TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Rene Allkivi 203918IABM

A ROADMAP FOR THE RAPID MATURATION OF THE TEXT-DIALOG SYSTEM

Master's Thesis

Supervisor: Janek Metsallik
MSc

Co-supervisor: Kristian Juha Ismo Kankainen
MSc

Tallinn 2023

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Rene Allkivi 203918IABM

TEEKAART TEKST-DIALOOG SÜSTEEMI EDASIARENDUSEKS

Magistritöö

Juhendaja: Janek Metsallik
MSc

Kaasjuhendaja: Kristian Juha Ismo Kankainen
MSc

Tallinn 2023

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Rene Allkivi

10.05.2023

Abstract

The digital health sector has gained significant momentum in recent years, largely driven by the integration of information and communication technologies in healthcare delivery. One promising innovation in this domain is the Text-Dialog System, designed to convert clinical data in a written textual form into structured, machine-readable data using a controlled natural language based on the SNOMED Clinical Terminology. This thesis proposes a strategy for advancing the Text-Dialog System by assessing its current maturity level and identifying development priorities optimized for rapid maturation. The outcome of this research will provide a roadmap for the system's future growth, ensuring its successful integration into the healthcare industry and the realization of its full potential.

The findings of this research have the potential to impact researchers' strategies for technology maturation advancement, leading to improved technology development and better alignment with funding agencies' expectations. This could ultimately enhance the quality and efficiency of healthcare technology and benefit various stakeholders, including healthcare providers and policymakers.

The thesis is written in English and is 56 pages long, including 6 chapters, 9 figures and 4 tables.

Annotatsioon

Teekaart Tekst-Dialoog Süsteemi edasiarenduseks

Digitaalse tervise sektor on viimastel aastatel oluliselt hoogu kogunud, suuresti tänu info- ja kommunikatsioonitehnoloogiate integreerimisele tervishoiuteenuste pakkumises. Üks paljutõotav uuendus selles valdkonnas on Tekst-Dialoog Süsteem, mis on loodud kliiniliste andmete teisendamiseks kirjalikust tekstivormist struktureeritud, masinloetavaks andmeteks, kasutades kontrollitud loomulikku keelt, mis põhineb SNOMED kliinilisel terminoloogial. Käesolev töö pakub strateegiat Tekst-Dialog Süsteemi edasiseks arenduseks, hinnates selle praegust küpsustaset ja tuvastades arenguprioriteete. Uurimistöö pakub välja teekardi süsteemi tulevastele arendajatele.

Magistritöö on kirjutatud inglise keeles ning sisaldab teksti 56 leheküljel, 6 peatükki, 9 joonist, 4 tabelit.

List of Abbreviations and Terms

AI	Artificial Intelligence
DoD	U.S. Department of Defense
EAS	Enterprise Estonia
EC	European Commission
ESA	European Space Agency
EU	European Union
RE	Requirements engineering
R&D	Research & Development
SNOMED CT	Systematized Nomenclature of Medicine Clinical Terms
SWEBOK	Software Engineering Body of Knowledge
TRA	Technology Readiness Assessment
TRL	Technology Readiness Level
V&V	Verification and Validation

Table of Contents

List of Figures

List of Tables

1. Introduction

The advancements in information and communication technologies have led to significant transformations in various industries, including healthcare. Digital health, which refers to the use of these technologies to improve healthcare delivery, has gained momentum as a result of increased adoption and integration of these technologies in medical practices [1]. The European Union has shown a strong commitment to supporting innovation and development in this field through initiatives such as Horizon Europe, which aims to foster industry-driven research and development [2]. One promising development in the digital health sector is the Text-Dialog System, which is designed to capture clinical data in a written textual form and convert it into structured, machine-readable data using a controlled natural language based on the SNOMED Clinical Terminology [3]. This conversion is crucial in healthcare, as it enables more effective data management, analysis, and interoperability between systems, ultimately improving patient care and outcomes[4].

1.1 Objective

The objective of this thesis is to develop a strategy for the technological maturation of the Text-Dialog System being developed at Tallinn University of Technology. The system's current maturity and technological readiness are unknown, and no roadmap has been created for the continued development of the system. The goal of this thesis is to determine the current maturity level of the Text-Dialog System and develop a prioritized order of development areas optimized for swift maturation, ensuring the successful integration of this system into the healthcare industry and the realization of its full potential.

1.2 Expected outcome

The anticipated outcome of this research is a high-level roadmap that outlines the prioritized development areas for the Text-Dialog System, facilitating accelerated maturation and enabling researchers to effectively advance the system towards successful integration in the healthcare industry.

1.3 Thesis structure

Chapter 2: In this chapter, the Technology Readiness Level (TRL) scale is discussed, including its origins, applications, and variants.

Chapter 3: In this chapter, the role of Requirements Engineering (RE) in software development is covered, along with the RE process and the unique challenges of RE for the Digital Health sector.

Chapter 4: In this chapter, an overview of the current state of the Text-Dialog System is presented, including its high-level architecture, primary components, and their technology readiness levels (TRL).

Chapter 5: In this chapter, feasible development approaches for the Text-Dialog System are explored, featuring a SWOT analysis for each approach, and a development direction is recommended based on project resources and goals.

Chapter 6: In this chapter, alternative solutions and challenges related to the application of technology readiness levels in software maturity assessment and maturation are discussed.

Chapter 7: In this chapter, a summary of the key findings and insights gained from this research is provided.

The steps taken in chapters 4 and 5 are shown in Figure 1.

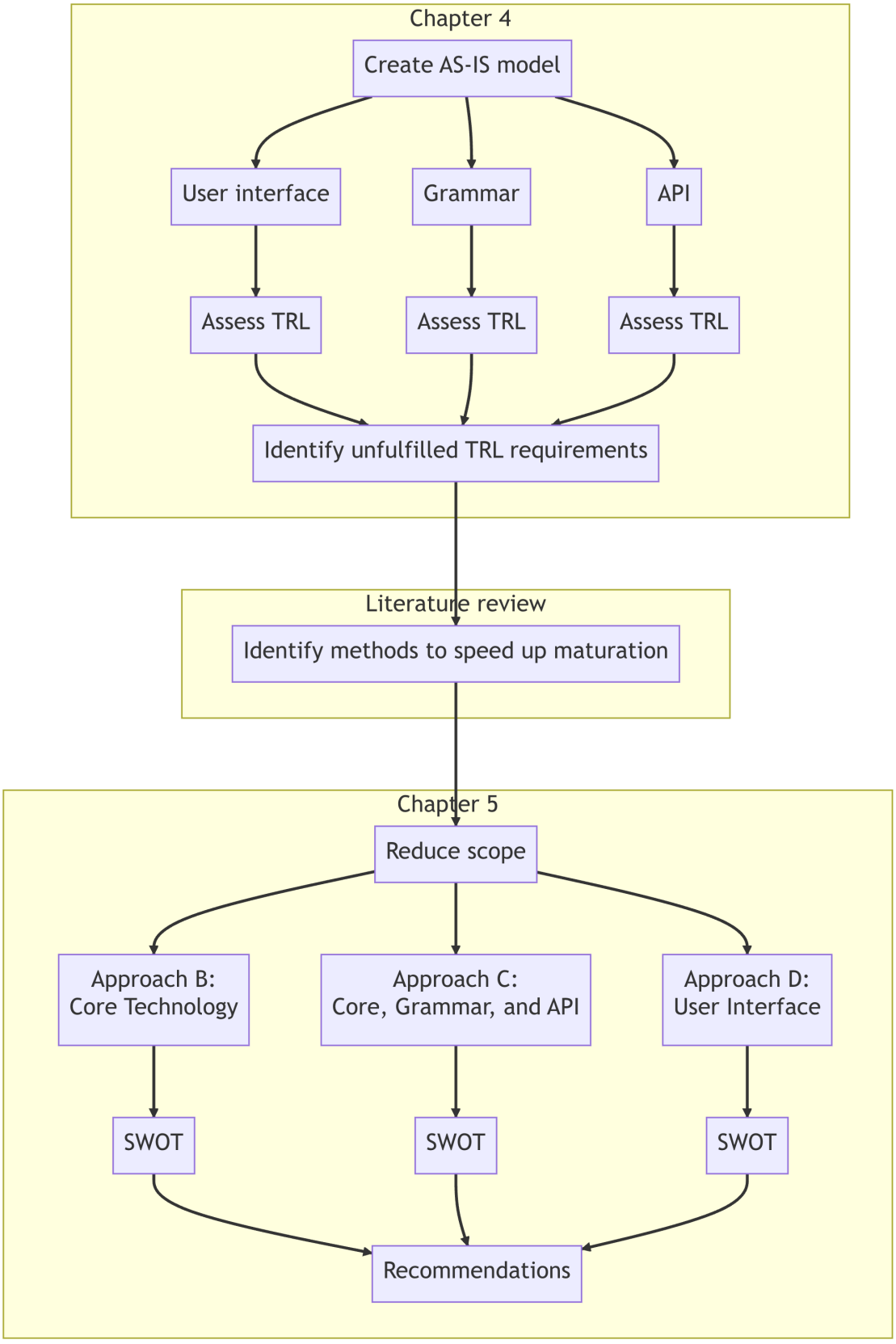

Figure 1. *Thesis process (chapters 4 and 5)*

2. Technology readiness levels

This chapter delves into the origins and applications of the Technology Readiness Level (TRL) scale, providing the foundation for selecting an appropriate TRL standard for Text-Dialog System, guidelines for assessing the TRL, and input for the discussion chapter to understand the limitations and shortcomings of the TRL framework. This chapter will also clarify that TRL is not a single framework, but rather a method and a collection of frameworks united by a common ancestor.

2.1 Origins of technology readiness levels

Efforts to incorporate emerging technologies into products before they reach maturity have been demonstrated to result in increased project costs and extended timelines [5]. This prompted NASA to develop the Technology Readiness Level (TRL) scale as a systematic tool for assessing and comparing the maturity of various technologies, thereby enabling more informed decisions and reducing the risks associated with premature technology integration.

The concept of "articulating" the maturity of a new technology for use in space technology was first introduced in 1969. The original TRL scale, consisting of six or seven levels, was developed by NASA's Office of Aeronautics and Space Technology in the late 1970s. After the Challenger Space Shuttle disaster, in 1986, NASA emphasized rebuilding its technological foundations through focused programs, leading to wider acceptance of the TRL scale. The scale was then extended to the now-standard nine levels and the definitive set of definitions of the technology readiness levels was published by NASA in 1995 [6]. See Table 1.

In the early 2000s, the U.S. Department of Defense (DoD) recognized the value of TRLs and began incorporating them into their own technology development and acquisition processes. This marked the first use of TRLs outside of the space industry. Following this adoption, the TRL scale gained widespread acceptance among various agencies within the DoD and their contractors [7].

Table 1. *Original technology readiness levels (1995) [6]*

Level	Summary
1	Basic principles observed and reported
2	Technology concept and/or application formulated
3	Analytical and experimental critical function and/or characteristic proof-of-concept
4	Component and/or breadboard validation in laboratory environment
5	Component and/or breadboard validation in relevant environment
6	System/subsystem model or prototype demonstration in a relevant environment (ground or space)
7	System prototype demonstration in a space environment
8	Actual system completed and “flight qualified” through test and demonstration (ground or space)
9	Actual system “flight proven” through successful mission operations

2.2 Adoption of the scale by European Space Agency

The TRL scale was introduced to Europe through its own space agency. In the mid-2000s, the European Space Agency (ESA) adopted the TRL scale, with a handbook [8] that closely mirrors NASA’s definition of TRLs, even reusing the thermometer-shaped TRL figure from NASA. Although the handbook does not include a strategic planning framework as sophisticated as NASA’s Systems Engineering Handbook and Technology Readiness Assessment best practices guide, it does provide a comprehensive description of each level, a workflow for technology readiness assessment, and additional guidance on applying the TRL scale to software development.

In 2013, the ISO 16290 standard, titled "Space systems – definition of the Technology readiness Levels (TRLs) and their criteria of assessment," was published. This standard shares similarities with the 1995 Mankins white paper, providing three examples for each of the nine TRL levels. It also resembles the DoD’s use of TRL by providing documented achievements for every level, which serve as the basis for TRL assessment [9].

At ESA, ISO 16290 replaced the agency’s internal document and became the official TRL scale definition. However, NASA continues to utilize its own TRL scale [2].

See Table 2) with differences to NASA standard in **bold**.

