MASTER

Determining Platform Openness

A Value Proposition Driven Methodology to Support Openness Trade-off Decision Making for Digital Platform Design

Mulders, J.A.C.

Award date:
2022

Eindhoven University of Technology

Department of Industrial Engineering and Innovation Sciences

Operation, Management and Logistics, Information systems Research Group

Master Thesis

Determining Platform Openness: A Value Proposition Driven Methodology to Support Openness Trade-off Decision Making for Digital Platform Design

J.A.C. (Jori) Mulders

0908466

Supervisors

Dr. B. (Baris) Ozkan	Eindhoven University of Technology
Dr. A. (Alexia) Athanasopoulou	Eindhoven University of Technology
Niek Versteegde	GOAL3

Eindhoven, 19/05/2022

Abstract

Designing a technically sound, socially sustainable and economically viable digital platform is a complex endeavor. Digital platforms enable value co-creation through the exchange of services between actors based on value propositions. Therefore, value proposition design is considered one of the first steps in platform design. Furthermore, a platform's openness affects the ability of an actor to create and accept new value propositions. Therefore, openness is also an important design aspect that needs to be taken into account in early platform design stages. However, knowledge on the practical execution of processes preceding platform launch is scarce and existing literature only discussed these design aspects individually. Considering these challenges, this research's objective was to develop a platform design methodology that is value proposition driven and is able to determine a corresponding degree of platform openness. To do so, a design science research methodology was followed combined with Situational Method Engineering (SME) to extend the Value Proposition driven Business Service Identification Method (VP-BSIM) with an adapted non-functional requirement (NFR) approach. The designed method was demonstrated for a healthcare start-up and evaluated through a focus group and questionnaire. The method was received with mixed results and requires additional design science research cycles for improvement. The contribution of this research is an integrated method that provides guidance on platform design decisions based on its value proposition design and desired degree of openness.

Summary

From a service dominant logic (S-D) logic perspective, digital platforms facilitate the interaction of actors through value propositions (i.e. offerings) embodied by service exchanges. This makes value propositions key determinants of the success and level of engagement between actors on digital platforms (Chandler & Lusch, 2015). Value proposition design is seen as a systematic search process that actors in a service (eco)system (e.g. platform owners, third-party developers) can perform to improve existing offerings, create new offerings, and reconfigure the ecosystem (Maglio & Spohrer, 2013). Therefore, value proposition design is considered one of the first steps in platform design (Saarikko, 2016; Tura, Kutvonen, & Ritala, 2018). While traditional companies often use a tight coupling with other actors in the form of strategic partnerships to co-create value (Steensma, Kevin; Corley, 2000), owners of digital platforms often have a loosely coupled approach to integrating different actors into the service ecosystem (Ghazawneh & Henfridsson, 2013). This loose coupled approach can be understood as dictated by a platform's openness. Openness has been defined as the level of restrictions on using, developing or commercializing functionality of a platform (West, 2003). The degree of platform openness affects the ability of an actor in a service (eco)system to create and accept new offerings (i.e. value propositions) (Lusch & Nambisan, 2015). However, deciding on the appropriate degree of platform openness by platform owners, is not a simple choice between open or closed, but rather it involves a complex set of decisions (Broekhuizen et al., 2021). This is often reflected in various trade-offs that need to be balanced by platform owners. The research question that follows from this is: *How to determine the appropriate degree of platform openness in value proposition driven digital platform design?*

An answer to this question is by applying specific tools and techniques which help practitioners in their analysis and decision making with regard to these key design aspects. However, there are few artifacts that help guide academics or practitioners to create transparency about the design of a digital platform (De Reuver et al., 2018; Hein et al., 2018). Furthermore, methods and approaches that have been proposed, only treat the design aspects individually. There is a lack of an integrated method, hence this research has the objective: *To develop a digital platform design methodology that is value proposition driven and supports the determination of the appropriate degree of platform openness by making trade-offs explicit.*

To achieve this objective, the service (eco)system perspective based on S-D logic was used and a design science research methodology was followed (Peffers et al., 2007) consisting of six steps: Identification of the research problem; definition of design objectives; design and development; demonstration; evaluation; and communication.

Design and development

For the design and development phase of the design science research, the following design objectives (DO) were identified; The method should be value proposition driven (DO1), the method should define explicit trade-offs related to a certain degree of openness (DO2) and the method should be useful, easy to use, and encourage intention to use the method (DO3). To develop the artifact, Situational Method Engineering (SME) was chosen. More specially the extension strategy was applied to extend the VP-BSIM (Adali, et al., 2021). By performing a literature review, the method by Sadi & Yu, (2017ab) was identified to help determine the appropriate degree of platform openness. However, the method by Sadi & Yu, (2017ab) required (functional) requirements as input. Therefore, an additional step had to be introduced to translate the business services from the VP-BSIM. To do this, a heuristic was designed by the author of this research which derives software services from the business services via a few guidelines. Software services support the execution of a business service with information technology and expose functionalities that can be reused and composed based on business needs (Kohlborn, et al., 2009). The software services were defined and treated as product level requirements (i.e. goals) (Gorschek & Wohlin, 2006).

The final designed artifact is a method consisting of eight main steps and an input step. In the input step, the co-created value of the service ecosystem is determined by applying the Service Dominant Business Model Radar (SDBM/R) from Turetken et al. (2019). This is used as input for the VP-BSIM. The first three main steps follow from the VP-BSIM. In the first main step (1), the goals that actors pursue in the service ecosystem and the dependencies between actors to achieve these goals are determined via Strategic Dependency (SD) and Strategic Rationale (SR) models. In the second main step (2), business capabilities required to enable the goals and tasks actors pursue in the service ecosystem are defined. The SR model is translated into the Service Domain – Business Capability Matrix and (future) business capabilities of actors are matched to service domains and service operations (i.e. tasks) performed in the service ecosystem. In the third main step (3), coherent modular business services are composed of the business capabilities and service operations. A service analysis is performed combined with a feature binding analysis to ensure the business services are modular.

The fourth main step (4) is the designed heuristic based on Kohlborn et al. (2009) to determine software services which form design requirements of the digital platform. To do so, each service operation of the identified business services is examined and determined if they could be supported by information technology. This is followed by a step to determine if the software service is required for openness towards an actor using the platform (e.g. third-party developers).

The fifth up and including the eight main step of the designed method follow from Sadi & Yu (2017ab). In the fifth main step (5), non-functional openness requirements are determined for a single software service and depicted in a goal model (i.e. interdependency graph). In the sixth main step (6), multiple design alternatives are determined which are able to implement a software service (i.e. requirement). Each identified design alternative needs to be characterizable by a different degree of openness. In the seventh main step (7), the design alternatives are evaluated by assessing if they meet the non-functional openness requirements determined in the fifth main step. For this step the goal model evaluation procedure from Horkoff & Yu (2009) is applied by Sadi & Yu (2017ab). In the eighth and last main step (8), the scores on the most critical non-functional openness requirements are compared between the design alternatives to reason about the openness trade-offs. The chosen design alternative implements the software service (i.e. requirement) with a degree of openness judged to be appropriate by the platform owner.

Demonstration

The aim of this step in the design science research method by (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007) was to demonstrate the use of the designed artifact to solve one instance of the problem. An organization was selected for the demonstration case based on two reasons: their desire to develop an open digital platform and their business domain of healthcare and low resource setting (i.e. low- and middle-income country setting). This business context brought many risks regarding privacy and security of exchanged patient data which needed to be balanced with an appropriate degree of platform openness. Therefore, the entirety of the designed method could be demonstrated for the chosen organization.

After the selection of the appropriate organization, the demonstration case was performed by the author of this research with the help of three input sessions from the chosen organization. The scope of the demonstration was limited to one iteration of the designed method. This consisted of the creation and selection of one SDBM/R, generating multiple software services then working out the non-functional openness requirements and (evaluated) design alternatives for one selected software service. The output of the demonstration was one selected design alternative that was judged to best balance the openness trade-offs when implementing one software service that was required for opening up the digital platform.

