Interaction Design Foundation. *Making Your UX Life Easier with the MoSCoW*. https://www.interaction-design.org/literature/article/making-your-ux-life-easier-with-the-moscow. 2016.

[53] Kate Kaplan. *5 UX Workshops and When to Use Them: A Cheat Sheet*. https://www.nngroup.com/articles/5-ux-workshops/. 2019.

[54] Google. *Crazy 8's*. https://designsprintkit.withgoogle.com/methodology/phase3-sketch/crazy-8s. n.d. Retrieved 2023-05-16.

[55] William Buxton. *Sketching user experiences: getting the design right and the right design*. Elsevier/Morgan Kaufmann, 2007.

[56] Google. *Solution Sketch*. https://designsprintkit.withgoogle.com/methodology/phase3-sketch/solution-sketch. n.d. Retrieved 2023-05-16.

[57] Interaction Design Foundation. *Flowcharts*. https://www.interaction-design.org/literature/topics/flowcharts. n.d. Retrieved 2023-05-16.

[58] Alita Joyce. *Formative vs. Summative Evaluations*. https://www.nngroup.com/articles/formative-vs-summative-evaluations/. 2018.

[59] Runar Patel and Bo Davidsson. *Forskningsmetodikens grunder*. Studentlitteratur, 2019.

[60] Steve Mann. *The research interview: reflective practice and reflexivity in research processes*. Palgrave Macmillan, 2016.

[61] Lisa Given. *The SAGE Encyclopedia of Qualitative Research Methods*. SAGE Publications, Inc., 2008. DOI: 10.4135/9781412963909.

[62] TechTerms. *Localization Definition*. https://techterms.com/definition/localization. 2021.

[63] TechTerms. *CMS Definition*. https://techterms.com/definition/cms. 2013.

[64] TechTerms. *Digital Asset Definition*. https://techterms.com/definition/digital_asset. 2019.

[65] Yuhki Yamashita. *Introducing FigJam*. https://www.figma.com/blog/introducing-figjam/. 2021.

[66] Sarah Gibbons. *Cognitive Maps, Mind Maps, and Concept Maps: Definitions*. https://www.nngroup.com/articles/cognitive-mind-concept/. 2019.

[67] Tom Tullis and Bill Albert. *Measuring the user experience*. Elsevier Inc, 2013.

[68] Carol A. B. Warren. "Qualitative Interviewing". In: *Handbook of Interview Research*. Ed. by Jaber Gubrium and James Holstein. SAGE Publications, Inc., 2001.

[69] European Commission. *General Data Protection Regulation (EU GDPR) 2016/679*. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679. 2016.

[70] Helen Simons. *Case Study Research in Practice*. SAGE Publications, Ltd, 2009. DOI: 10.4135/9781446268322.

A

Project plan

Before beginning the project, a time plan was established in accordance with the design thinking process. Documentation and report writing was continuous throughout the process. The dates and methods of each phase are described below.

Thesis preparation
Jan 30th-Feb 16th

- Familiarise with Ericsson and the problem scope
- Literature review related to the problem scope
- Write and submit planning report

Empathise and define
Feb 13th-Mar 13th

- Prepare, conduct, transcribe, and analyse interviews
- Conduct competitor analysis
- Summarise findings and synthesise data

Ideate
Mar 13th-Mar 27th

- Prepare, conduct, and analyse workshop to inform internal ideation
- Define and prioritise requirements

Prototype and evaluate
Mar 27th-Apr 27th

- Create mindmaps to contextualise insights and inform further prototyping
- Initial sketching
- Create user flows to inform wireframes
- Conduct design walkthrough to evaluate wireframes

- Iterate design
- Finalise semi-functional prototype

Write up and finalise

Apr 27th-Jun 2nd

- Finalise thesis
- Presentation preparations
- Thesis defence and thesis opposition

B

Interview protocol

Reiterate the study's purpose and give participants the consent form to sign.