Table 2. *Technology readiness levels (ISO 16290) [9]*

Level	**Summary**
1	Basic principles observed and reported
2	Technology concept and/or application formulated
3	Analytical and experimental critical function and/or characteristic proof-of-concept
4	Component and/or breadboard validation in laboratory environment
5	Component and/or breadboard **critical function verification** in relevant environment
6	**Model demonstrating the critical functions of the element** in a relevant environment
7	**Model demonstrating the element performance for the operational environment**
8	**Actual system completed and accepted for flight**
9	Actual system “flight proven” through successful mission operations

2.3 Adoption of the scale by European Commission

The road to adopting TRL outside of space technology in the European Union (EU) was paved by the European Commission’s Communication 512, "Preparing for our future: Developing a common strategy for key enabling technologies in the EU" [2]. This work highlighted the importance of key enabling technologies (KET) for society and economy, and the need for policy makers to set up the framework conditions and support instruments for supporting the development of key enabling technologies in the EU. The five identified key enabling technologies were nanotechnology, micro- and nanoelectronics, photonics, advanced materials, and biotechnology [10].

To aid implementation of the KETs, the High-Level Expert Group on Key Enabling Technologies was founded and tasked with coming up with a strategy for strengthening Europe’s innovation capacity. This group presented the recommendation for EU to begin applying the "TRL scale R&D definition" [10].

This ignited the widespread use of TRLs in proposals for EU-funded projects. The Horizon 2020 program (total budget of 80 billion) included the definitions of the TRL levels, which

have subtle differences compared to the NASA levels [11]. However, the use of TRLs was not universal, with some programs making heavy use of TRLs, and others omitting technology levels entirely. The Horizon Europe program continues to make use of TRL in many work programs [12].

In Estonia, TRL has been used in some project calls funded by the European Commission, for example the Inter-sectoral mobility grant by the State Shared Service Centre [13] and Enterprise Estonia's applied research programme [14].

Table 3. *Technology readiness Levels (Horizon Europe) [12]*

Level	**Summary**
1	Basic principles observed
2	Technology concept formulated
3	Experimental proof of concept
4	Technology validated in a lab
5	Technology validated in a relevant environment (industrially relevant environment in the case of key enabling technologies)
6	Technology demonstrated in a relevant environment (industrially relevant environment in the case of key enabling technologies)
7	System prototype demonstration in an operational environment
8	System complete and qualified
9	Actual system proven in an operational environment (competitive manufacturing in the case of key enabling technologies, or in space)

2.4 Technology readiness levels for software

This section explores the application of TRLs specifically to software development since the process of creating and deploying software differs from traditional hardware or system development. Various organizations and programs have adapted the TRL scale to better suit the unique characteristics and requirements of software projects. The following tables and descriptions provide an overview of these adaptations and guidelines for assessing the maturity of software technologies at each level.

Table 4. *Technology readiness levels for software [15]*

Level	**Summary**
1	Scientific knowledge
2	Individual algorithms or functions are prototyped
3	Prototype of the main functionalities of the integrated system
4	preliminary release of not-mature software version, distributed to a community at an early stage of the software development life-cycle, that implements the main functionality of the software and by which preliminary V&V activities are achieved
5	preliminary release of not-mature software version, distributed to a community at an early stage of the software development life-cycle, that implements the complete functionality of the software and by which preliminary V&V activities are achieved
6	Ready for use in an operational or production context, including user support, as a building block or a tool.
7	Building block and tailored generic software product: qualified for a particular purpose Tool: ready for market deployment
8	System qualified and ready to be applied in the execution of a real space mission
9	Has been applied in the execution of a real space mission

Enterprise Estonia has published a more comprehensive set of level criteria specifically for IT projects as part of its European Commission funded programme for applied research [14].

The **ESA TRL handbook** includes a dedicated set of guidelines (Table 4) for software technology readiness levels, dividing space software into three categories of

1. Software building block to be reused in a range of missions, either flight or ground software. This software is executed in a wider software application context. It interacts with other software and also with HW
2. Software tools. They run in a stand-alone mode
3. Software that cannot be considered separated from the HW it runs on, e.g. equipment embedded software.

The handbook also documents the following software engineering terms relevant to each level [15]:

TRL1 Mathematical formulation
TRL2 Algorithm
TRL3 Prototype
TRL4 Alpha version
TRL5 Beta version
TRL6 Product release
TRL7 Early adopter version
TRL8 General product
TRL9 Live product

The regular set of TRLs is recommended for the third type, but an alternative approach is proposed for the first two. TRL1-4 are used to go from mathematical formulation to prototyping and an eventual "alpha" version. TRL5-6 take the software from "alpha" version to a released product. TRL7-9 involve integration into the spacecraft and eventually launch into space [15].

2.5 Technology readiness assessment

Understanding who should be involved in the Technology Readiness Assessment (TRA) and the process of conducting the assessment is crucial for an accurate and effective evaluation of a technology's maturity. TRAs should "follow a disciplined and repeatable process", concentrating on the end user's intended application of the technology, and utilize evidence. The quality of a TRA is reliant upon strong collaboration and communication among all involved parties, such as the technology developer, program manager, governance body, and other team members participating in the assessment [16].

In the United States, the U.S. Government Accountability Office has published a Technology Readiness Assessment Guide (more than 100 pages) containing thorough instructions for conducting a TRA. According to this guide, a TRA should be credible (requires an understanding of the technology's requirements), objective (based on trustworthy information), reliable (requires following a disciplined process that also ensures repeatability) , and useful (requires an understanding of the report's target audience). The five-step process is described in detail, starting from assembly of the TRA team and ending with suggestions on how to use the results [16].

The TRL ISO standard does not contain instructions for conducting a TRA, but does

include a set of documented work achievement for each level. The European Cooperation for Space Standardization has published TRL guidelines that act as a supplement to the ISO standard. These guidelines include an overview of a "typical technology readiness assessment process" and require a TRA to be completed by an independent expert not part of the technology developer engineering team who is required to have access to the necessary information and data concerning the technology and the level to be assessed [15].

The European Commission (EC) has not specified any formal process or requirements for conducting a TRL assessment, leaving the actual implementation details and guidelines to the individual agencies organising funding calls. This can probably be attributed to the sheer diversity of technologies that are being funded by the EC.

2.6 Technology readiness is not commercial readiness

Reaching a higher TRL does not necessarily guarantee a successful product. Although TRL 9 indicates that the system is proven in its operational environment, it does not account for factors such as branding, cost, or competition, which are essential for market readiness., which are essential for market readiness. In Australia, the Commercial Readiness Index (CRI) was created to address these risks, measuring the maturity of a technology based on its financial deployment arrangements [2].

3. Requirements engineering

This chapter delves into the critical role of Requirements Engineering (RE) in the development of software. The stages and activities of the RE process are described, and the unique challenges of RE for Digital Health sector are discussed. Inadequate requirements were the primary reason for the failure of software projects. [17]

This overview will be used in later chapters to describe the appropriate development activities for the Text-Dialog System.

Sommerville and Sawyer have defined it well: "Requirements are defined during the early stages of a system development as a specification of what should be implemented. They are descriptions of how the system should behave, or of a system property or attribute. They may be a constraint on the development process of the system." [18]

Requirements engineering is the process of defining, documenting, and maintaining the requirements for a system or product. It is a critical phase in the development process, as it helps to ensure that the final product or system is valuable for its owner.

According to Robertson, "requirements exist whether you discover them or not, whether you write them down or not". The focus of requirements engineering is not on creating a list of requirements but on understanding and addressing a business problem. Identifying the true problem is the key to finding the best solution. The term "business" in this context can encompass various domains, including commercial, scientific, government, military, or any other type of activity or service. [19]

Requirements engineering continues to be one of the most-discussed topics in software engineering. [20]

Requirements engineering can be split into requirements development and requirements management. **Requirements development** is the process of creating, refining, and documenting the requirements for a system or product, while **requirements management** deals with maintaining and controlling changes to the requirements throughout the project life cycle [21].

3.1 Requirements development process

The Software Engineering Body of Knowledge (SWEBOK) subdivides requirements development into requirements elicitation, requirements analysis, requirements specification, and requirements validation [22]. Some authors further distinguish additional phases such as requirement planning, and requirement negotiation [1].

Elicitation is the process of gathering information about the requirements from various sources, such as stakeholders, users, customers, domain experts, and existing documentation. The goal is to identify the needs and expectations of the system or product. Elicitation techniques may include interviews, workshops, questionnaires, observations, and analysis of existing systems or documentation [22].

Analysis and planning phase involves organizing, structuring, and prioritizing the gathered information. It helps identify inconsistencies, conflicts, gaps, and redundancies in the requirements. Techniques used in this phase may include modeling, use case analysis, and prototyping. The output of this phase is a refined set of requirements that are clear, consistent, and complete [22][1].

Specification phase involves documenting the requirements in a structured and formal manner. The requirements specification should be clear, concise, and unambiguous, making it easier for stakeholders to review and provide feedback. Common types of documents include Software Requirements Specification (SRS), use case documents, or user stories [22].

Validation and negotiation ensures that the documented requirements are accurate, complete, and consistent before actual development starts. It involves reviewing the requirements with stakeholders, customers, and users to obtain their feedback and confirm that the requirements meet their expectations. Validation techniques include walkthroughs, inspections, or formal reviews. This phase helps identify and resolve any issues or misunderstandings before the requirements are used in subsequent phases of the project [22][1].

Require developments techniques involve several methods for identifying stakeholders, and eliciting, analyzing, specifying, and validating requirements. A lot of research has been done to identify and document the available techniques [23][21][22][19] and their effectiveness for achieving product and project success [17][24][1].

3.2 Requirements development for digital health

Requirements engineering practices can vary significantly across different fields due to the unique characteristics and constraints of each domain. Several factors can increase the need for diligent requirements engineering in a project. Digital health has many factors affecting the requirements of its products, including, but not limited to

- **Patient safety**: Ensuring patient safety is a top priority in healthcare product development. It is essential to identify and mitigate potential risks, perform thorough testing, and implement fail-safe mechanisms to prevent adverse effects on patients.
- **Data security and privacy**: Digital health products often deal with sensitive patient data, making data security and privacy a vital consideration [25].
- **Regulatory compliance**: Digital health products must adhere to strict regulations set by governing bodies, such as the U.S. Food and Drug Administration (FDA) or the European Medicines Agency (EMA). Compliance with these regulations is crucial to obtain necessary approvals and certifications for market entry [26].
- **Interoperability**: Healthcare systems often need to communicate and share data with other systems or devices. Ensuring interoperability and compatibility with existing systems, industry standards, and protocols is essential for seamless integration and data exchange [4].
- **Ethical considerations**: Healthcare products should be developed and used ethically, respecting patient autonomy, beneficence, non-maleficence, and justice. Consider conducting ethical reviews and consulting with ethics committees to address potential ethical concerns during product development [27].

Stakeholders in Digital Health can be divided into the following four groups [28]. Stakeholders from all four groups must be identified and prioritised.

- Entities accepting care
- Entities providing care
- Supporting entities
- Controlling entities

In the context of digital health solutions, the relationship between stakeholders is not merely a simple two-sided market, and the risks involved are far greater than in standard consumer-facing projects. In fact, an inadequately designed, developed, or implemented digital health solution may not only result in financial losses, wasted time, and inconvenience but also jeopardize lives and inflict physical and psychological harm. Digital health presents unique challenges, such as mismatched technology and use case, a long list of potentially

conflicting stakeholders. Doctors and nurses are often over-represented in requirements engineering activities, while patients and other users and stakeholders receive less attention [28].

The **business case** for digital health solutions is often complex, extending beyond typical provider-consumer relationships. These services are frequently offered for free as part of a larger care concept by non-profit medical organizations, but they may also involve for-profit providers like insurance companies and technology providers. To develop successful digital health solutions, it's crucial to understand the specific context and socio-technological alignment. A **multidisciplinary team with a deep understanding of the technology and context** should work together to apply a human-centered, well-organized, flexible, and creative process in developing the requirements [28].

Development process

Volk et al. propose a tailored process for the requirements development of Digital Health projects, emphasizing the importance of a project-specific RE approach rather than relying solely on research conducted independently of product development [28]. The process consists of the following stages:

1. **Preparation:**
   (a) Team identification: Assembling a multidisciplinary RE team to ensure diverse perspectives and expertise.
   (b) RE Plan Definition: Developing a high-level description of the targeted digital health system, discovering crucial technologies, and preparing architecture descriptions, following international standards and recommendations.
2. **Elaboration:**
   (a) Stakeholder identification: Identifying and profiling relevant stakeholders to better understand their needs and expectations.
   (b) Initial inquiry: Confirming and detailing the vision and system context, including security, privacy, and data handling considerations in line with national and EU regulations.
   (c) System context discovery: Investigating the standardization and certification landscape for the digital health system, with a focus on healthcare systems and medical devices.
3. **Increments and Iterations:** Planning the RE process iteratively and incrementally, with at least two cycles of design, prototyping, and evaluations. The proposed hybrid process model combines an in-depth, comprehensive RE phase at the beginning to establish a deep understanding of the digital health context, followed by continuous

RE iterations throughout the system design and development phases to maintain agility and adapt to project changes. Each cycle includes the following activities:

- Establishment of the vision and system context
- Stakeholder identification and profiling
- Requirements inquiry, analysis, and prototyping
- Vision, context, and requirements documentation
- Requirements validation, negotiation, and refinement

This approach is distinct from the scope of requirements engineering in SWEBOK, as it includes elements such as team identification, RE plan definition, and stakeholder identification. Volk et al. also recommend the use of *observation as a requirements elicitation technique*, which is particularly useful in the varied environments of healthcare workers, such as hospitals, offices, homes, and ambulances [28].