Evaluation

The empirical evaluation of the designed method consisted of a focus group session with practitioners and a short questionnaire. The goal of the evaluation was to assess the third design objective which used the design evaluation criteria of the technology acceptance model (TAM) (Davis, et al., 1989); perceived usefulness, ease of use and intention to use. The focus group was held online with participants from different roles within the demonstration case company. This company functioned as the focal organization actor in the service ecosystem and would be the main user of the method. The selection of the five participants was done such that all skills deemed necessary to execute the designed method were represented. This ranged from business developers to (software) requirement engineers. After the empirical evaluation, all design objectives were reflected on to assess their fulfilment. The first two design objectives were self-assessed based on the designed method construction and relevant related comments made during the empirical evaluation.

Results from the empirical evaluation were mixed. Participants of the focus group recognized the value of the designed method, but also agreed that the method contained too many steps and was too complicated in practice. Some participants were intimidated by the complexity, terminology and tools, seeing it as a hurdle for use. However a notable strength was the ability of the method to bring the business and technical aspects of platform design together. Therefore, when assessing the third design objective it can be considered only partially fulfilled. Perceived usefulness scored 3,4 on the five-point Likert scale, while Ease of Use and Intention to Use scored 3,1 and 3,2 respectively.

Three improvement directions were identified based on the empirical evaluation and assessment of the design objectives. The first improvement consists of redesigning step 4 of the method. The demonstration and empirical evaluation showed that it was either not clear enough which activities of the service ecosystem could be enabled by a digital platform and/or how these platform design elements formed a coherence. Improvements without entirely recreating this step is to clarify the description of the service identification template accompanied by an improved application description. Alternatively another sub step could be added in which users create consensus on which digitalized or automated service operations could form a coherent platform. For the second improvement, another step needs to be added after step 8. Participants of the focus group noted that decomposing the platform and making openness decisions on individual software services would give a different result compared to considering all design decisions as a whole. This would for example show conflicts between individual decisions. The additional step would focus on identifying compatibility between design alternatives of multiple software services on both a high level and individual non-functional requirements. The third improvement aimed to reduce the complexity of the method. A suggestion was to improve the method-overview to show which elements of the method could be executed in isolation and under which conditions. Another suggestion was to make the relationship between method-steps and the eventual openness trade-off decision making more explicit.

Conclusion

Through extending the VP-BSIM with the method proposed by Sadi & Yu (2017ab) this research provided an integrated method combining the key platform design aspects of value proposition design and openness, which to the best of the author of this research's knowledge did not yet exist. Other research contributions are the additional validation for the method by Sadi & Yu (2017ab), exposing their method to non-embedded platforms and applying the method to a higher level of abstraction (product level requirements instead of functional requirements). Finally, the designed method and demonstration contributes to the creation of design knowledge for business-to-business platforms, which so far have been under researched. Practical implications from the designed method are the ability for practitioners to systematically reason about an appropriate degree of platform openness starting a platform from scratch. Another practical implication is the bridging of business and technical design aspects, which may result in a more appropriate degree of openness. Furthermore, this research showed that decomposition of platform design elements is a viable option for determining platform openness. However, it remains a complex qualitative task requiring human judgement.

The main limitations of this research stem from the choice to make the author of this research perform the demonstration (1), the scope of the demonstration (2) and the selection of participants of the focus group (3). First, the author of this research had limited experience with the demonstration application domain and business context, making this a validity threat. Secondly, the designed method was only demonstrated for one software service. Third, the selected participants had limited experience with a platform design process and the concepts of the designed method, posing a validity threat to the empirical evaluation results. Future research could focus on applying the method to different actors in the service ecosystem or to place the designed method in the wider openness discussion regarding governance and non-technical platform design elements or focus on the three proposed improvement directions.

Preface

This Master thesis is submitted for the Master program 'Operations Management & Logistics' at the Eindhoven University of Technology (TU/e) and was performed for the Information systems Group at the faculty of Industrial Engineering and Innovation Sciences. This document is the result of a personal journey. It is the result of months of labor, studying countless papers and numerous discussions, but above all it is the result of the will to graduate. Decisions I made during this research have not led to a straightforward process. However, without these decisions, this research would never have gotten to its conclusion. I would like to use this opportunity to thank those who have helped me face this challenge. First, I would like to thank my university supervisors Dr. B. Ozkan and Dr. A. Athanasopoulou. Baris, you were assigned to me and we had a long lead up to commence this research, however our relationship took a positive turn when we initiated this subject. Thanks for your trust and confidence in me on the moment I wanted to give up and your relentless feedback and guidance. I am glad to have been supervised by you. Alexia, after your step into motherhood we managed to reunite, with you as my second supervisor. Your insightful comments and questions always knew to compliment those from Baris. To Niek, my company supervisor, thanks for your never-ending trust in a positive conclusion, willingness to go to any length for me and care for my mental health. Your vision at GOAL 3 will continue to inspire people. To my other colleagues at GOAL 3, thanks for the fantastic workplace atmosphere. I am glad to have met, you deserve all the success. To Fleur, my girlfriend, when I was on my lowest and could not see any path forward, you were there. Always urging me to persevere and reminding me of the feeling of fulfilment by achieving this milestone. To Remco, my brother, thanks for giving me advice on how to change my course, encouraging me and always knowing how to put it all in perspective. To my parents who I tried to avoid sharing the progress of my research with, thanks for giving me the freedom to pursue my own path without questioning and providing the change of scenery when I needed it. To Martijn, thanks for being a catalyst to transforming my mindset. Thanks to Casper, for the coffee breaks in which we could share our frustrations. Thanks to Ron, for dragging me out to the university campus. Finally, my apologies to all neglected friendships.

Jori Mulders

List of Figures

List of Tables

List of Abbreviations

API = Application Programmable Interface

BS = Business Service

DO = Design Objective

DSR = Design Science Research

GORE = Goal Oriented Requirements Engineering

SD = Strategic Dependency

S-D Logic = Service Dominant Logic

SDBM/R = Service Dominant Business Model Radar

SR = Strategic Rationale

SS = Software Service

TAM = Technology Acceptance Model

VP-BSIM = Value Proposition Business Service Identification Method

Contents

1 Introduction

The rise of digital platforms has had an enormous impact on society. For consumers they have changed the way we eat, how we travel and where we sleep. Along the way, digital platforms have held a central position in the business models of the largest companies in the world. The decision of companies to develop and operate platforms has changed the status quo of old industries and created new markets (De Reuver et al., 2018). The world health organization has even identified digital platforms as a key technology to realize its sustainable development goals (WHO, 2021).

Digital platforms have been defined from several research perspectives which complement each other. Gawer (2014) combined the economic perspective, which identifies platforms as multi-sided markets, and technical perspective, recognizing platforms as technical architectures, by defining digital platforms as a product, service or technology that acts as a foundation upon which external innovators, often organized arranged in a business ecosystem, can develop their own complementary products, technologies or services. Actors in this business ecosystem have various roles, such as the role of platform owner, who controls the platform, and third-party developers (i.e. external innovators) who contribute to the digital platform, but also the customer, who eventually consumes the products or services provided by the digital platform.

Digital platforms have also been described as service platforms. Lusch & Nambisan (2015) define a service platform as technology that enables the integration of customers, partners and other actors for value co-creation that moves beyond individual companies. The term service platform has been used by literature viewing digital platforms from a service-dominant logic (S-D logic) perspective. S-D Logic explains the logic of offering services to a customer instead of goods (Vargo & Lusch, 2004). Vargo and Lusch (2004) define a service as the application of specialized competences (knowledge and skills) through activities, processes, and performances for the benefit of another actor or the actor itself. According to Lusch & Nambisan (2015), these actors are loosely coupled in a network or community that is a self-adjusted, self-contained system, also described as a service ecosystem. This service ecosystem needs to provide an architecture which facilitates the interaction of resources and actors, this is described as the service platform. Co-creation of value then refers to the process of value creation through resource integration and service exchanges between actors within a service ecosystem enabled by a service platform (Lusch & Nambisan, 2015). In this value creation process, actors in the service ecosystem offer, accept or reject value propositions, embodied by services, based on their competences and capabilities (Lusch & Nambisan, 2015). In other words, value propositions are seen as invitations from actors to one another to engage in a service (Chandler & Lusch, 2015). Service platforms support actors with the construction and offering of value propositions, while also facilitating the search for and identification of appropriate value propositions (Lusch & Nambisan, 2015).