Introduction

- Can you please start by briefly describing your role at Ericsson.
- How long have you been working in that role?
- What are the *UX/FE/BE/CPI* responsibilities for NMP?

Cross-functional collaboration

- As a part of the *UX/FE/BE/CPI* team, what teams do you have the closest collaboration with?
  - How would you describe the relationship with these teams?
- In what situations do you usually interact with each other?
  - How frequently do you interact with other teams?
  - How do you usually interact with other teams?
  - What tools do you use for collaboration?
    - Can you name any pros and cons with those tools?
- Are there any other teams you collaborate with?
  - How would you describe the relationship with these teams?
  - Are there any consequences of not collaborating as closely?

Text management

- What types of text is your team involved in for NMP?
- Can you give us an example of how you work with text for the NMP?

- What tools do you use within the text management process?
  - Do you encounter any issues with your current process or tools?
- In what ways do you collaborate with other teams regarding text management?
  - What does the collaboration look like?
  - Where is collaboration necessary?
  - Can you give an example?
    - What happens if the collaboration is inadequate in these instances?

Scenarios

You notice there is an issue where a term in the NMP has been used incorrectly.

- How do you handle it?
- Is there a specific way to report issues?
- How do you inform others about the issue?

You are going to write a piece of text for the NMP, that will be visible to the users.

- How do you approach it?
- How do you know what terms to use?
- Are there any guidelines for text production?

C

Consent form

Interview about text management processes for NMP

Maja Ulén and Tara Lynch
Master's Thesis in Interaction Design and Technologies
Chalmers University of Technology

Study purpose
You are invited to participate in a research study for our master's thesis about text management for NMP. Your participation in this interview will help us understand how collaboration and text management work within and across Ericsson teams. In addition, the information will act as a basis for defining requirements for a text management tool.

Your participation is voluntary, and you can withdraw at any time during the study, including withdrawal of any information provided.

Information we will collect
The following material will be collected during the study for further analysis:

- Audio recording (if consented)
- Interview observations/notes

Privacy
All collected data will be treated as confidential and stored securely. Recordings and written notes will not contain any identifying information about you. All collected data will be anonymized. Only the researchers of the study will be granted access to the data for legitimate research purposes.

You are free to contact us if questions arise. You can also request to read the results before being published.

Consent

- I have read and understood the participant form
- I understand that my participation is voluntary and that I am free to withdraw at any time during the study without giving any reason
- I understand that this information will be published, with the understanding that anonymity will be preserved
- I agree to being recorded (audio only) during the study

Name _____

Signature _____

Date _____

If you want to withdraw your consent
If you want to withdraw your consent, contact one of us by email:

Tara Lynch: tara.lynch@ericsson.com
Maja Ulén: maja.ulen@ericsson.com

Researcher/witness
A copy of the signed and dated consent form and the participation information leaflet should be given to the participant. The researcher should also retain the original to be kept securely on file.

Name _____

Signature _____

Date _____

Name _____

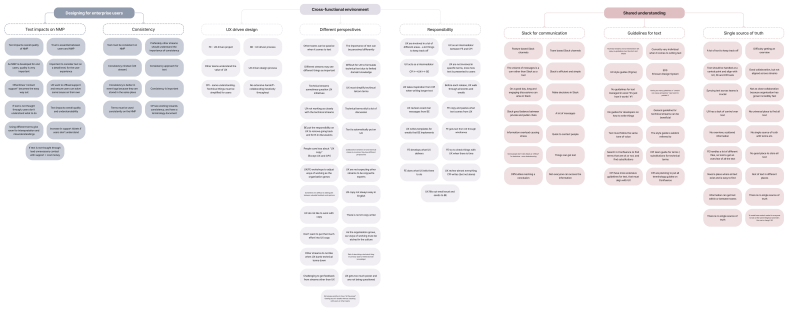
Signature _____

Date _____

D

Affinity diagrams

Interviews



Workshop