3.3 Requirements development for early-stage technologies

Requirements play a crucial role in both research-oriented and product-oriented development, but their approach and focus can differ significantly between the two. Research-oriented developments often emphasize exploring new ideas, developing innovative concepts, and assessing the feasibility of potential solutions. In this context, requirements tend to be broader, more experimental, and less defined [29]. In contrast, product-oriented technologies focus on addressing specific market needs or solving well-defined problems. Requirements for these projects are more concrete, well-specified, and directly related to the product's functionality, performance, and quality [30].

In research-oriented projects, requirements engineering tends to be more flexible and iterative, with requirements changing frequently as new insights are gained and hypotheses are tested [29]. However, in product-oriented projects, requirements are generally more stable and well-defined, as they are based on a clear understanding of the market, user needs, and business goals. Changes to requirements at this stage can significantly impact the product development timeline and associated costs [1].

Requirements in research-oriented projects may be derived from theoretical foundations, prior research, experimental data, or expert opinions. Validation of these requirements may involve testing hypotheses, conducting experiments, or building prototypes [29]. On the other hand, product-oriented projects typically gather requirements through market research, user interviews, surveys, and other customer-focused activities. Validation of requirements in this scenario involves ensuring alignment with customer needs, meeting regulatory requirements, and assessing feasibility within the project's constraints [1][30].

4. Current state of Text-Dialog System

This chapter presents an overview of the current state of the Text-Dialog System, its high-level architecture, primary components, and their technology readiness levels (TRL)

Text-Dialog System is designed to capture clinical data in a written textual form and convert it into structured, machine-readable data using a controlled natural language based on the SNOMED Clinical Terminology [3].

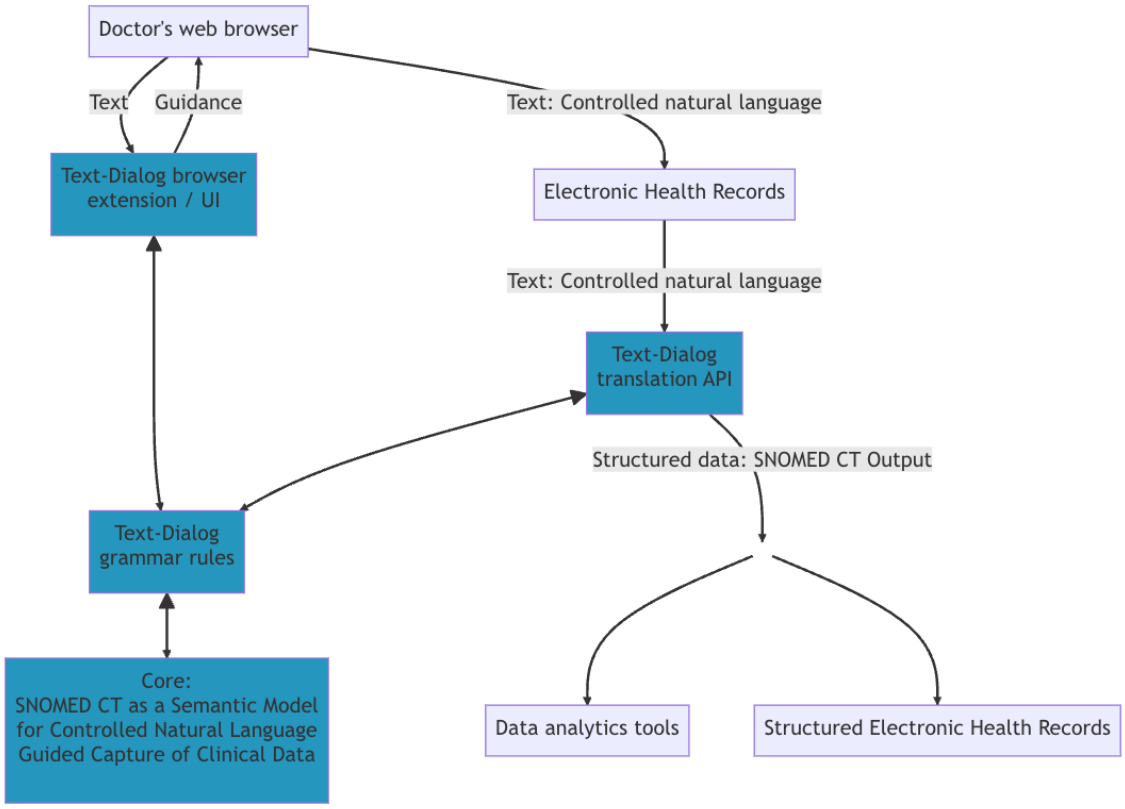

Figure 2. *High-level model of Text-Dialog System*

4.1 Description and model

Text-Dialog System is a system that facilitates bidirectional conversion between unstructured textual data and formal representations using SNOMED Clinical Terminology (SNOMED CT) [3].

The technology can be divided into two components: the user interface and the actual grammar of the Controlled Natural Language (CNL). The user interface is responsible

for facilitating interaction between users and the system, ensuring the text users are creating is compliant with the grammar rules of the CNL. The CNL's grammar defines the linguistic rules and structures that enable the formation and comprehension of meaningful expressions in the language [3].

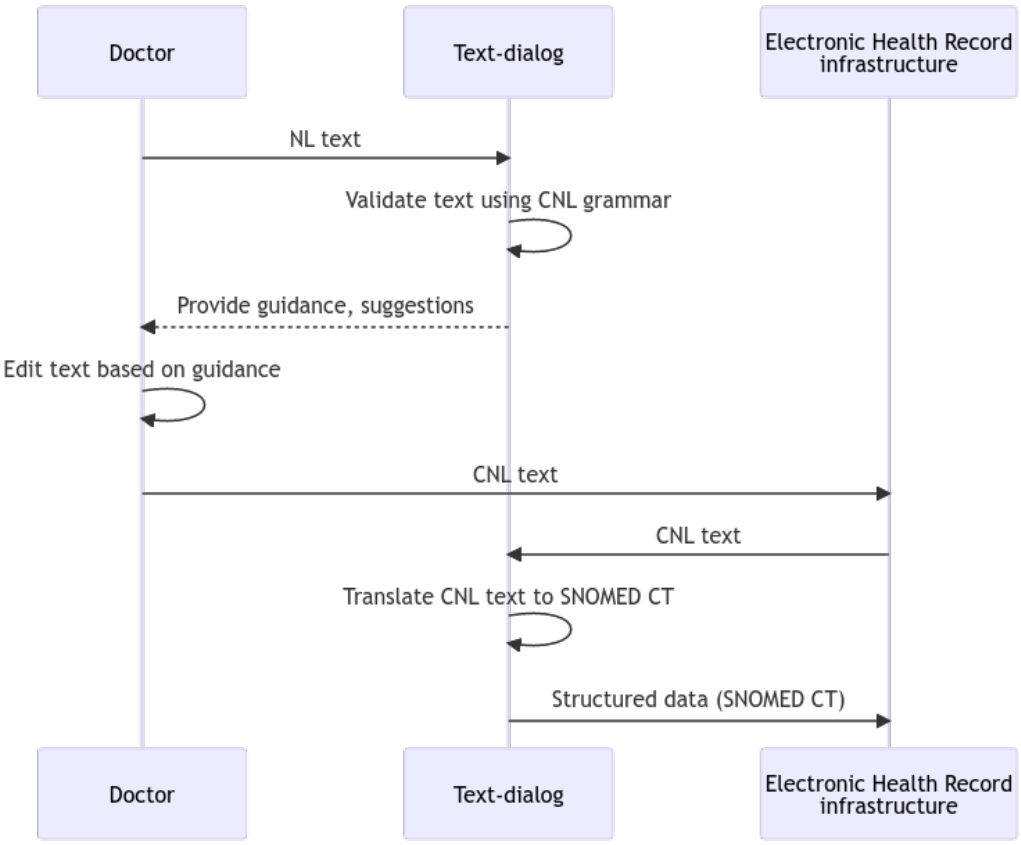

Figure 3. *High-level sequence diagram of Text-Dialog System*

While multiple articles concerning the development and viability of the technology have been published [3, 31, 32], none of them have explained the vision for the system architecture. In this assessment, the descriptions and details of the Text-Dialog System components and their current state of development are based on an extensive interview conducted with the leading researcher, Kristian Kankainen, who has provided valuable insights into the project's progress and challenges.

The three identified modules are as follows, and their relationship can be seen in Figure 2. The parts of the Text-Dialog System are shown with a darker background, other relevant modules and participants are shown in a lighter color. A sequence diagram of the main flows can be seen in Figure 3

4.1.1 Grammar

The **grammar** component of the Controlled Natural Language (CNL) for Guided Capture of Clinical Data serves as the foundation for constructing and understanding meaningful expressions within the language. This critical aspect of the CNL system consists of a set of linguistic rules and structures that determine the formation of syntactically and semantically valid sentences. The grammar encompasses various language constructs, such as morphological, syntactic, and semantic elements, which enable the generation and interpretation of complex expressions. By adhering to the CNL's grammar, users can create clear, unambiguous statements that effectively convey their intended meaning and are translatable to a structured data format employing the SNOMED Clinical Terminology.

The overall goal in the development of the grammar is to create a set of rules that has a high coverage for supported terminology. A higher coverage signifies that the grammar can handle a wide range of concepts, leading to better usability in diverse clinical contexts. To achieve this balance of high coverage and intuitiveness, the development process should involve collaboration between computational linguists, subject matter experts such as clinicians, and software developers. This interdisciplinary approach ensures that the grammar captures the nuances of clinical language while remaining easy to use and interpret.

At the same time, the created grammar must be intuitive for its users, which include clinicians, medical professionals, and other healthcare staff. The grammar should be easy to learn and understand, reducing the cognitive burden and allowing users to focus on expressing their clinical observations and decisions effectively. Using familiar terminology and structures that closely resemble natural language can make the CNL more accessible and user-friendly.

To achieve this balance of high coverage and intuitiveness, the development process should involve collaboration between computational linguists, subject matter experts such as clinicians, and software developers. This interdisciplinary approach ensures that the grammar captures the nuances of clinical language while remaining easy to use and interpret.

4.1.2 User interface

The user interface component of the Controlled Natural Language is a critical aspect as it facilitates the interactions between users and the system. The researchers have built two

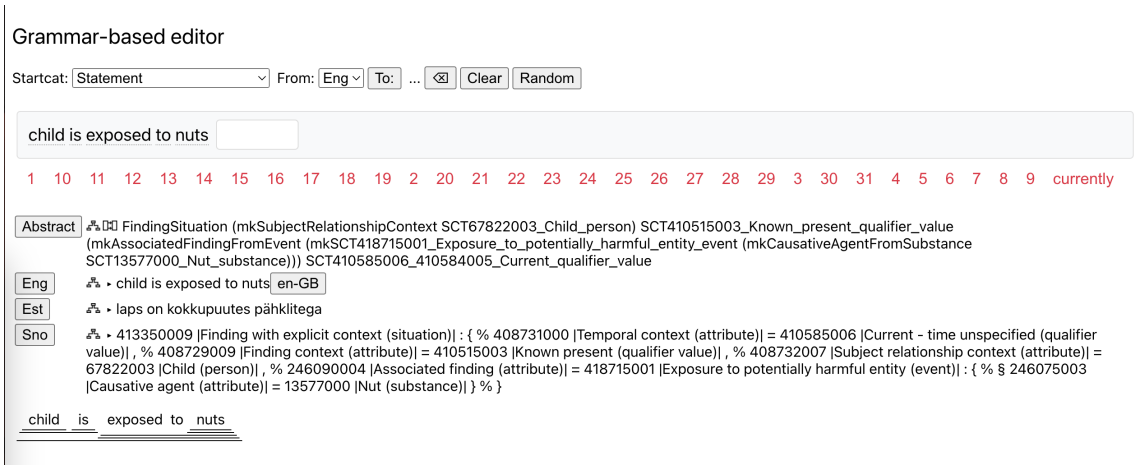

Figure 4. *First Text-Dialog System UI prototype*

user interface prototypes which will be henceforth briefly described.

The first prototype (Figure 4 provides an interactive way of modifying the input text by allowing users to easily change specific parts of the text by clicking on words. For example, if the user enters "grandmother has diabetes", they can click on "grandmother" to choose a different person. The flexibility and ease of text modification in this prototype make it an attractive option for users who need to quickly edit and adjust their input.