While the introduction of digital platform's has enabled actors access to innovate and an increasing pace of (service) innovation through co-creation (Bonina et al., 2021; Lusch & Nambisan, 2015), the design of a digital platform that is technically sound, socially sustainable and economically viable is a complex endeavor. Digital platform design has been described as the configuration of specific design aspects when building a new digital platform (Tura et al., 2018). Designing a platform is considered complex because it involves various interrelated aspects ranging from its business model (Fehrer, Woratschek, & Brodie, 2018), to its technical design that includes aspects such as determining the architecture and technologies of the platform, including interfaces to other actors (Tiwana, 2015), to activities related to governance that regulate interactions on a platform (Bresnahan & Greenstein, 2014).

1.1 Problem Definition and Research Objective

Two key design aspects of a digital platform are its value proposition design and degree of platform openness. Value propositions are deemed key determinants of the success and level of engagements between actors on digital platforms (Chandler & Lusch, 2015). Value proposition design is seen as a systematic search process that actors in an ecosystem (e.g. platform owners, third-party developers) can perform to improve existing offerings, create new offerings, and reconfigure the ecosystem (Maglio & Spohrer, 2013). Therefore, value proposition design is considered one of the first steps in platform design (Saarikko, 2016; Tura et al., 2018). Value proposition design is integral to platform business models (Fehrer et al., 2018). Although the business model requires a balancing act between value co-creation and value capture (i.e. internalized positive externalities generated by each actor's value creation) for economic viability of the digital platform (Amit & Zott, 2015), value creation is considered a pre-requisite for value capture (Storbacka, 2011).

Openness has been defined as the level of restrictions on using, developing or commercializing functionality of a platform (West, 2003). While traditional companies often use a tight coupling with other actors in the form of strategic partnerships to co-create value (Steensma, Kevin; Corley, 2000), owners of digital platforms often have a loosely coupled approach to integrating different actors into the service ecosystem (Ghazawneh & Henfridsson, 2013). This is most noticeable at the boundaries between the digital platform and its service ecosystem (Hein et al., 2019).

Designing for openness is considered an important task for digital platform owners, especially when the purpose of a digital platform is to maximize the opportunities for (service) innovation (Lusch & Nambisan, 2015). Openness decisions then affect the ability of an actor in a service (eco)system to create and accept new offerings (i.e. value propositions) (Lusch & Nambisan, 2015). Platform openness can be provided, for example, via platform boundary resources (Ghazawneh & Henfridsson, 2013). These boundary resources are the organizational arrangements between actors such as entrance or exit (i.e. access) rules and the technologies such as API's that enable third parties to interface with a platform. However, deciding on the appropriate degree of platform openness by practitioners, is not a simple choice between open vs closed as described earlier, but rather it involves a complex set of decisions (Broekhuizen et al., 2021). This is often reflected in various trade-offs that need to be balanced by platform owners such as platform adoption vs appropriability (i.e. value capture) (Parker, Alstyne, & Jiang, 2017) and diversity of the platform's offerings vs platform control (Boudreau, 2010).

The research question that follows from this is: *How to determine the appropriate degree of platform openness in value proposition driven digital platform design?*

An answer to this question is by applying specific tools and techniques which help practitioners in their analysis and decision making with regard to these key design aspects. However, literature related to the question of 'how to design' aspects of a digital platform such as its value propositions and degree of openness is relatively scarce (De Reuver et al., 2018). There is limited design knowledge on the practical execution of processes preceding the launch of a platform (Otto & Jarke, 2019; Tura et al., 2018) and there are few artifacts that help guide academics or practitioners to create transparency about the design of a digital platform (Hein et al., 2018). Existing methods and approaches that have been proposed, only treat design aspects individually. There is a lack of an integrated method, hence this research has the objective:

To develop a digital platform design methodology that is value proposition driven and supports the determination of the appropriate degree of platform openness by making trade-offs explicit

To achieve objective, the service (eco)system perspective based on S-D logic was applied and a design science research methodology was followed (Peffers et al., 2007). During the design and development phase, the VP-BSIM (Adali, et al., 2021) was extended with the adapted NFR approach (Sadi & Yu, 2017ab) by adopting a situational method engineering method (Ralyté, Deneckère, & Rolland, 2003).

The value proposition business service identification method (VP-BSIM) by Adali et al. (2021) uses the input of the SDBM/R (Turetken et al., 2019) to translate value propositions of actors in a service ecosystem into modular business services, bringing value propositions closer to realization. However, the output of this method needs to be translated to platform requirements. Sadi & Yu (2017ab) developed a method which defines openness requirements as a distinct class of non-functional requirements (NFR) and used these as criteria to evaluate trade-offs and select the appropriate degree of openness via design alternatives that could implement functional requirements needed to open up a digital platform.

1.2 Thesis Report Structure

This research report has been divided into several chapters, following the structure of the design science research methodology. Chapter 2 presents a literature review which introduces the theoretical background and related literature. In Chapter 3 the research design is presented. In Chapter 4 the final refined designed artefact is presented. Chapter 5 presents the demonstration of the artefact in a demonstration case in the healthcare domain. The evaluation and refinement of the artefact is presented in Chapter 6. Finally, Chapter 7 concludes this research and describes research implications and future work.

2 Conceptual Background and Related Work

This chapter is the result of a narrative literature review. The first few sections present the main concepts and their dimensions used in this research. Section 2.8 presents related work. Section 2.9 concludes the review with a short discussion of the identified frameworks, methods and approaches and the identified research gap which led to this research.

2.1 Conceptualizing Digital Platforms and Ecosystems

Digital platforms share three basic characteristics: they are technically mediated, enable interaction between users groups and allow those user groups to carry out defined tasks (De Reuver et al., 2018). However, definitions depend on the research perspective from which they are studied. Several typologies have been introduced to define digital platforms. For example, digital platforms have been characterized as internal or external (i.e. industry) platforms (Gawer & Cusumano, 2014). Internal platforms have been defined as a set of assets organized in a common structure from which a company can efficiently develop a stream of derivative products (Gawer & Cusumano, 2014). External (industry) platforms are defined as a product, service or technology that acts as a foundation upon which external innovators, often organized arranged in a business ecosystem, can develop their own complementary products, technologies or services (Gawer & Cusumano, 2014). This definition combined the economic perspective, which identifies platforms as multi-sided markets, and technical perspective, recognizing platforms as technical architectures. Cusumano, Gawer & Yoffie (2019) defined platforms according to their main purpose and identified two broad categories: transaction and innovation oriented. The main purpose of transaction-oriented platforms is to facilitate transactions between different organizations, entities and individuals, while innovation-oriented platforms are defined similarly as external (i.e. industry) platforms. This literature review is primarily concerned with external platforms with an innovation purpose.

In line with these definitions, Poniatowski, Lüttenberg, Beverungen, & Kundisch (2021) build on, among others, Tiwana, Konsynski, & Bush (2010) and conceptualized digital platforms as consisting of three parts: A platform periphery, a platform core and a platform infrastructure. This is depicted in Figure 1. According to Poniatowski et al. (2021), the platform infrastructure is used as a foundation of the platform core. The core can be described as the part managed by a platform owner and with which third parties can interact. Furthermore, the platform owner controls the platform periphery, which consists of the contributions provided by third parties often based on or realized by functionalities supplied by the platform owner.

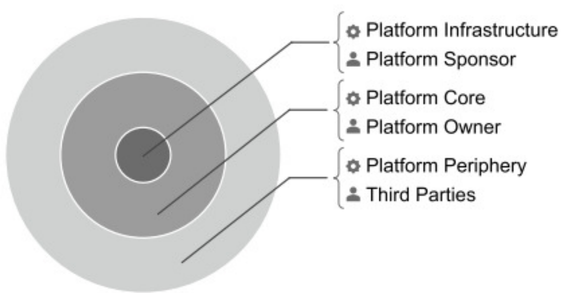


Figure 1: Conceptualization of a digital platform in three parts (Poniatowski et al. 2021)

Research on digital platforms has been conducted from many disciplines, for example, information systems, management and economics (De Reuver et al., 2018). However, boundary spanning research is scarce (Gawer, 2014). Therefore, Poniatowski, Lüttenberg, Beverungen, & Kundisch (2021) introduced a framework to provide a conceptual point of reference for research on digital platforms. They aimed to introduce a systematic, integrated perspective on digital platforms to reveal the interactions and contradictions involved in investigating platforms by providing a comprehensive structured overview of concepts from management, economics and information systems research. The only other literature reviews attempting a broader perspective on digital platforms and ecosystems come from De Reuver et al. (2018) and Schreieck, Krcmar, & Wiesche (2016), but neither puts forward a model to structure and explain digital platforms and their ecosystems. The framework introduced by Poniatowski et al. (2021), depicted in Figure 2, introduces three layers of abstraction; Conceptualizing platforms as information systems, as systems for actor engagement or as ecosystems. Therefore, the framework conceptualizes digital platforms as nested hierarchies of systems that are shaped by, and in interaction, with their environment.