The second prototype employs a Google Translate-like analogy, emphasizing whether the enter text is grammatically correct and translatable into SNOMED language. This prototype focuses on demonstrating the machine-readability and translation capabilities of the system.

While existing prototypes are standalone web applications, but the integration of the technology into existing software systems, where clinical data is recorded, is a crucial step in bringing the benefits of this technology to real-world applications.

The currently favored approach that was mentioned by the researchers in interviews conducted by the author is the creation of a **browser extension**, which would collaborate with the existing web-based healthcare applications. The browser plugin would analyze the text input fields in the existing software systems and provide real-time feedback, guidance, and suggestions based on the grammar rules of the Controlled Natural Language (CNL). This would enable users to continue using their familiar interfaces while benefiting from the enhancements provided by the Text-Dialog System. Moreover, the browser plugin approach would provide a layer of abstraction between the Text-Dialog System and the existing software systems, allowing for easier updates and maintenance of the Text-Dialog System's components without the need for major changes in the underlying systems.

4.1.3 Translation API

Once free-text data adhering to the CNL grammar has been recorded in the electronic health records, methods are needed for converting it into structured data. This part of the system has not yet been explored by the researchers. Potential approaches might include the creation of a Text-Dialog **translation API** that accepts free-text data as input and returns the corresponding structured data based on SNOMED CT. The API should be designed to handle requests from statistical tools or systems storing electronic health records.

4.2 Technology readiness assessment methodology

TRL	Engineering terms relevant to software	Additional explanation to cover software	Description	Requirements	Verification	Viability
4	Alpha version	See definition in A.1.1	Documentation as for TRL 3 plus: • User manual • Design file	Clear identification of the domain of applicability. Requirements for solutions to a range of problems specified. All use cases implemented.	V&V process is partially completed, or completed for only a subset of the functionality or problem domain. Execution target is representative of the final target, including hardware aspects. V&V activities executed in a representative simulated laboratory environment.	Feasibility to complete missing functionality and reach a product level quality demonstrated.
5	Beta version	See definition in A.1.2	Full documentation according to the applicable software engineering and quality standards, including test reports and application examples.	Formal definition of the domain of (re)use and associated variability features of the implementation. All use cases and error handling specified.	Validated against the requirements of the complete domain of applicability including robustness. Quality assurance aspects taken into account. V&V activities executed in an end-to-end representative laboratory environment including real target (hardware execution target).	Feasibility to fix reported problems within available resources evaluated. User support organization in place.

Figure 5. *Target TRL criteria [15]*

The purpose of the assessment is to evaluate the current maturity of the technology and its primary components, identify any challenges and limitations, and provide recommendations for future development. By examining each component's readiness, we aim to better understand the overall functionality of the Text-Dialog System and its potential for successful implementation in real-world healthcare settings.

Resources available for the TRA include an extensive interview with Kankainen, the leading researcher of Text-Dialog System. Existing documentation and the interview will provide the evidence needed for the assessment. The assessment will be carried out by the author of this thesis.

It is important to note that a clear objective has not yet been set for the technology, since the development is still in an exploratory phase. Thus the current results only provide an overview of the software's readiness for the currently known high-level objective and requirements. Once more detailed requirements are established, reassessing TRL is recommended to provide more accurate results, which will allow for a better understanding of the software's maturity and potential for success.

4.2.1 Technology readiness level framework

The assessment will use readiness level descriptions from the Space Engineering Technology Readiness Level (TRL) Guidelines compiled by the European Cooperation for Space Standardization. The guidelines contain a chapter dedicated to readiness levels of software technologies, which are distinct from the readiness levels for hardware and electronics systems. The Space Engineering TRL Guidelines were chosen after comparing them to other TRL frameworks. The Space Engineering software-specific TRL Guidelines were found to be more suitable for assessing software technologies, whereas other framework (Horizon Europe TRL, ISO standard TRL) were not tailored for software technology assessments.

Initally, the author completed a readiness assessment using the TRL framework used by Enterprise Estonia in their funding programme for applied research. However, during this assessment, severe shortcomings were discovered for that framework, making it unsuitable for any kind of software techology readiness assessment. These discoveries are explained in more detail in the discussion part of this work, in section 6.1.

The framework for assessing the technology readiness level (TRL) of the Text-Dialog System is divided into the following four aspects: Description, Requirements, Verification, and Viability. For each aspect, the state of the three primary components of the technology—grammar, user interface, and translation API—will be evaluated and compared to the TRL framework provided by the European Cooperation for Space Standardization.

Description

The description aspect focuses on the current state of the technology and its components. This includes identifying the functionalities that have been implemented and the overall architecture of the system. An assessment of the technology's current state will be conducted, with the objective of determining the level of maturity and functionality of each component.

Requirements

The requirements aspect entails determining the extent to which the technology addresses its intended use cases and meets the needs of its users. This involves examining the specific needs and use cases the technology aims to fulfil and evaluating the degree to which the current implementation successfully addresses these requirements. The assessment of the technology's requirements will provide insight into the potential utility and effectiveness of the system in real-world healthcare settings.

Verification

The verification aspect focuses on the extent to which the technology has been tested and verified. This involves evaluating the testing methodologies and procedures applied to the technology and its components, as well as the results obtained from these tests. Verification aims to determine the reliability and correctness of the system in its current state and its ability to perform its intended functions in real-world scenarios.

Viability

The viability aspect assesses the feasibility of the technology and its components to be integrated into a larger system and to meet the performance and usability requirements of its intended users. This includes evaluating the potential challenges and limitations of the technology and identifying areas for future development and improvement.

Before beginning an assessment, the type of software needs to be identified for the purpose of TRL definition. The relevant types, as defined in [15] are as follows:

a. Software tool
b. Software element: software that necessarily interacts with other software and possibly also with hardware. Two categories exist as follows:
    1. Building block: software conceived to be reused in a range of missions, either flight or ground software. This software is executed as part of a larger software application.
    2. Specific software: software that is targeting a specific applic ation and that is not conceived to be reused in another domain of application, for example equipment embedded software
c. Generic software product (software tool or building block)
    1. Tailored generic software product
    2. Customized generic software product

When evaluating the software under consideration, which incorporates external open-source or third-party software (such as libraries or application modules in source or binary format), the overall Technology Readiness Level (TRL) assessment factors in the determined maturity of the included third-party software [15].

In the interview with Kankainen, the following modules were found to be part of the system:

- Controlled Natural Language Grammar
- Data entry user interface
- Translation APIs

In addition to these parts, the system also heavily employs the Grammatical Framework (GF), which is a special-purpose programming language for writing grammars of natural languages. Kankainen designates GF to be a highly mature technology, having been developed for more than 20 years, and in use in thousands of organisations and projects.

The system also employs the SNOMED Clinical Terms (SNOMED CT), which is not in itself a software [33], but is nevertheless a critical part of the Text-Dialog System. According to the US National Library of Medicine, SNOMED CT is the "most comprehensive and precise, multilingual health terminology in the world", which has been developed since 1965 [33].

Thus, the author assesses both GF and SNOMED CT as being fully mature technologies, and the assessment can concentrate on the first-party components and modules of the Text-Dialog System

Controlled Natural Language Grammar is the foundation and quintessential building block of all applications using the technology. It is categorised as b2: "software conceived to be reused in a range of missions, either flight or ground software. This software is executed as part of a larger software application."

Data entry user interface, or more specifically the browser extension, can be considered a software tool.

Translation APIs is considered to be a tailored generic software product, as its potential use cases are diverse and not under the control of the developers of Text-Dialog System.

In order for the Text-Dialog System product to be considered as having reached a technol-

ogy readiness level, all three of these modules need to have reached the designated level. Because no prototypes have been developed for the translation APIs, it's clear that the readiness level of the whole product can not be more than TRL1 (first formulation). The assessment will be conducted separately for all parts of the system.

4.3 Key personnel

The background, qualifications, and experience of the key personnel involved in the project provide an indication of the level of expertise and knowledge that is being applied to the technology's development. This can help instill confidence in the project's potential for success and ensure that the constraints and requirements for the system are understood, even if they are not yet documented.

Author

Kristian Kankainen is the visionary behind the Text-Dialog System and is the leading researcher.

Developers

Inari Listenmaa has played a crucial role in the development of Text-Dialog System as a prominent expert in the Grammatical Framework. Listenmaa has contributed to implementing the platform's best practices, ensuring the technology's effectiveness and usability.

Advisors

Igor Bossenko, a seasoned healthcare ICT solutions creator, has validated the scalability aspect of the Text-Dialog System by suggesting that the algorithm could be based on clinical data modeling. This approach not only benefits the technology but also allows for the observation of semantic information needed for data modeling through textual analysis.

Eno Martin Lotman, a health technology accelerator mentor, has validated the potential of Text-Dialog System to participate in accelerator programs, demonstrating its viability as a business solution in the healthcare sector.

Subject matter experts

Reet Laidoja, Madis Tiik, Gerhard Grentz, and Andres Lasn have provided valuable feedback regarding the user readiness of Text-Dialog System from the perspective of general practitioners *(perearstid)*. They have identified that the current user experience of

documenting objective findings is hindered by copying and pasting the same data between multiple text boxes. Furthermore, they have expressed their willingness to adapt their documentation habits if it enables them to input objective findings only once in a textual format.

Peeter Ross, a radiologist and e-health architect, has confirmed the need for Text-Dialog System and described current text input practices: 1) the use of widespread text bases, templates, and patterns, and 2) the reuse of texts indicating a limited need for expression. Julius Juurmaa, another radiologist, has expressed strong interest in incorporating such a text input interface into his daily work.

Liis Hamburg, a midwife, has researched the needs of midwives and validated their readiness to adopt a supported text input user interface.

4.4 Technology readiness level assessment of Text-Dialog System

Based on the TRL framework for software components presented in Table 4, we can now assess the technology readiness levels of the three primary components of the Text-Dialog System: grammar, user interface, and translation API.

4.4.1 Grammar

Description: The grammar component of the Controlled Natural Language (CNL) has a clear algorithmic formulation, and its feasibility to be implemented in software with available computing facilities has been demonstrated.

Requirements: The grammar has been developed to address a practical application, and a concrete specification of a part of the problem (maternity record standardization) has been identified.

Verification: Single algorithms of the grammar have been prototyped and tested with synthetic data, resulting in their characterization and feasibility demonstration. However, the execution target is not necessarily representative of the final target (running locally on a doctor's computer).

Viability: Feasibility to build important functions in a system architecture using the grammar has been demonstrated.

Based on these assessments, the grammar component of the Text-Dialog System can be considered to have reached **TRL 2**.

4.4.2 User interface

Description: The user interface prototypes have been developed, and their feasibility to be implemented in software with available computing facilities has been demonstrated.

Requirements: The user interface addresses practical applications and some use cases have been implemented.

Verification: Some functionalities of the user interface have been implemented and tested, allowing for the demonstration of global operation and performance. The execution target (web application) is representative of the final target (browser extension). Preliminary verification and validation activities have been executed in a simulated laboratory environment.

Viability: Feasibility to build an operational system taking into account preliminary performance and usability aspects has been demonstrated.

Based on these assessments, the user interface component of the Text-Dialog System can be considered to have reached **TRL 2**.

4.4.3 Translation API

Description: The translation API component of the Text-Dialog System is still in its conceptual stage and has not been prototyped or implemented. Existing user interface prototypes demonstrate the capability of the technology for bi-directional translation between controlled natural language and SNOMED CT, but no prototypes have been developed how third-parties would programmatically use the system for translations.

Requirements: The practical application for the translation API has been identified, but a concrete specification of the problem has not yet been documented.

Verification: As the translation API is still in its conceptual stage, no verification activities have been conducted.

Viability: The feasibility to build important functions in a system architecture using the translation API has not yet been demonstrated. However, the existing systems already offer

translation capabilities, thus implementing an additional API can be deemed feasible.

Based on these assessments, the translation API component of the Text-Dialog System can be considered to have reached **TRL 1**.

4.5 Summary of assessment

The technology readiness levels of the three primary components of the Text-Dialog System have been assessed as follows:

- Grammar: TRL 2
- User Interface: TRL 2
- Translation API: TRL 1

As all three components must reach a specific TRL for the entire system to be considered at that level, the Text-Dialog System is currently at **TRL 1**, mainly due to the conceptual stage of the translation API.

In conclusion, the Text-Dialog System is still in its early stages of development. The grammar component has shown promise in terms of its ability to support a wide variety of terms and compatibility with SNOMED CT. However, it does not yet fully cover all possible use cases and has not undergone testing and refinement in real-world scenarios. The user interface prototypes have demonstrated the potential for improved user experience, but they have not been integrated into actual workflows, and the feasibility of building a system that meets performance and usability requirements remains unclear. The translation API, which is crucial for the conversion of free-text data into structured data, has not yet been developed.