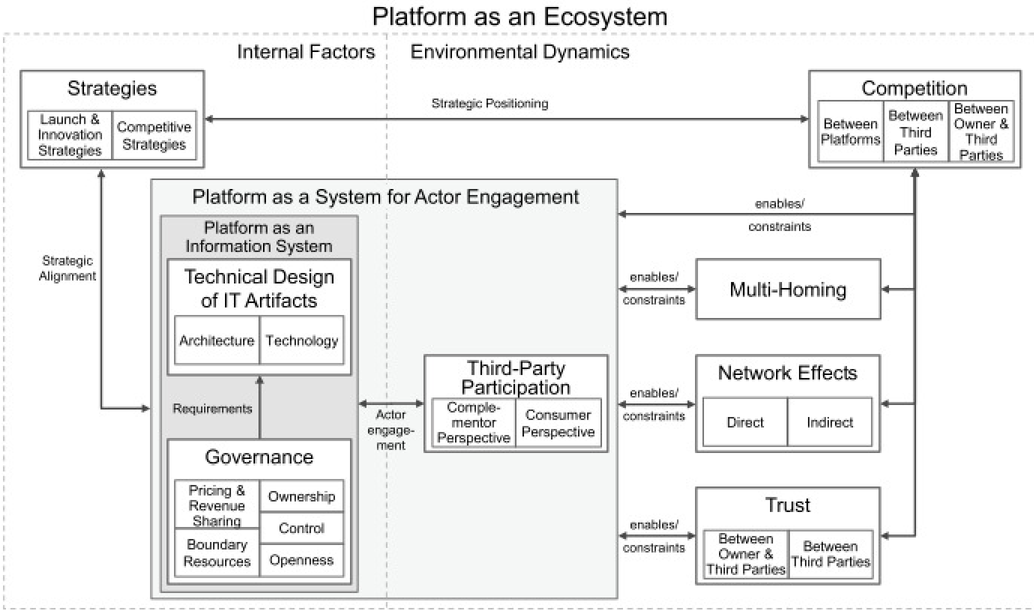


Figure 2: Three layers of abstraction for theorizing digital platforms (Poniatowski et al. 2021)

Conceptualizing platforms as an information system, Poniatowski et al. (2021) consider platforms to refer to the technical design of IT artefacts and their governance. This views platforms as an IT artefact that needs to be designed and managed by an organization. Conceptualizing platforms as a system for actor engagement, third parties engage in the value proposition offered by a platform owner by using the information system to interact and co-create value, by offering content or consuming content. This conceptualization is similar in abstraction to the concept of service ecosystems (Lusch & Nambisan, 2015). When conceptualizing platforms as an ecosystem, this expands the perspective from a single platform to including environmental dynamics that affect or are affected by the system for actor engagement. This perspective comprises the internal factors and environmental dynamics of platforms. Internal factors are controlled by a platform owner directly, consisting of strategies, and technical designs, among others. In contrast, environmental dynamics lie outside of a platform owners' direct control, resulting from performances of actors in the ecosystem. Strategies of a platform owner may adapt to changes in the ecosystem and might cause modifications of the platform as an information system.

2.2 Digital Platform Design

Digital platform design has been described as the configuration of specific design elements when building a new digital platform (Tura et al., 2018). This consists of both the technical design of IT artefacts and platform governance (Tiwana et al., 2010), which must be aligned with an overall platform strategy (Poniatowski et al., 2021). One of these 'platform launch strategies' is called coring (Gawer & Cusumano, 2015). Coring describes activities that identify elements that can be used to develop a new platform and is focused on defining the core of a platform such that it solves a systematic problem faced by potential users (Gawer & Cusumano, 2008). According to Gawer & Cusumano (2008), coring considers both technology and business-related aspects.

Technology related design aspects of a platform consider its functional architecture. This is depicted in Figure 3 for an innovation platform. Platforms have been described as modular architectures (Ulrich, 1995) consisting of a core and periphery (Baldwin & Woodard, 2009) that is often centrally governed by a platform authority (i.e. owner) (Ghazawneh & Henfridsson, 2013). The core contains modules which are accessible through interfaces by third-party developers (i.e. complementors) to innovate applications and services. It is possible that elements of the platform core are sourced from suppliers other than the platform authority (Bonina & Eaton, 2020) Transactional platforms might also provide functionality accessible via a more limited set of interfaces into other digital services that require its functionality (Bonina et al., 2021).

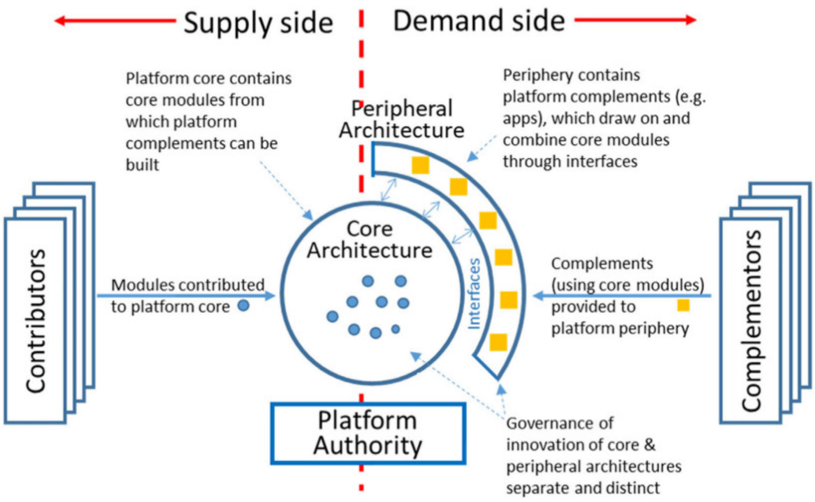


Figure 3: Overview of the functional architecture of an innovation platform (Bonina et al., 2021)

Business related design aspects of a platform consist various aspects. For example, platform governance, which was broadly defined to regard who decides what on the platform (Tiwana et al., 2010). This encompassed three facets: How decision rights are divided between platform owner and third-party developers, what type of formal and informal control mechanisms are used to align interests and incentive structures which relates to pricing and sharing of revenues.

Literature rarely adapts an integrated view of both business and technology platform aspects, although this is considered necessary to describe platforms (Basole, 2009; Schreieck et al., 2016). This is exemplified by Schreieck et al. (2016) with an application store. This can be considered a marketplace that matches demand for and supply of applications on mobile devices, while at the same time it is an artifact to co-create value on a technology platform, i.e. the operating system of mobile devices.

Although identifying platform design elements and other considerations during platform design is crucial, this does not yet offer guidance to practitioners on how to design and govern platforms and their ecosystems. However, the literature on the process of platform design is rather limited (De Reuver et al., 2018; Otto & Jarke, 2019; Saarikko, 2016). Existing research is either focused on a specific domain in which the platform operates or a certain platform ownership model (e.g. shared ownership). For example, Spagnoletti, Resca, & Lee (2015) created design theory for online communities as digital platforms. Two papers describing the platform design process based on case studies each followed a different process, although mostly containing similar platform design elements (Fürstenau, Auschra, Klein, & Gersch, 2018; Otto & Jarke, 2019). However, both papers describe the design process of a platform that has shared ownership (i.e. a joint-venture, alliance or consortium) instead of a single platform owner driving the design. Otto & Jarke (2019) consider these design processes to be different. For example, the platform architecture and general design consist of a more consensus-oriented design process determined by shared interests of the multiple owners when ownership is shared. Furthermore, single platform owner driven platforms have been shown to start with a limited number of platform actors with limited options to interact between them, gradually increasing, while the alliance driven platform studied, started from a complex ecosystem and then reduced to a core platform (Otto & Jarke, 2019).

Finally, the design of a digital platform is understood to be an evolutionary, dynamic and complex process (Otto & Jarke, 2019). The evolutionary lifecycle of a digital platform typically consists of three or four major phases: a first phase of platform design, a second phase of platform adoption by user groups, third phase of scaling and growth activities and a fourth phase of renewal to ensure continuation of the platform (owner's) existence (Henfridsson & Bygstad, 2013; Isckia, De Reuver, & Lescop, 2018). Therefore, platform design elements may continuously be redesigned to adjust for changes in the platform's ecosystem (Poniatowski et al., 2021).

2.3 Value Proposition Design and Digital Platforms

Although coring is recognized as a platform launch strategy, it offers limited practical insights into how a core is identified (Saarikko, 2016). Furthermore, coring is described as one party leveraging a solution in order to solicit support from potential adopters. It is questionable if this is viable for platforms that incorporate multiple actors. Therefore, Saarikko (2016) proposed to treat the process of platform coring as a form of co-creation of multiple actors. More specifically, Saarikko (2016) argued that since co-creation often relies on finding a shared perspective on what constitutes value, the coring process is not a (future) platform owner offering value per se, but rather offering value propositions that are accepted or rejected by receiving actors (Lusch & Nambisan, 2015). Therefore, value proposition design can be considered an important early step in digital platform design.

The underlying ideas from Saarikko (2016) originate from literature on service (eco)systems based on Service-Dominant (S-D) Logic. S-D Logic explains the logic of offering services to a customer instead of goods (Vargo & Lusch, 2004). Vargo and Lusch (2004) define a service as the application of specialized competences (knowledge and skills) through activities, processes, and performances for the benefit of another actor or the actor itself. The creation, accepting and offering of services occurs within service systems and service ecosystems. Service systems are complex socio-technical systems that enable collaborative value creation through value propositions, service exchange and resource integration processes (Böhmann et al., 2014). Service ecosystems widened this concept to focus more explicitly on actor-to-actor relationships that are continuously re-created for mutual value creation (Brozovic & Tregua, 2022)**.** Thus, Lusch & Nambisan (2015) defined a service ecosystem as a network or community of loosely coupled actors that is a self-adjusted and self-contained system. These service (eco)systems need to provide an architecture which facilitates the interaction of resources and actors, this is also described as the service platform (i.e. digital platform) (Lusch & Nambisan, 2015). Co-creation of value then refers to the process of value creation through resource integration and service exchanges between actors within a service ecosystem enabled by a service platform (Lusch & Nambisan, 2015).

Value propositions are deemed key determinants of the success and level of engagement between actors on a digital platform (Chandler & Lusch, 2015). Value propositions are seen as invitations from actors to one another to engage in a service enabled by a digital platform (Chandler & Lusch, 2015). From this perspective, value proposition design is considered a systematic search process that actors in an ecosystem (e.g. platform owners, third-party developers) can perform to improve existing offerings, create new offerings, and reconfigure the ecosystem (Maglio & Spohrer, 2013).

Value proposition design of digital platforms is part of a platform business model (Fehrer et al., 2018) and one of the first activities to determine in a business model (Chesbrough & Rosenbloom, 2007). To reflect the nature of digital platforms, Fehrer et al. (2018) proposed a new business model logic which highlights value processes focused on actor-to-actor service exchanges which challenges the idea of traditional business model logic which focused on firms controlling entire activity systems based on Porter's value chain. This new business model logic emphasizes value co-creation within open networks based on the idea of continuously emerging, non-hierarchical collaboration among various actors (Ketonen-Oksi et al., 2016). Therefore, Fehrer et al. (2018) introduced the logic of value co-creation in service (eco)systems based on S-D Logic into the value creation logic of platform business models.

2.4 Openness and Digital Platforms

West (2003) defined platform openness as the level of restrictions on using, developing or commercializing functionality of a digital platform. The importance of platform openness has been well documented in academic literature, mainly from a management perspective. Boudreau (2010) observed increased innovativeness as a result of openness, while other benefits such as increased end-user adoption (West, 2003) and potential network effects (Parker et al., 2017) were noted. Other authors such as Eisenmann et al., (2008) identified, among others, interoperability with established rival platforms, backward compatibility with prior platform generations, securing exclusive rights to certain third-party provider services or even absorbing third-party provider services into the core of a platform (i.e. enveloping) as drivers for openness. Van Alstyne, Parker and Choudary (2016) take it further and claim that platforms often fail because they do not optimize openness; too closed and potentially desirable actors don't participate, too open and poor-quality contributions or misbehavior of actors causes other actors to defect. From a service dominant view, In line with Lusch & Nambisan (2015), Thomas, Autio, & Gann (2014) argue that openness is important for platform success as it determines how well platforms can leverage their external users' resources to match their internal capabilities.

The degree of openness of a digital platform may vary on a continuum from private, proprietary internal and external digital platforms (Gawer & Cusumano, 2014) to free open source software (e.g. public service platforms) (Franco-bedoya, Ameller, Costal, & Franch, 2017). However, different degrees of openness can be found in practice even among platforms with similar purpose and ownership such as mobile platforms like iOS and Android (Benlian, Hilkert, & Hess, 2015), digital marketplaces (Ghazawneh & Henfridsson, 2015) and payment platforms (Ondrus, Gannamaneni, & Lyytinen, 2015). Furthermore, the degree of openness is not a static, fixed choice but is dynamic and may thus vary over time, shifting from closed to open or vice-versa (Eisenmann et al., 2008; Homscheid, Kilian, & Schaarschmidt, 2015).

Several authors have tried to measure the degree of openness of digital platforms. For example, Ondrus et al. (2015) examined openness at three levels: provider, technology and user level. The provider level recognizes the strategic involvement of various key stakeholders that provide a platform, the technology level involves interoperability of a platform with various technologies and the user level is concerned with determining to what extend a platform discriminates between different segments of a customer base. Eisenmann et al., (2008) introduced the notion of horizontal and vertical platform openness. Horizontal openness refers to allowing the users of a competitor's platform to interact with the platform or allowing third parties to participate in commercialization or technical development, while vertical openness refers to granting third-party developers to resources for developing complementary products or services. Furthermore, Eisenmann et al. (2008) defined platform openness on two dimensions: access (i.e.

who is allowed access to the platform) and authority (i.e. control, how much is the actor allowed to do on the platform). Broekhuizen et al. (2021) build on the work of Eisenmann et al. (2008) and defined five dimensions of platform openness; suppliers, customers, complementary service providers, categories and channels. Each dimension can have a certain degree of access to or authority on the platform. These together define a digital platform's 'signature'; the platform's image or fingerprint for openness as perceived by actors on the platform (Broekhuizen et al., 2021). Thus, platforms can exhibit varying degrees of openness from both a business perspective and its technical architecture.

Findings from Boudreau (2010), who characterize the relationship between innovation on a platform and openness as curvilinear, suggest openness can be optimized. However, deciding on the appropriate degree of platform openness by practitioners, is not a simple choice between open vs closed as described earlier, but rather it involves a complex set of decisions (Broekhuizen et al., 2021). This is often reflected in various trade-offs that need to be balanced by platform owners such as adoption vs appropriability (Parker et al., 2017) and diversity vs control (Boudreau, 2010). In the first trade-off, higher openness leads to adoption by third-party developers. However, it may also reduce switching costs (i.e. costs for platform users to switch to a competing platform) and therefore increase inter-platform competition, making it more difficult to generate profit. In the second trade-off, higher openness leads to a greater diversity of complementary applications through open innovation. However, a platform owner may lose control over the quality of applications and be faced with a more complex coordination of resources and strategic interests.

While the exemplified trade-offs describe strategic business concerns, literature is also concerned with mechanisms to implement openness in practice that relate to a platform's technical architecture. In particular, vertical openness (Eisenmann et al., 2008)) can be implemented through boundary resources (Setzke, David, Böhm, & Krcmar, 2019). Boundary resources are the tools and regulations that facilitate the arm's length relationship between actors on a digital platform (Ghazawneh & Henfridsson, 2013). Boundary resources can thus be technical and non-technical (Setzke, David et al., 2019). For example, technical boundary resources include application programming interfaces (API's) and software development kits (SDKs) for third-party developers to interface with the digital platform. Furthermore, the platform owner can use standards to ensure compatibility and interoperability between third-party developers (Tiwana, 2015). Non-technical boundary resources can be technical documentation and provided support (Setzke, David et al., 2019). Thus, through boundary resources, actors can interact in order to co-create value (Eaton, Elaluf-Calderwood, Sorensen, & Yoo, 2015). However, boundary resources may also restrict co-creation of value. For example, rigid regulations for the approval of third-party developer services on a platform may decrease a third-party's motivation (Eaton et al., 2015). This relates to governance mechanisms to implement openness which includes ownership of intellectual property rights and formal and informal control mechanisms to moderate behavior of third-party developers (Tiwana, 2015).