5. Development plan for Text-Dialog System

The future development of Text-Dialog System holds immense potential for revolutionizing data capture [34] in the healthcare industry. To ensure the successful realization of this potential, it is crucial to assess the various approaches available for the advancement of the technology. This chapter explores the feasibility of different development approaches and their implications on the Text-Dialog System's progress and adoption in the healthcare sector are explored.

The chapter presents a SWOT analysis for each of the four proposed development approaches, highlighting their strengths, weaknesses, opportunities, and threats. This comprehensive assessment provides valuable insights into the potential challenges and advantages associated with each approach. Following the SWOT analysis, a discussion on the recommended development direction based on the current project resources and goals is presented.

By carefully examining the various development approaches, this chapter aims to provide a clear roadmap for the Text-Dialog System's future growth, ensuring its successful integration into the healthcare industry and the realization of its full potential.

5.1 Scope for future development

Before delving deeper into the exact areas of improvement required for the Text-Dialog System to progress to a higher maturity level, this section proposes alternative development scopes that could be used to limit the work required for the technology to reach a high maturity level.

5.1.1 Approach A: core, grammar, API, and user interface

During interviews, the system's leading researcher proposed the all-encompassing approach, also shown in Figure 2, that involves the simultaneous development and improvement of all three components of the Text-Dialog System: grammar, user interface, and translation API. This method aims to provide a comprehensive solution that meets the needs of healthcare professionals and can be easily integrated into existing workflows.

By dedicating resources to all three components, this approach ensures that the technology

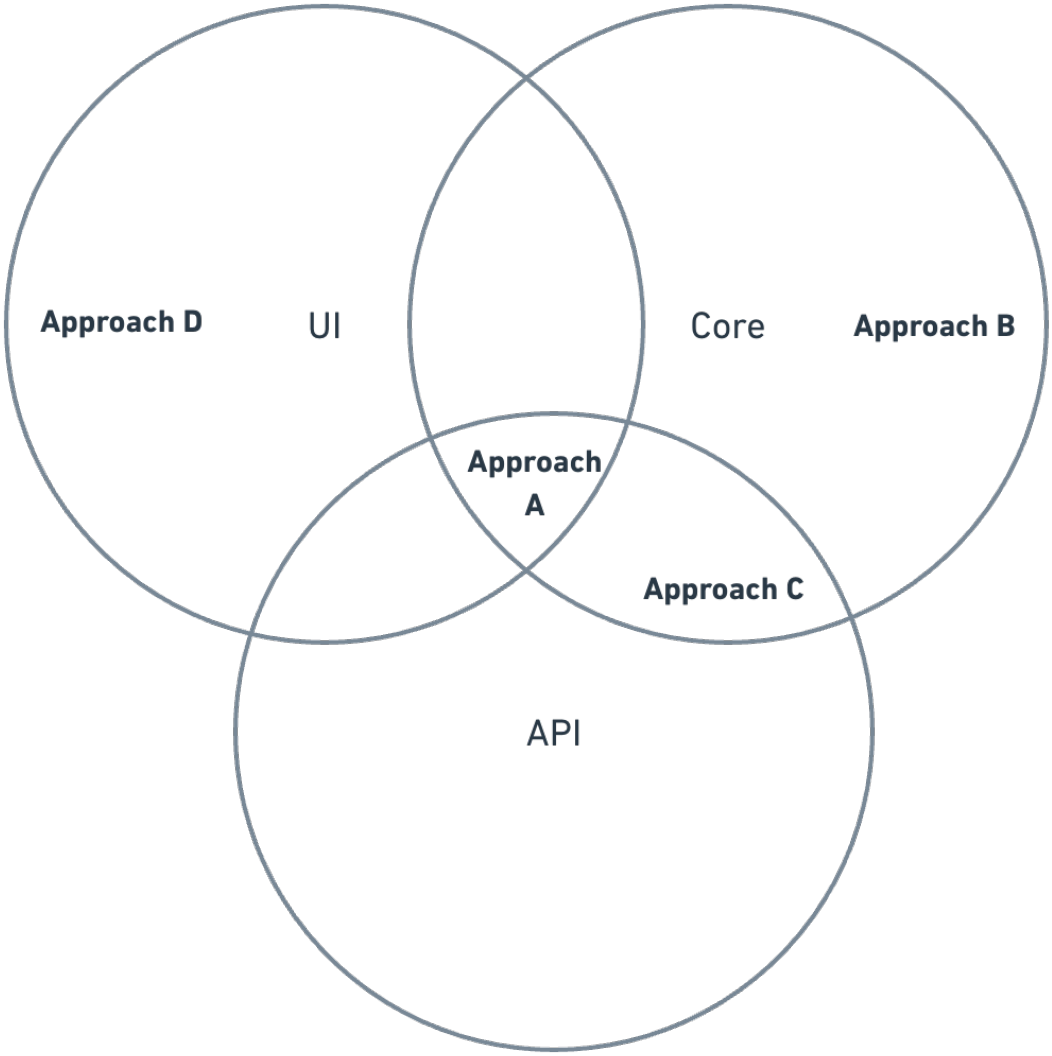

Figure 6. *Potential development scopes of Text-Dialog System*

remains balanced and cohesive, addressing the requirements of both its users and the broader healthcare system. This approach is likely to result in a more polished and unified product, which could potentially lead to higher adoption rates among healthcare professionals.

However, such full-scale development requires a more significant investment of time and resources, and the progress in each component may be slower compared to the technology-only or API-only approaches. This method also requires close collaboration between developers, healthcare professionals, and subject matter experts to ensure that the technology addresses the specific needs and challenges faced by its users.

The choice of development approach depends on the available resources and the specific needs and goals of the project [19]. The technology-only and API-only approaches may lead to faster improvements in specific components but may also limit the adoption of

the technology due to the absence of a user-friendly interface or reliance on third-party developers. On the other hand, the full-scale approach provides a more comprehensive solution addressing all aspects of the technology but may require a more significant investment in time and resources.

To determine the most suitable development approach, it is crucial to consider the needs of the target users, the resources available for the project, and the long-term goals of the Text-Dialog System [28].

5.1.2 Approach B: core technology

In this collaborative approach, the development team would focus primarily on creating and refining the underlying technology while allowing other interested parties to develop and implement their own grammars for specific use cases. This approach encourages collaboration and innovation within the community, as various organizations and developers can contribute their expertise to build grammars tailored to their unique needs and contexts.

By providing a robust and flexible foundation for the Text-Dialog System, the development team can enable a wide range of specialized grammars to be created and integrated into the system. This would result in a more versatile and powerful tool that can address the diverse requirements of the healthcare industry.

To facilitate this collaborative approach, the development team could create comprehensive documentation, guidelines, and resources for developing custom grammars for various specific medical fields. They could also establish a community platform to encourage knowledge sharing, collaboration, and support among developers and organizations working with the Text-Dialog System.

By focusing on fostering a collaborative development environment, the Text-Dialog System can become a highly adaptable and valuable tool for healthcare professionals, driving innovation and the creation of tailored solutions for various clinical settings and use cases.

5.1.3 Approach C: core, grammar and API

In this approach, in addition to the core technology, the focus would also be on enhancing the grammar component and developing a robust API or Software Development Kit (SDK) for third-party integrations. This is illustrated in Figure 7 and would allow other developers and organizations to build applications and interfaces that leverage the Text-Dialog System

Figure 7. *Development scope: Approach C*

for efficient and accurate clinical documentation.

Improvements in the grammar component could include increasing coverage, refining rules for better accuracy, and incorporating feedback from subject matter experts to ensure the grammar remains intuitive and user-friendly. Developing an API or SDK could involve creating comprehensive documentation, sample code, and support resources for developers to easily incorporate the Text-Dialog System into their applications.

By focusing on the technology aspect, the Text-Dialog System can become a more versatile and powerful tool for various healthcare applications, driving wider adoption and innovation in the industry.

5.1.4 Approach D: user interface

Alternatively, the development team could focus primarily on creating a user interface that can be adapted to work with any Grammatical Framework Controlled Natural Language (GF CNL), as shown in Figure 8. This approach would emphasize the design and implementation of a user-friendly and intuitive interface that healthcare professionals can easily use and adapt to their specific needs.

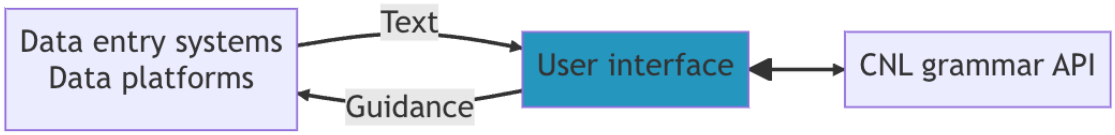

Figure 8. *Development scope: Approach D*

The user interface would need to be flexible, supporting a wide range of CNLs and use cases. It could include features such as real-time feedback, suggestions, and error checking to improve the overall user experience.

In this section, the feasibility of future development of the Text-Dialog System will be explored. The analysis will concentrate on each of the module identified in chapter 4.

The strengths, weaknesses, opportunities, and threats (SWOT) of the technology are analyzed.

Approach A is a comprehensive solution but requires significant investment in time and resources. Approach B focuses on core technology, fostering collaboration within the community. Approach C emphasizes grammar and API improvements, making it more developer-friendly. Approach D, which focuses on user interface, is not recommended due to the lack of UI/UX experts in the current team.

The most suitable development approach depends on the goals, priorities, and resources available. Collaboration between researchers, developers, and healthcare professionals is essential for the technology's success [28].

5.2 SWOT Analysis

This chapter presents a SWOT analysis for the four proposed approaches to further develop and deploy the Text-Dialog System. A comprehensive assessment of the strengths, weaknesses, opportunities, and threats associated with each approach will provide valuable insights into the potential challenges and advantages in pursuing each path.

5.2.1 Approach A: core, grammar, API, and user interface

Strengths

1. **Comprehensiveness**: Equal attention given to core functionality, usability, and integration capabilities. This approach provides a comprehensive solution that aims to deliver a complete and cohesive product that addresses the need of healthcare professionals and can easily be integrated into existing workflows.
2. **Adoption**: By providing an unified end-to-end product that addresses both technical and user experience aspects, this approach might lead to higher initial adoption rates by end-user facilities.

Weaknesses

1. **Resource Intensive**: All-encompassing, and thus most capital-intensive approach. The full-scale development of all components requires a more significant investment of time and resources, which may strain the development team and slow down progress.
2. **Complexity**: Managing and coordinating the development of all components simultaneously may be complex and require comprehensive project management collaboration with all layers of the healthcare sector.

Opportunities

1. **Competitive advantage**: Providing a comprehensive solution that covers the core technology, grammar, API, and user interface aspects can set the Text-Dialog System apart from competing solutions. Keeping end-to-end ownership of the technology provides more control over its solutions.

Threats

1. **Delays and Competition**: Balancing the development of all components may lead to delays hindering the overall progress of the project, allowing competitors

focusing on a narrower problem set to overtake Text-Dialog. The healthcare industry is continuously evolving, and new technologies and solutions emerge regularly. Focusing on a comprehensive development approach may cause the Text-Dialog System to lag behind or miss out on opportunities to adapt to market changes and trends.

2. **Collaboration Challenges**: The success of approach A relies heavily on the collaboration and input from subject matter experts, developers, and healthcare professionals, which may be difficult to secure and maintain throughout the development process.

5.2.2 Approach B: core technology

Strengths

1. **Focus**: Concentrates on the core technology, allowing the development team to dedicate their resources and expertise on fewer problems. Providing a solid foundation allows for a wide range of specialized grammars to be created, addressing diverse requirements in the healthcare industry.
2. **Collaboration**: Scientific, research-based approach makes it feasible to establish a community of researchers, developers, and organizations collaborating on the core technology. Encouraging the development of tailored grammars for specific medical fields and use cases can result in a more adaptable and valuable tool for healthcare. Demonstrating the potential of the core technology and its adaptability to various applications might attract interest, funding, and partnerships from health organizations interested in leveraging the system for their specific needs.

Weaknesses

1. **Misprioritization**: Concentrating on the core without prototyping the integrations and UI can lead to misprioritisation of features, as it will not be possible to gather user feedback before another organization has built applications that use the technology.
2. **Limited control over end-user experience**: Focusing on the core technology might limit the development team's ability to ensure a consistent and user-friendly experience across different applications and interfaces built on top of the Text-Dialog System.

Opportunities

1. **Collaboration**: Scientific, research-based approach makes it feasible to establish a community of researchers, developers, and organizations collaborating on the core technology. Encouraging the development of tailored grammars for specific

medical fields and use cases can result in a more adaptable and valuable tool for healthcare. Demonstrating the potential of the core technology and its adaptability to various applications might attract interest, funding, and partnerships from health organizations interested in leveraging the system for their specific needs.

Threats

1. **Reliance on third-party developers**: This approach depends on the contributions and expertise of other developers and organizations to create and implement grammars tailored to their unique needs and contexts, which might introduce variability in quality and consistency.
2. **Fragmentation**: With multiple developers and organizations working on their own grammars and applications, there is a risk of fragmentation and lack of standardization, making it harder to maintain consistency and interoperability across the different implementations.
3. **Quality and consistency**: Relying on third-party developers to create and implement grammars might result in varying quality and consistency across different applications, which could negatively impact the overall effectiveness and adoption of the Text-Dialog System.

5.2.3 Approach C: core, grammar and API

Strengths

1. **Critical Components**: Addresses critical components, enabling third-party integrations, encouraging wider adoption by software vendors.

Weaknesses

1. **Dependence on third-party developers**: This approach depends on the interest and expertise of third-party developers and vendors to integrate and adopt Text-Dialog System, which might not be feasible or desirable for all organizations.

Opportunities

1. **Adoption by Vendors**: Providing an out-of-the-box API makes it possible for software vendors to adopt the technology with less effort (compared to approach B).

Threats

1. **Developer interest**: The potential lack of interest (or expertise) in the developer community to work with the Text-Dialog System. Dependence on interest from third-party developers and vendors to integrate and adopt Text-Dialog System.
2. **Competition**: Competition from competitors offering an end-to-end solution.

5.2.4 Approach D: user interface

Strengths

1. **Small Vertical**: Concentrates on a small vertical, while leaving the use case undefined. This well-designed user interface could potentially be adapted to work with a wide range of controlled languages, making it a versatile tool for various applications and use, perhaps even outside of healthcare applications.

Weaknesses

1. **Lack of expertise**: The current development team lacks UI/UX expertise, focusing primarily on the UI/UX interface using currently available resources is thus not feasible.

Opportunities

1. **Multiple Applications**: A well-designed user interface could be used for applications outside of the healthcare domain.

Threats

1. **Lack of adoption**: The UI approach might not be adopted by healthcare professionals or organizations if it does not meet their specific needs and requirements.

5.3 Summary of SWOT analysis

In summary, the SWOT analysis of the four proposed approaches for the further development and deployment of the Text-Dialog System reveals distinct strengths, weaknesses, opportunities, and threats associated with each approach.

Approach A offers a comprehensive solution that addresses core functionality, usability, and integration capabilities, potentially leading to higher initial adoption rates. However, it is resource-intensive, complex, and may face delays and competition due to its broad

scope.

Approach B focuses on the core technology, allowing for collaboration and adaptability to diverse healthcare requirements. This approach, however, may lead to misprioritization of features and limited control over the end-user experience.

Approach C targets critical components for third-party integrations, encouraging wider adoption by software vendors. This approach relies heavily on the interest and expertise of third-party developers and vendors, which might not be feasible or desirable for all organizations.

Approach D concentrates on a small vertical with a well-designed user interface that could potentially be adapted for multiple applications. The lack of UI/UX expertise within the current development team and the risk of not meeting healthcare professionals' specific needs and requirements are significant challenges for this approach.

5.4 Recommendation for development direction

Considering the current project team, this study suggests initially focusing on Approach B to expedite the technology maturation process. Adopting this focused scope facilitates faster progression. It is worth noting that Approaches A and C are dependant on the maturation of the core technology. Given that Approach B independently generates value, it is advisable to commence with a confined scope, which can be subsequently expanded.

Moreover, the scientific and research-based nature of Approach B can attract interest, funding, and partnerships from health organizations interested in leveraging the system for their specific needs. This approach also allows the team to focus on the most critical aspects of the technology, ensuring its effectiveness.

Once a higher maturity level (TRL6 or higher) has been reached, focus should be shifted to the user interface and integrations needs. A simple illustration of the proposed approach, based on best practices according to [28], also described in section 3.2 is shown in Figure 9.

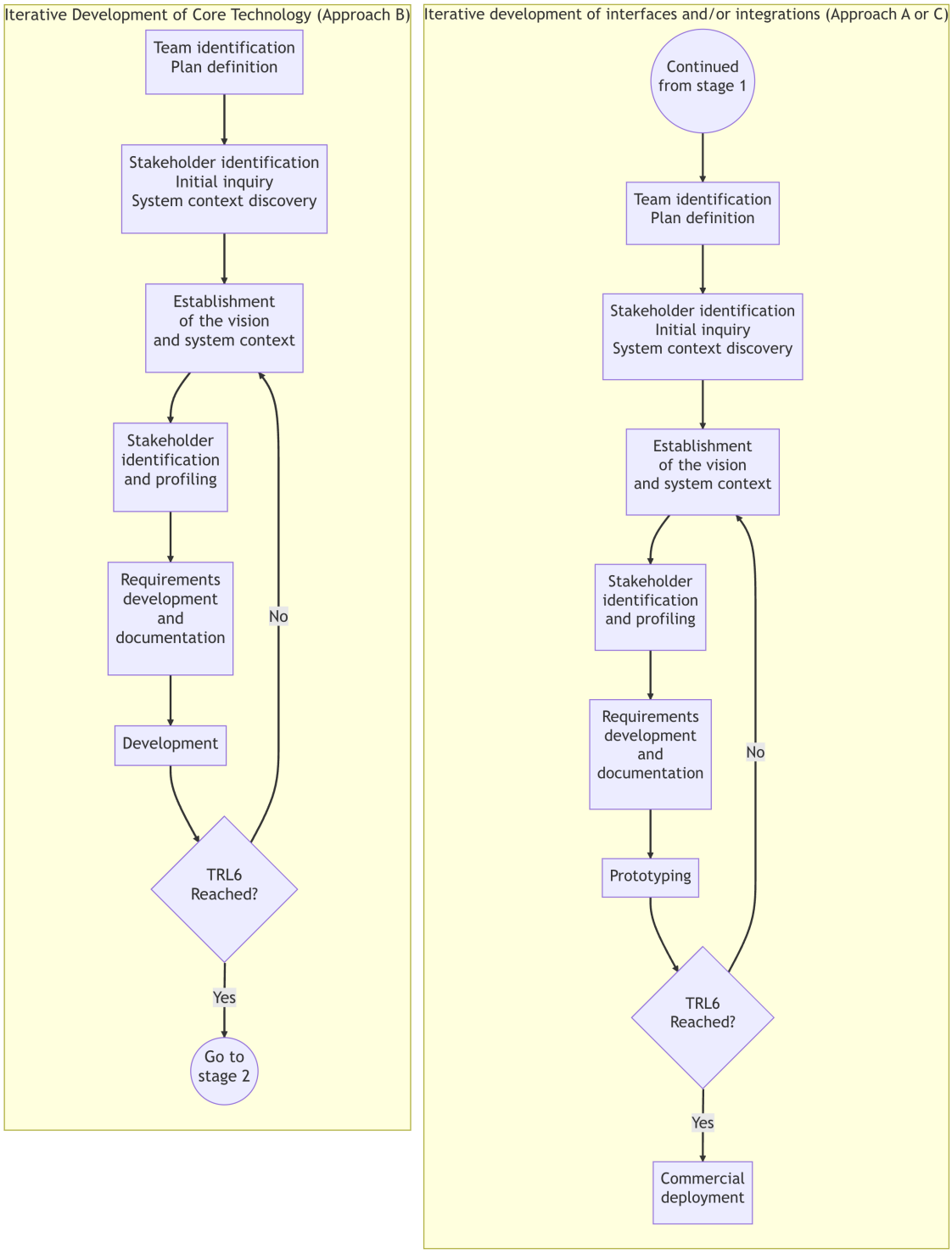

Figure 9. *Proposed development steps for Text-Dialog System*

6. Discussion

6.1 Challenges of software technology maturity assessment

While assessing the maturity of the Text-Dialog System, several problems and issues were encountered:

Many frameworks, no standards

Although TRLs are in use in a vast amount of organizations across the world (including but not limited to NASA, European Space Agency, European Commission, Enterprise Estonia, U.S Department of Energy, and many more), there are no robust standards outside of the space industry.

TRL is heavily used in funding calls by the European Commission, but TRL standards tailored for different fields do not exist, and guidelines for assessment of a TRL have not been published. As TRL for projects sent to funding calls is usually self-reported, the assessment for each project in a funding call is done by a different team. This leads to incomparable results even across projects in a single funding call.

TRL for software

Some critics argue that TRL has not been proven effective outside of space and weapons technology for other types of innovation. Thus the "absence of discipline-specific guides" is likely to cause confusion and cause TRL to become a subject of abuse in efforts to obtain EU funding [2].

One of the primary reasons TRLs are not suitable for software development is that the framework was initially designed for hardware projects. Hardware development typically follows a linear path, where a physical prototype is iteratively improved until it reaches a mature and market-ready state. In contrast, software development is characterized by its non-linear, iterative, and incremental nature. The rapid advancement of software technologies and the intangible nature of software products make it difficult for the TRL framework to accurately capture the nuances of software development [35].

The software industry is characterized by rapid changes and continuous innovation, with new technologies and platforms emerging constantly. This fast-paced environment often

requires software developers to adapt quickly, making decisions and pivoting as needed. The TRL framework, however, is a more rigid and structured approach, which may not accommodate the agility and flexibility required in software development. For instance, software projects may quickly progress through various stages of maturity or revert to earlier stages when new issues are discovered. This fluidity can be difficult to capture accurately within the constraints of the TRL framework.

Additionally, software development methodologies, such as Agile and DevOps, emphasize continuous delivery, collaboration, and responsiveness to change. These approaches can be difficult to reconcile with the TRL framework, which is more focused on assessing the maturity of a technology at a specific point in time.

Highly mature, yet unsuccessful

The success of a software product is not solely dependent on its technological maturity but also on factors such as market fit, user experience, and business model viability. The TRL framework, with its emphasis on technology readiness, does not provide a comprehensive assessment of a software product's readiness for market launch or its potential for success. Alternative approaches, such as Agile methodologies or software-specific maturity models, may be more appropriate for evaluating software development projects and ensuring their success in the ever-evolving technology landscape.

Local landscape: Enterprise Estonia

Enterprise Estonia, as part of its programme for applied research, has published a set of TRL criteria tailored for IT projects. The author of this work attempted to use this set of criteria for assessing the Text-Dialog System, but did not find the results of the assessment useful or usable for the following reasons.

One-sided requirements development: The Enterprise Estonia levels assume clients possess a thorough understanding of their needs, requiring them to create a comprehensive requirements document. However, requirements should be developed collaboratively with the development team to ensure clear understanding and avoid miscommunication. Encouraging close collaboration between clients and development teams throughout the process is crucial for successful project outcomes. The Text-Dialog System is still in an early phase of development and does not even possess a client, rendering the Enterprise Estonia levels unusable.

Commercialization and project management in TRL criteria: Upon analyzing the TRL4 criteria for software provided by Enterprise Estonia, it is evident that these criteria

combine elements of both technology maturation and commercialization. However, mixing the two approaches into a single framework is not ideal for several reason. By combining these objectives into a single framework, developers may face difficulty in setting clear priorities and balancing resource allocation between advancing the technology and developing market-ready products. This challenge arises because the goals and methods of technology maturation and commercialization projects differ significantly. Attempting to address both in the TRL framework can lead to confusion, inefficiency, and potential conflicts of interest among team members.

Opinionated: Enterprise Estonia levels contain criteria that should not be part of a TRL framework because they mix technical maturity with project management activities and decisions. The purpose of the TRL framework is to assess the maturity of a technology, not to measure project management practices or make decisions about the project's direction. For example, pricing and costs are part of the assessment criteria.

The author of this thesis would recommend maintaining separate frameworks for technology maturation and commercialization, tailoring each to the specific needs and priorities of the respective development approach. By doing so, developers can ensure a more streamlined and focused development process, ultimately leading to better outcomes for both technology advancement and product commercialization.

6.2 Avenues for software technology maturation optimisation

One goal of this work was to discover whether it's possible to plan a roadmap for a system that prioritises technology maturation over other concerns.

TRL was originally devised for use in acquisition programs, but for Text-Dialog System the requirements for the technology are defined by the developer themselves. This allows for a wiggle room and the following conclusion. Speeding up technology maturation can easily be accomplished by reducing the scope, and thus the work that needs to be done. This finding aligns with current trends in software development that favor development and delivery in small iterations, with continuous feedback. Whenever possible, the author of this work recommends the same approach for technology development. Try to find ways to divide the development of a technology into smaller pieces.

For maturation-focused development, success is defined as completion of all the criteria and requirements to reach TRL4. It's important to higlight that it's possible for a highly mature, high-TRL technology to be commercially unattractive and unviable. Developers of high-TRL technologies should focus not only on technical maturity but also on addressing

market needs, promoting their solutions, and ensuring that their technology is cost-effective and easy to integrate. By doing so, they can increase the commercial attractiveness and popularity of their products.