To summarize, West (2003) defined platform openness as the level of restrictions on using, developing or commercializing functionality of a digital platform. Platform openness is an important aspect to consider as it, among others, stimulates and restricts value co-creation among platform actors, such as third-party developers (Eaton et al., 2015). Therefore, openness (co-) determines platform success (Van Alstyne et al. 2016) and its market potential (Ondrus et al., 2015). Openness is considered a multidimensional concept (Benlian et al., 2015; Eisenmann et al., 2008), consisting of both managerial and technical aspects (Setzke, David et al., 2019). Furthermore, openness can be considered for multiple platform actor's (Broekhuizen et al., 2021; Eisenmann et al., 2008). Considering these dimensions may lead to a platform's openness signature (Broekhuizen et al., 2021) or perceived openness from individual actors, such as third-party developers (Benlian et al., 2015). Platform owners are able to control openness deliberately (Boudreau, 2010), both during the platform design and dynamically to govern the platform and it's ecosystem after its establishment (Eisenmann et al., 2008; Homscheid et al., 2015). Therefore, the concept of openness is often related to governance. Furthermore, findings from Boudreau (2010), who characterize the relationship between innovation on a platform and openness as curvilinear, suggest openness can be optimized. This is reflected in the various trade-offs that need to be balanced by platform owners. Openness can be implemented via mechanisms such as (technical) boundary resources, which could be considered platform design elements. Finally, all the mentioned openness considerations make determining the appropriate degree of platform openness a challenging task for practitioners and needs further research (Setzke, David et al., 2019).

2.5 Service Dominant Business Model Radar (SDBM/R)

Co-creation of value is considered the fundament of a service ecosystem. This ensures that a service ecosystem and a digital platform that enables (parts of) the service ecosystem fulfils the needs of the involved actors. Furthermore, business requirements for co-creation of value through services are changing faster and complexity of required value networks are increasing. One way to align the efforts and requirements of a spectrum of collaborating actors needed to co-create value is through a business model. Therefore, Turetken, Grefen, Gilsing, & Adali (2019) created the Service Dominant Business Model Radar (SDBM/R), grounded in S-D Logic. This is a practical visual tool designed to engineer complex digital innovations in a multi-stakeholder business environment. A template of the model is depicted in Figure 4.

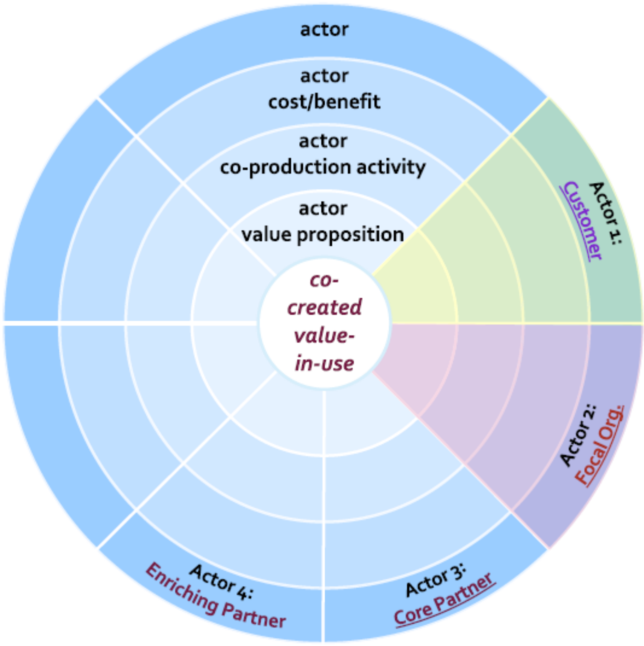


Figure 4: Service Dominant Business Model Radar (SDBM/R) Template (Turetken et al., 2019)

The center of the SDBM/R consists of the co-created value-in-use. This represents the value of a solution to the main beneficiary of the business model (i.e. customer). Therefore, this element of the radar is neither a service nor product. The first layer of the radar consists of the actor value propositions. This represents the part of the central co-created value-in-use that is delivered by individual actors. The second layer of the radar consists of the co-production activities. This represents the individual activities that each actor performs in the business model to achieve their individual actor value proposition. The third layer depicts the actor's cost and benefits. This represents the value that participating in the service ecosystem brings for each individual actor. Costs and benefits can be either financial or nonfinancial in character. The fourth and last layer of the model depicts the individual actors participating in the business model. The model is made such that each individual actor is contained to a 'slice' of the radar. The model uses several actor roles. The focal organization is the party that often initiates the business model and participates actively in the co-creation of value. The customer is the main beneficiary of the business model, but also actively contributes to the co-creation of value. The core partner(s) are those actors that also actively contribute to the business model, while enriching partners only enhance the value. There are no limitations on the number of actors participating in the model. The SDBM/R is set up such that it clearly depicts the mutual benefit for collaborating in the business model.

2.6 Value Proposition Driven Business Service Identification Method (VP-BIM)

The Value proposition driven Business Service Identification Method (VP-BSIM) by Adali, Ozkan, Gilsing, & Grefen (2021) guides the transformation of a value proposition into modular, standardized and contextualized actor resource configurations. Their method is based on the business service paradigm originating at the intersection of service-oriented technology, management and enterprise architecture research streams. Therefore, the concept of business service is used to model the modular, standardized resource configurations that an actor uses to co-create value in a service system. Furthermore, a service identification method is the systematic procedure to determine the resource configurations of actors in a service system. Therefore, this method goes beyond value propositions and towards realization of services that can possibly be exchanged on a digital platform.

Input of the VP-BSIM consists of value propositions designed, for example, with the Service Dominant Business Model Radar (SDBM/R). The method consists of three main steps that are performed once for each value proposition. The method steps are depicted in Figure 5 and will be briefly summarized. In the first step, elements of the SDBM/R are decomposed into Strategic Dependency (SD) and Strategic Rationale (SR) Models from the i* framework (Yu, 1995) to capture interdependencies and motives of actors in a service system to co-create value. In the second step, business capabilities are identified which actors need to co-create value in the service system. Therefore, the method employs Capability-Business Service Domain Mapping (Kohlborn et al., 2009) which reconfigures processes, activities and resources of actors into service domains, service operations and matched with business capabilities. In the third and final step of the method, a feature binding technique and service analysis is applied to ensure that business services are modular and replicable across value propositions. During the service analysis, principles from Service Oriented Architecture design are used as guidelines to define modular business services. The modular business service should be self-contained, stateless and representative of a domain-specific service.

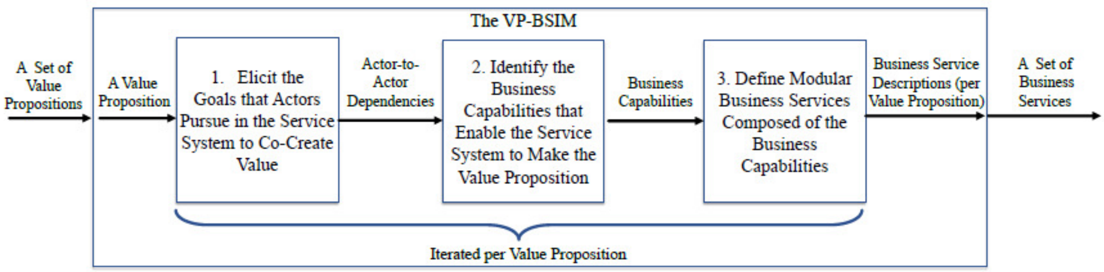


Figure 5: Main steps of the VP-BSIM (Adali, Ozkan, Gilsing, et al., 2021)

2.7 Adapted NFR Approach To Determine Platform Openness

Sadi & Yu (2017a, 2017b) see open innovation as an important strategy in software development. When following this strategy a software company opens up their software platform to third-party developers. According to the authors, opening up software platforms to third-party applications raises concerns about critical quality requirements. Therefore, a deliberate analysis of openness requirements is needed early on when opening up a software platform. To be able to perform such an analysis, the authors treated openness as a distinct class of non-functional requirements that need to be refined and analyzed in parallel with other design concerns, using a goal-oriented approach. The openness requirements are then used as criteria for selecting the optimal degree of openness via design alternatives. The authors hereby extended the Non-Functional Requirements (NFR) analysis method (Chung, Nixon, Yu, & Mylopoulos, 2000). Furthermore, the method complements research which is either focused on technical designs of platforms or only on the business aspect.