6.3 Alternative solutions

Design thinking approach

Instead of focusing solely on the technological maturation of the Text-Dialog System, the research could adopt a design thinking approach that emphasizes user-centered design principles. This would involve identifying and understanding the needs of healthcare professionals and patients, and using this understanding to inform the development and maturation strategy for the system [36].

Benchmarking with existing systems

Another alternative approach could involve conducting a comparative analysis of the Text-Dialog System with other existing clinical data capturing systems. This analysis would help place the current system in the marketplace, and by learning from the success and challenges faced by other systems, researchers could develop a more effective roadmap for the maturation of the Text-Dialog System [37].

Requirements development

Instead of a high-level SWOT analysis, investing in requirements development and identifying the most critical use cases for the Text-Dialog System would allow researchers to prioritize development areas with the most significant impact on the system's functionality and user experience.

Business focus

If this thesis were to be undertaken by someone pursuing a business development degree, the focus would likely shift from the technical aspects of the Text-Dialog System to the strategic, operational, and market-oriented aspects of the project.

6.4 Future work

To further advance the Text-Dialog System, the following research directions should be considered: **Integration with other healthcare systems:** Investigate the compatibility and interoperability of the Text-Dialog System with various electronic health record (EHR) systems and other relevant healthcare applications. Develop standardized data exchange

protocols to facilitate seamless data transfer between systems.

Customization and adaptability: Investigate methods and processes to allow healthcare organizations to tailor the Text-Dialog System to suit their specific needs, workflows, and terminologies. Develop a modular architecture that enables users to select and configure components according to their requirements.

Commercialization and market assessment: Conduct a market assessment to identify potential customers, competitors, and opportunities for the Text-Dialog System. Develop a business plan and marketing strategy to promote the adoption of the system in the healthcare industry.

Regarding **technology maturation**, additional research is required to comprehend the challenges of employing Technology Readiness Levels (TRLs) in software projects. The author anticipates the development of new TRL frameworks tailored to specific industries and sectors, particularly those concentrating on software development. Furthermore, analyzing the compatibility of TRLs with various software development methodologies and evaluating their efficacy in different contexts constitutes a promising area for future investigation.

7. Summary

This thesis focuses on refining the Text-Dialog System, which converts clinical data into structured, machine-readable data using controlled natural language. The main challenge is determining the system's current technological maturity and creating a clear development roadmap. The goal is to assess the system's maturity and prioritize accelerated development areas for successful healthcare industry integration.

The technology readiness assessment shows that the grammar and user interface components are at level 2 (algorithm), while the translation API is at level 1 (mathematical formulation), indicating early development stages.

A SWOT analysis evaluates four development areas. Approach B (core technology) is recommended based on resources and goals, giving priority to core technology over integrations and user interface to achieve broader medical term coverage, versatility, and adoption.

The discussion compares maturation-focused and commercialization-focused development, as well as the limitations of the TRL framework for software projects. By narrowing the scope, higher maturity levels can be achieved with fewer resources, aligning with current software engineering best practices.

References

[1] Samuel A. Fricker, Rainer Grau, and Adrian Zwingli. "Requirements Engineering: Best Practice". en. In: *Requirements Engineering for Digital Health*. Ed. by Samuel A. Fricker, Christoph Thümmler, and Anastasius Gavras. Cham: Springer International Publishing, 2015, pp. 25–46. ISBN: 978-3-319-09797-8. DOI: 10.1007/978-3-319-09798-5_2. URL: https://link.springer.com/10.1007/978-3-319-09798-5_2 (visited on 04/08/2023).

[2] Mihály Héder. "From NASA to EU: the evolution of the TRL scale in Public Sector Innovation". en. In: *THE INNOVATION JOURNAL* 22.2 (2017). Number: 2, pp. 1–23. ISSN: 1715-3816. URL: https://www.innovation.cc/discussion-papers/22_2_3_heder_nasa-to-eu-trl-scale.pdf (visited on 03/04/2023).

[3] Kristian Kankainen et al. "Using SNOMED CT as a Semantic Model for Controlled Natural Language Guided Capture of Clinical Data". en. In: 2022. URL: https://ceur-ws.org/Vol-3264/HEDA22_paper_9.pdf (visited on 01/25/2023).

[4] Karima Bourquard, Franck Le Gall, and Philippe Cousin. "Standards for Interoperability in Digital Health: Selection and Implementation in an eHealth Project". In: *Requirements Engineering for Digital Health*. Ed. by Samuel A. Fricker, Christoph Thümmler, and Anastasius Gavras. Cham: Springer International Publishing, 2015, pp. 95–115. ISBN: 978-3-319-09798-5. DOI: 10.1007/978-3-319-09798-5_5. URL: https://doi.org/10.1007/978-3-319-09798-5_5.

[5] Alison L. Olechowski et al. "Technology readiness levels: Shortcomings and improvement opportunities". en. In: *Systems Engineering* 23.4 (2020). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sys.21533, pp. 395–408. ISSN: 1520-6858. DOI: 10.1002/sys.21533. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.21533 (visited on 03/04/2023).

[6] John C Mankins. *Technology Readiness Levels: A White Paper*. en. Apr. 1995. URL: https://www.spacepropulsion.org/uploads/2/5/3/9/25392309/john_mankins_paper_of_4-6-95_trl.pdf (visited on 05/08/2023).

[7] John C. Mankins. "Technology readiness assessments: A retrospective". en. In: *Acta Astronautica* 65.9 (Nov. 2009), pp. 1216–1223. ISSN: 0094-5765. DOI: 10.1016/

j.actaastro.2009.03.058. URL: https://www.sciencedirect.com/science/article/pii/S0094576509002008 (visited on 03/04/2023).

[8] *Technology Readiness Levels Handbook for Space Applications*. en. Sept. 2008. URL: https://artes.esa.int/sites/default/files/TRL_Handbook.pdf (visited on 05/08/2023).

[9] *EVS-EN 16603-11:2019*. en-GB. URL: https://www.evs.ee/en/evs-en-16603-11-2019 (visited on 04/21/2023).

[10] *Preparing for our future: Developing a common strategy for key enabling technologies in the EU*. en. Sept. 2009. URL: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex%3A52009DC0512 (visited on 05/08/2023).

[11] *Horizon 2020 - Work Programme 2014-2015*. en. 2014. URL: https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-ga_en.pdf.

[12] *Horizon Europe Work Programme 2023-2024*. en. 2022. URL: https://cinea.ec.europa.eu/news-events/news/horizon-europe-work-programme-2023-24-now-available-2022-12-07_en.

[13] *„Sektoritevahelise mobiilsuse toetamine" elluviimiseks struktuuritoetuse andmise tingimused ja kord*. ee. Oct. 2022. URL: https://www.riigiteataja.ee/akt/106102020007?leiaKehtiv (visited on 04/21/2023).

[14] *Tehnilise valmisoleku taseme määratlemine IT-projektides*. et. URL: https://www.eas.ee/wp-content/uploads/2021/06/EAS_TRL_A4-IT-1.pdf.

[15] *Space engineering technology readiness level (TRL) guidelines*. Mar. 2017.

[16] *Technology Readiness Assesssment Guide*. EN. Tech. rep. U.S. Government Accountability Office, Jan. 2020. URL: https://www.gao.gov/products/gao-20-48g.

[17] H.F. Hofmann and F. Lehner. "Requirements engineering as a success factor in software projects". In: *IEEE Software* 18.4 (July 2001). Conference Name: IEEE Software, pp. 58–66. ISSN: 1937-4194. DOI: 10.1109/MS.2001.936219.

[18] Ian Sommerville and Pete Sawyer. *Requirements engineering: a good practice guide*. en. Chichester, Eng. ; New York: Wiley, 1999. ISBN: 978-0-471-97444-4.

[19] James Robertson and Suzanne Robertson. *Mastering the Requirements Process*. 3rd. Addison-Wesley Professional, 2012.

[20] George Mathew and Tim Menzies. “Software Engineering’s Top Topics, Trends, and Researchers”. In: *IEEE Software* 35.5 (Sept. 2018). Conference Name: IEEE Software, pp. 88–93. ISSN: 1937-4194. DOI: 10.1109/MS.2018.3571230.

[21] Karl Wiegers and Joy Beatty. *Software Requirements*. 3rd. Microsoft Press, 2013.

[22] Pierre Bourque and R. E. Fairley, eds. *SWEBOK: guide to the software engineering body of knowledge*. en. Version 3.0. OCLC: 880350861. Los Alamitos, CA: IEEE Computer Society, 2014. ISBN: 978-0-7695-5166-1.

[23] Klaus Pohl. *Requirements Engineering*. Springer, 2010.

[24] Arthur Ahimbisibwe, Robert Y Cavana, and Urs Daellenbach. “A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies”. In: *Journal of Enterprise Information Management* 28.1 (Jan. 2015). Publisher: Emerald Group Publishing Limited, pp. 7–33. ISSN: 1741-0398. DOI: 10.1108/JEIM-08-2013-0060. URL: https://doi.org/10.1108/JEIM-08-2013-0060 (visited on 04/21/2023).

[25] Gerd Stefan Brost and Mario Hoffmann. “Identifying Security Requirements and Privacy Concerns in Digital Health Applications”. en. In: *Requirements Engineering for Digital Health*. Ed. by Samuel A. Fricker, Christoph Thümmler, and Anastasius Gavras. Cham: Springer International Publishing, 2015, pp. 133–154. ISBN: 978-3-319-09797-8. DOI: 10.1007/978-3-319-09798-5_7. URL: https://link.springer.com/10.1007/978-3-319-09798-5_7 (visited on 04/21/2023).

[26] Nadezhda Purtova, Eleni Kosta, and Bert-Jaap Koops. “Laws and Regulations for Digital Health”. In: *Requirements Engineering for Digital Health*. Ed. by Samuel A. Fricker, Christoph Thümmler, and Anastasius Gavras. Cham: Springer International Publishing, 2015, pp. 47–74. ISBN: 978-3-319-09798-5. DOI: 10.1007/978-3-319-09798-5_3. URL: https://doi.org/10.1007/978-3-319-09798-5_3.

[27] Ai Keow Lim Jumelle and Ioana Ispas. “Ethical Issues in Digital Health”. In: *Requirements Engineering for Digital Health*. Ed. by Samuel A. Fricker, Christoph Thümmler, and Anastasius Gavras. Cham: Springer International Publishing, 2015, pp. 75–93. ISBN: 978-3-319-09798-5. DOI: 10.1007/978-3-319-09798-5_4. URL: https://doi.org/10.1007/978-3-319-09798-5_4.

[28] Mojca Volk, Niklas Falk-Andersson, and Urban Sedlar. “How to Elicit, Analyse and Validate Requirements for a Digital Health Solution”. In: *Requirements Engineering for Digital Health*. Ed. by Samuel A. Fricker, Christoph Thümmler, and Anastasius Gavras. Cham: Springer International Publishing, 2015, pp. 155–188. ISBN: 978-

3-319-09798-5. DOI: 10.1007/978-3-319-09798-5_8. URL: https://doi.org/10.1007/978-3-319-09798-5_8.

[29] *NASA Systems Engineering Handbook*. en. Revision 2. NASA, 2016. URL: https://www.nasa.gov/connect/ebooks/nasa-systems-engineering-handbook.

[30] Nirnaya Tripathi et al. "An anatomy of requirements engineering in software startups using multi-vocal literature and case survey". en. In: *Journal of Systems and Software* 146 (Dec. 2018), pp. 130–151. ISSN: 0164-1212. DOI: 10.1016/j.jss.2018.08.059. URL: https://www.sciencedirect.com/science/article/pii/S0164121218301729 (visited on 04/24/2023).

[31] Kristian Juha Ismo Kankainen et al. "Digital health data capture with a controlled natural language". en. In: ().

[32] Hamburg, Liis, Peeter Ross, and Kristian Juha Ismo. "Feasibility of maternity record standardization on the example of midwives' free text entries". en. In: (2021).

[33] *Overview of SNOMED CT*. eng. Product, Program, and Project Descriptions. Publisher: U.S. National Library of Medicine. URL: https://www.nlm.nih.gov/healthit/snomedct/snomed_overview.html (visited on 05/08/2023).

[34] Christophe Gaudet-Blavignac et al. "Use of the Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) for Processing Free Text in Health Care: Systematic Scoping Review". EN. In: *Journal of Medical Internet Research* 23 (Jan. 2021). Company: Journal of Medical Internet Research Distributor: Journal of Medical Internet Research Institution: Journal of Medical Internet Research Label: Journal of Medical Internet Research Publisher: JMIR Publications Inc., Toronto, Canada, e24594. DOI: 10.2196/24594. URL: https://www.jmir.org/2021/1/e24594 (visited on 05/08/2023).

[35] Jr Blanchette et al. *Beyond Technology Readiness Levels for Software: U.S. Army Workshop Report:* en. Tech. rep. Fort Belvoir, VA: Defense Technical Information Center, Dec. 2010. DOI: 10.21236/ADA535517. URL: http://www.dtic.mil/docs/citations/ADA535517 (visited on 04/16/2023).

[36] Tim Brown. "Design Thinking". In: *Harvard Business Review* (June 2008). Section: Innovation. ISSN: 0017-8012. URL: https://hbr.org/2008/06/design-thinking (visited on 05/07/2023).

[37] Tim Stapenhurst. *The Benchmarking Book*. en. Google-Books-ID: kaRWPZzRq5cC. Routledge, June 2009. ISBN: 978-1-136-43468-6.

Appendices

Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis[1]

I Rene Allkivi

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "A Roadmap for the Rapid Maturation of the Text-Dialog System", supervised by Janek Metsallik and Kristian Juha Ismo Kankainen
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

10.05.2023

[1]The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 - Discarded technology readiness assessment

This appendix contains TRL assessment of the technology that was done using Enterprise Estonia's TR levels. The assessment result was deemed unsuitable for further use and follow-up.

Assessment (TRL3)

Requirement (estonian)	Requirement (english, translated)	Assessment
Analüütiliste uuringutega on kontrollitud ennustusi, loodud eeldused algoritmide loomiseks	Predictions have been verified through analytical studies, forming the basis for algorithm development	The paper presents an extended language that can capture numerical values and units. (Digital health data capture with controlled natural language)
Olemas on saadaval olevate tarkvara algoritmide ülevaated	A review of existing software algorithms has been conducted	Yes. (Digital health data capture with controlled natural language) (Using SNOMED CT as a Semantic Model for Controlled Natural Language Guided Capture of Clinical Data)
Esmase kodeerimisega on kontrollitud, kas tarkvara suudab töövajaduse rahuldada	Initial prototype code has been developed to assess whether the software can fulfill the desired requirements	The technology has been implemented in a prototype for Estonian concrete grammar. (Using SNOMED CT as a Semantic Model for Controlled Natural Language Guided Capture of Clinical Data)
Laborikatsetega on kontrollitud rakendamise teostatavust	The feasibility of implementation has been validated through laboratory testing	The technology has been tested with a limited set of concepts and attributes, demonstrating its ability to express various situations and translate between Estonian and SNOMED CT post-coordinated expressions. (Using SNOMED CT as a Semantic Model for Controlled Natural Language Guided Capture of Clinical Data)

Tellija esindaja on liitunud arendusmeeskonnaga	**A client representative has joined the development team**	**There is no client and thus no client representative. (Interview)**
Klient osaleb nõuete koostamises	**The client actively participates in requirements development**	**There is no client yet at this stage. (Interview)**
Tehnoloogiaüleseid mõjusid (kui neid on) on hakatud tuvastama	Potential cross-technology impacts have been identified and are being investigated	Basic overview of compatibility requirements with existing systems and data formats is known, but not thoroughly researched or documented. (Interview)
Lauauuringud näitavad, et süsteemi komponendid peaksid töötama koos	Research suggests that the system components should be compatible and work together effectively	Prototypes already exist demonstrating the co-functioning of the components. (Digital health data capture with controlled natural language)
Klient tuvastab kasutuse võimaluse/mooduse	**The client has identified potential use cases and modes of operation for the technology**	**There is no client yet at this stage. (Interview)**
Mõõdikud on loodud	Performance metrics have been established to evaluate the software	Yes. Primary metric is number of supported medical terms.
Alustatud on skaleerimist	Initial steps for scaling the technology have been taken	Yes. It is also known known that increasing support for supported medical terms scales linearly.

Väikeste tüüpiliste andmekogumitega on tehtud katsed	Testing has been performed with small, representative data sets	Yes. (Feasibility of maternity record standardization on the example of midwives' free text entries)
Algoritmid töötavad laboratoorses keskkonnas, asendusprotsessoril	Algorithms have been successfully tested in a laboratory environment using a stand-in processor	Yes, all work so far has been conducted in a controlled environment. (Interview)
Olemas on teadmine, milline tarkvara on praegu saadaval ja mis täidab sarnast ülesannet	An understanding of available software solutions with similar functionality has been established	Paper mentions related works and positions its approach in the bigger picture of Explainable Artifical Intelligence. (Digital health data capture with controlled natural language)
Olemasolevat tarkvara uuriti võimaliku taaskasutuse osas.	Existing software has been examined for potential reuse within the project	Yes. While some solutions exist for converting text to Snomed CT, they are proprietary. (Digital health data capture with controlled natural language)
Olemas on teadmised praegu saadaval oleva tarkvara piirangutest *(praeguse tarkvara analüüs on lõpule viidud)*	The limitations of currently available software have been analyzed and documented	Yes. 1. Clinithink's CLiX CNLP with Recognosco3 recognizes phrases in a text and converts them to Snomed CT expressions – but due to machine learning, it lacks systematic overview of which phrases and to what they map. 2. CanvasMedical's narrative charting solution inserts form templates for diagnosis or lab orders in the text, but these forms are not linked with the rest of the text, breaking the principle of "insert once, use multiple times."

		3. Solutions based on information extraction and other general NLP-based approaches are biased towards the extractor's needs instead of the data capturer/writer, missing the point of an ever-increasing need for more data to be captured. 4. Machine and deep learning approaches risk failing because the texts used for training reflect an episode-based healthcare process, while there is a societal need to change healthcare to a continuous and preventive approach.
Teaduslik teostatavus on täielikult tõestatud.	The scientific feasibility of the project has been fully demonstrated	Yes. (Using SNOMED CT as a Semantic Model for Controlled Natural Language Guided Capture of Clinical Data, Digital health data capture with controlled natural language)
Praeguse tehnikataseme analüüs näitab, et tehnoloogia täidab vajaduse	An analysis of the current state of the technology confirms that it addresses the identified need	Yes. “Midwives are ready to use modern systems and it can be concluded that the standardization of the maternity record with programming language Grammatical Framework and The Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT) is feasible but needs further research and analysis.” (Feasibility of maternity record standardization on the example of midwives’ free text entries)
Kindlaks on tehtud riskide maandamise strateegiad	**Risk mitigation strategies have been developed and implemented**	**No. Existing work has been research-oriented, not product-oriented, thus risk management is not deemed necessary.**
Esitatud on parim võimalik lahendus, ilma majandusliku kalulatsioonita	The optimal solution has been presented without	Yes. (Interview)

	considering economic factors	

Assessment (TRL3)

Requirement (estonian)	**Requirement (english, translated)**	**Assessment**
Tehnoloogiaülesed probleemid (kui neid on) on täielikult tuvastatud.	All potential cross-technology issues have been identified and addressed	In progress. Compatibility requirements with existing systems and data formats in a single healthcare facility in Estonia have been defined. Some limitations arising from law are known. (Interview)
Algab süsteemi ametliku arhitektuuri väljatöötamine.	Development of the official system architecture has commenced	No. (Interview)
Klient koostab/kinnitab nõuete dokumendi.	The client has prepared/approved the requirements document	No, there is no client yet at this stage. (Interview)
Üldised süsteeminõuded lõppkasutaja rakendustele on teada	General system requirements for end-user applications are defined	In progress. • System must be able to run Grammatical Framework in a web browser extension.

		(Interview)
Süsteemi toimivusmõõdikud on loodud	Key performance indicators have been established for the system	In progress Usablity research in progress. (Bachelor's thesis Figma) Primary metric is the number of supported medical terms. (Interview)
Analüüs annab üksikasjalikke teadmisi konkreetsetest funktsioonidest, mida tarkvara peab täitma.	In-depth analysis provides a detailed understanding of specific software functions	No, there is no client yet at this stage. (Interview) However, the conducted research and prototypes provide detailed knowledge of the specific functions that the software must perform.
Kehtestatakse süsteeminõuetest tulenevad laboratoorsed nõuded	Laboratory requirements based on system requirements have been defined	Not yet, as the official system architecture is being developed. (Interview)
Loodud on nõuded igale funktsioonile	Requirements have been developed for each function of the software	Not yet, as the official system architecture is being developed. (Interview)
Olemas on pseudokoodiks teisendatud algoritmid	Algorithms have been translated into pseudocode	Yes, based on the research and prototypes. (Feasibility of maternity record standardization on the example of midwives' free text entries)
Andmenõuete ja vormingute analüüs on lõpule viidud	Data requirements and formats have been analyzed	Not yet, further analysis is required. (Interview)

Eraldi moodulid järgivad süsteemi esialgset arhitektuuriplaani	Individual modules adhere to the preliminary system architecture plan	Yes, based on the prototypes. (Interview)
Kavandid on kontrollitud ametliku kontrolliprotsessi kaudu	Designs have undergone formal review and approval processes	Not yet, as the official system architecture is being developed. (Interview)
Kehtestatud on teadus- ja tehnoloogiaarendusest väljajäämise kriteeriumid.	Exit criteria for research and technology development phases have been established	No. (Interview)
Tehnoloogia demonstreerib põhifunktsionaalsust lihtsustatud keskkonnas	The technology demonstrates basic functionality within a simplified environment	Yes, based on the prototypes and limited tests conducted. (Digital health data capture with controlled natural language)
Osatakse hinnata tarkvara suurust koodiridade ja / või funktsioonipunktide kaupa	Software size estimation is possible using lines of code and/or function points	Yes. Number of supported medical terms scales linearly with codebase. (Interview)
Ideekavandid on dokumenteeritud	Conceptual designs have been documented and refined	Yes, in the research papers and prototypes. (All of existing documentation)
Funktsionaalsus on kontrollitud laboratoorses keskkonnas	Software functionality has been validated in a laboratory environment	Yes, all work so far has been conducted in a controlled environment. (Interview)

Esialgsed kulutegurid on tuvastatud.	Initial cost factors have been identified and analyzed	Yes. Fixed costs: Number of supported medical terms scales linearly with codebase. Variable costs: Software will be run on the user's local machine, thus per-user cost is very low. (Interview)
Tehtud on katsed täisskaala probleemide ja representatiivsete andmekogumitega	Tests have been conducted using full-scale problems and representative data sets	No. (Interview)
Alustatud on integratsiooniuuringutega	Integration studies have commenced	Yes. (Interview)
Kulueesmärgid on seatud (CAIV)	Cost objective have been set (CAIV)	No. (Interview)
Laboratoorses keskkonnas on demonstreeritud üksikuid funktsioone või mooduleid	Individual functions or modules have been successfully demonstrated in a laboratory setting	Yes, the functionality has been tested in a controlled environment with prototypes and limited tests. (Digital health data capture with controlled natural language)
Skaleerimisdokumendid ja tehnoloogia skeemid on valminud	Scaling documentation and technology schematics have been completed	Not yet, as the official system architecture is still being developed. (Interview)

Teatud funktsioonide või moodulite katseline integreerimine näitab nende koostööd	Preliminary integration of specific functions or modules indicates successful collaboration	Yes, based on the prototypes that have demonstrated the co-functioning of components. (Interview)
Süsteemitehnika üldkavand on valmis (SEMP)	The System Engineering Master Plan (SEMP) has been finalized	Not yet, as the official system architecture is still being developed. (Interview)
Madala täpsusega tehnoloogia „süsteemi" integreerimine ja inseneritöö on lõpule viidud laborikeskkonnas	Initial, low-fidelity technology integration and engineering have been completed within a laboratory environment	Not yet, as the official system architecture and full-scale integration are still being developed. (Interview)
Klient kohustub süsteemi kasutusele võtuks läbi ATD kasutuselevõtu ja / või MOU	The client commits to system deployment via an ATD and/or MOU	No, there is no client yet at this stage. (Interview)
Integreeritud tootemeeskond (IPT) on ametlikult loodud	An Integrated Product Team (IPT) has been officially founded	Not yet, as there is no client and product-oriented development has not begun. (Interview)
Kliendi esindaja on IPT liige	A client representative serves as an active member of the IPT	Not applicable, as there is no client yet at this stage. (Interview)
Alustatud on ametlikku riskijuhtimise programmi	A formal risk management program has been initiated	No, existing work has been research-oriented, not product-oriented, thus risk management is not deemed necessary at this stage. (Interview)

Tehti esialgne rikete režiimi ja mõjude analüüs (FMEA) või riskikuse analüüs	A preliminary Failure Mode and Effects Analysis (FMEA) or risk analysis has been conducted	No, as the focus has been on research and development, not on product-oriented aspects such as risk analysis. (Interview)
Tehnoloogia kättesaadavuse kuupäevad on kindlaks määratud	Availability dates have been determined and agreed upon	No, as the focus has been on research and development, not on product-oriented aspects such as technology availability dates. (Interview)