The seven main steps that are performed in the method are as follows: (1) specifying and refining openness requirements, (2) specifying and refining other design concerns, (3) Prioritizing the requirements, (4) Identifying possible alternative operationalizations, (5) Evaluating fulfillment degree of the identified requirements in each operationalization, (6) Analyzing potential trade-offs and (7) Selecting an appropriate design mechanism. An example of a goal model containing evaluated requirements for the design alternatives is depicted in Figure 6. This goal model is based on the concept of a soft goal interdependency graph.

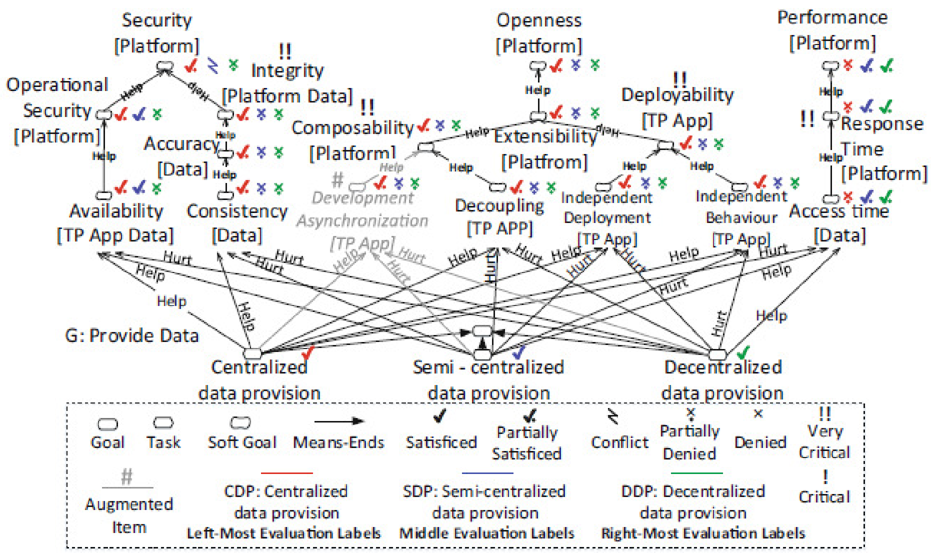


Figure 6: Example Specification, refinement and evaluation of design requirements for 'Design Objective: Data sharing' (Sadi & Yu, 2017a)

Sadi & Yu (2017a) complemented the goal-oriented approach from Sadi & Yu (2017b) with three types of catalogues to facilitate the specification and analysis of openness requirements; (1) openness requirements specification and refinement catalogue, (2) openness operationalization catalogues and (3) Openness correlation catalogues.

The openness requirements specification and refinement catalogue helps characterize and refine the specific requirements and concerns that openness introduces on the design of a software platform. The catalogue consists of three sub catalogues: (1) Business-level openness requirements, (2) System level openness requirements and (3) General design concern catalogues. Business-level openness requirements characterize non-technical requirements in an open software platform and are related to social, business or organizational environment of a software platform. They are also described as openness business objectives. The business-level openness requirement catalogue consists of two parts: a non-technical requirement and a related technical requirement that can refined to a system level openness requirement. System-level openness requirements characterize general technical and quality requirements that should be met in the design of open platforms. General design concerns characterize general concerns and requirements raised in opening up a software platform. These concerns can have conflicting relationships with the openness requirements. To develop each of these sub catalogues, a literature search should be performed in the domains of business and software engineering. Each catalogue proposes alternative paths for refining and operationalizing openness requirements, customizable for a particular design context.

The openness requirements operationalization catalogue identifies the platform system functionalities that should be specifically designed to open up platforms to third-party developers. The catalogue consists of two parts: (1) Design objectives which is the specific functionality needed to be designed or implemented, (2) Design alternatives which are alternative mechanisms to realize the design objective. The openness correlation catalogue identifies the impact of each openness design alternative (in the operationalization catalogue) on the fulfillment of the related openness requirements (in the specification and refinement catalogue).

2.8 Related Work

2.8.1 Digital Platform Design Frameworks

Several authors have developed frameworks which could be used as tools to help identify important design aspects to consider in the design of (domain specific) digital platforms and their ecosystems. The identified frameworks are described below.

Design framework for mobility service platform ecosystems

Hein et al. (2018) developed a structural, reproducible framework to design platform ecosystems from the service dominant (S-D) logic perspective. The aim was to help scholars with systematically comparing, and practitioners with designing a mobility service platform ecosystem (MSPE). The artifact consists of three morphological boxes to design a configurable MSPE from the platform owner perspective, depicted in Tables 1, 2 and 3. The morphological boxes are based on the three dimensions identified by Lusch & Nambisan (2015); service ecosystem, service platform and value co-creation. Furthermore, the boxes divide the dimensions into factors that summarize MSPE categories for the respective dimension. Moreover, each category consists of several concrete attributes that cover several entities. The source in the last row indicates where the attributes originate from (e.g. literature or interviews).

The service ecosystem dimension (Table 1) includes different sets of actors participating in an MSPE. Actor categories help to determine attributes of actors. Furthermore, each actor plays a general role and a context specific role (MaaS Role). Moreover, a motive for joining the ecosystem is defined. The box should be completed for each identified actor.

Table 1: Morphological box for the service ecosystem dimension (Hein et al., 2018)

Characteristics	Attributes	Entities							Source
Actor	Segment*	Business*	Government*	Consumer*					Interviews: 11
	Role[x]	Platform Sponsor	Platform provider (Intermediary)	Complementor (Designer)	Customer (Ideator)				(Kuebel and Zarnekow, 2014, Levina and Kranich, 2017, Jittrapirom et al., 2017) Interviews: 8
	Motive[x]	Access to customers	Access to resources	Financial benefits	Product/service based benefits*	Platform based benefits	Ethical benefits	Other*	(Frow et al., 2015, Pagani, 2013, Giessmann et al., 2014, Barrett et al., 2016) Interviews: 12
	MaaS Role*[x]	Passenger*	Driver*	Owner*	Provider*	Authority*			Interviews: 7

The second dimension is value co-creation (Table 2). This dimension covers the type of services provided and how value is created and captured. A service ecosystem can consist of multiple value co-creation services and thus multiple configurations of this morphological box.

Table 2: Morphological box for the value co-creation dimension (Hein et al., 2018)

Charac-teristics	Attributes	Entities								Source
Mobility Service	Geo-graphical Scope	Neighborhood-wide	Citywide	Region-wide	Countrywide*	Company-wide	Global			(Plenter et al., 2017, Täuscher and Laudien, 2017) Interviews: 12
	Service Pattern	Deferred	Immediate	Recurrent	Ongoing*					(Andersson et al., 2013, Plenter et al., 2017) Interviews: 7
	Nature^x	Trip planning	Naviga-tion	Smart logistic	Parking	Sharing*	Location-based	Transpor-tation*	Other*	(Schreieck et al., 2016c) Interviews: 8
	Openness	Public	Private							(Schreieck et al., 2016b, Leimeister et al., 2010, Benlian et al., 2015) Interviews: 8
Value Creation	Form	Co-Design	Co-Production	Co-Consump-tion	Co-Promotion	Co-Pricing	Co-Integration*	Other*		(Frow et al., 2015, Karpen et al., 2012, Jittrapirom et al., 2017) Interviews: 4
	Coopera-tion Chan-nel*^x	Physical*	Digital*							Interviews: 7
	Coopera-tion Intensity	Ecosystem	Strategic alliance	Loose cooperation	Purchase					(Labes et al., 2013) Interviews: 6
	Business Fields	Complementary	Similar	Substitutive						(Labes et al., 2013) Interviews: 7
	Activity Type^x	Production	Aggregation	Comparison & Categorization	Integration	Consulting	Other*			(Leimeister et al., 2010) Interviews: 5
	Duration	One-off	Recurring	Continuous						(Frow et al., 2015) Interviews: 6
Value Capture	Value Source*	Platform Sponsor*	Platform provider*	Complementor*	Customer*					Interviews: 5
	Value Stream	Transaction-based	Subscription-based	Indirect*						(Täuscher and Laudien, 2017, Jittrapirom et al., 2017) Interviews: 5

The third and last dimension is platform (Table 3). This is the actual artifact connecting two or more actors in the service ecosystem. The categories of governance and architecture define different configurations of the platform.

Table 3: Morphological Box for the platform dimension (Hein et al., 2018)

Charac-teristics	Attributes	Entities							Source
Governance	Structure^x	Authority-based	Contract-based	Trust-based					(de Reuver and Bouwman, 2012) Interviews: 5
	Control Mech-anisms^x	Input	Output	Behavior	Social	Access*			(Manner et al., 2013) Interviews: 12
Architecture	Resource Type^x	Hardware resource	Software resource	Data resource	Know-How resource	Human resource	Infrastructure resource*		(Labes et al., 2013, Plenter et al., 2017) Interviews: 12
	Focus*^x	Modularity	Flexibility*	Scalability*	Extensibility*	Availability*	Performance*		(Pagani, 2013) Interviews: 7
	Modules*^x	Ticketing and billing	Analytics	Monitoring	Filtering*	Matchmaking	Fleet-manage-ment*	Complemen-tary Modules	(Schreieck et al., 2016c, Jittrapirom et al., 2017) Interviews: 12

Generic Design framework for digital platforms

Tura et al. (2018) developed a framework to help practitioners with understanding the most crucial design choices preceding a platform launch. Therefore, it used an integrated view of platforms as both multi-sided markets and engineering artifacts. The framework consists of four main elements derived from a prior literature review: platform architecture, value creation logic, governance and platform competition. Therefore, platforms are viewed by the authors as consisting of multiple building blocks. The element of platform architecture focused on the platform actors, the market and the fundamental structure. The value creation logic involves design choices related to roles of actors participating on the platform, value propositions for these actors, network effects and the revenue model. The governance element consists of leadership, ownership and platform rules. Finally, the platform competition element consists of platform launch, competitiveness, innovation & learning and platform growth design choices. An excerpt of the framework is depicted in Figure 7. Filling in these elements of the framework for a specific context and domain generates a platform configuration that could be further validated, developed and implemented.

Problem focus	*Core design problem*
Platform architecture	Describes the structure of the actors, market structure
Core interaction	What is the main purpose of the platform? What is the (core) interaction that takes place in the platform? What are the value creating assets being transacted?
Market structure	Which markets are involved in the platform (two-sided, multi-sided markets)?
Key actors	Who are the actors representing different market structures and providing main functions?
Platform openness	How open is the platform and what is the strategy to manage openness? How openly is the data and information shared?

Figure 7: Excerpt of the platform design framework (Tura et al., 2018)

Framework for design, development and implementation of technology platforms in South African Healthcare

Herman, Grobbelaar, & Pistorius (2020) developed a framework that was intended as a practical tool for platform owners to increase the adoption of platforms in the sub—Saharan Africa (SSA) healthcare context. Therefore, the framework drew on both the engineering and economic research perspectives to provide a holistic understanding of digital platforms. The framework was developed by applying the grounded theory conceptual framework analysis process. The entire framework consists of multiple components. First, the pre-use framework, where several high-level platform dimensions should be determined. For example, to consider if the platform has a transactional or innovation purpose. Second, the ecosystem framework, which actually consists of three individual sub frameworks: platform owner, developer and end-user. These individual frameworks for each actor role on the platform depict important aspects to consider from their perspective when designing a platform. The third and final framework shows five steps in the design and development of the digital platform ecosystem. Starting with defining a platform core, then identifying the ecosystem and environment, considering the value creation logic, managing the platform and finally considering the platform evolution by evaluating performance. The specific SSA healthcare context elements are added to this last framework and remind the platform owner of the importance of, among others, local regulations and health and technology education.

To exemplify, the platform owner ecosystem framework is depicted in Figure 8. This platform owner perspective is highlighted since it depicts aspects to be considered when an organization aims to develop a platform. The framework comprises four main categories: by the platform owner firm, the platform, the ecosystem and evolution. Each of the main categories is then further composed of subcategories. With the subcategory of vision, Herman et al. (2020)

referred to the importance of defining, among others, the scope of the platform and it's ecosystem but also the goals the platform owner wants to achieve and the core interaction that will take place between actors with the help of the platform.

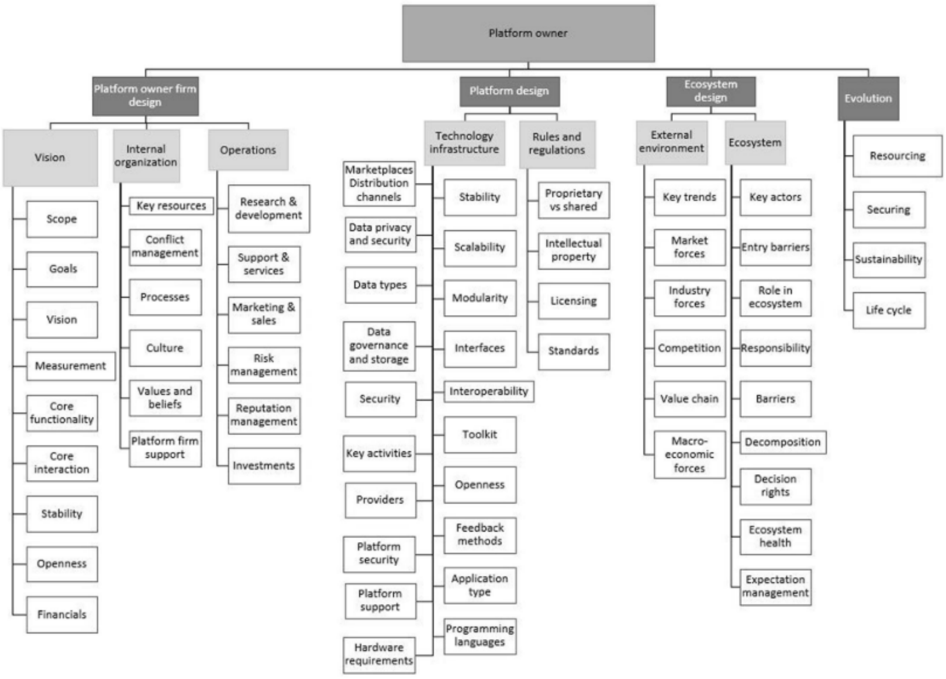


Figure 8: Platform owner ecosystem framework (Herman et al., 2020)

2.8.2 Value Proposition Design Methods

As described earlier in this literature review, value proposition design is an important early step in the design of a digital platform. This has also been confirmed by the various platform design frameworks described in section 2.8.1 which include this concept to a certain extend. This section describes various methods which use the concept of value propositions as a starting point for the design of (service) (eco)systems.

E^3 Value Ecosystem Modelling

The e^3 value modelling methodology has been used to model software ecosystems and value propositions of actors within this ecosystem. The method models a network of organizations that create, distribute and consume artifacts or services of economic value (Gordijn & Akkermans, 2001). Based on the understanding that requirements engineering consists of information system analysis of several distinct perspectives (business processes, IT architecture and the economic value proposition) the authors claimed there existed a gap in effective techniques to express and analyze the value viewpoint. Gordijn & Akkermans (2001) deemed the proper formulation of an e-business model the first step in the requirements analysis of e-business information systems.

In e^3 value modelling, relationships among actors are captured in terms of activity flows and input/outputs flows. An example and notation is depicted in Figure 9. The main modelling elements are 'actors', 'market segment', 'value activity', 'value exchange', 'value object', 'value interface' and 'value port'. Actors are economically independent entities and represent a company or consumers. Groups of actors with similar properties form a market segment. For example, in Figure 9 'Testing and verification party' is a market segment that represents companies which interact with the operating system manufacturer to test the system. An actor performs one or more value activities, which are assumed to yield a profit. Interactions between actors are captured in terms of value exchange and value object. Actors exchange value objects, which is a service, good, money, or experience, which is of economic value to at least one actor. Therefore, an actor uses a value port to exchange value objects. Furthermore, value ports are grouped into value interfaces to show economic reciprocity. Actors only offer objects to another actor if they receive compensation in return. Either each port in a value interface precisely exchanges one value object or none do.

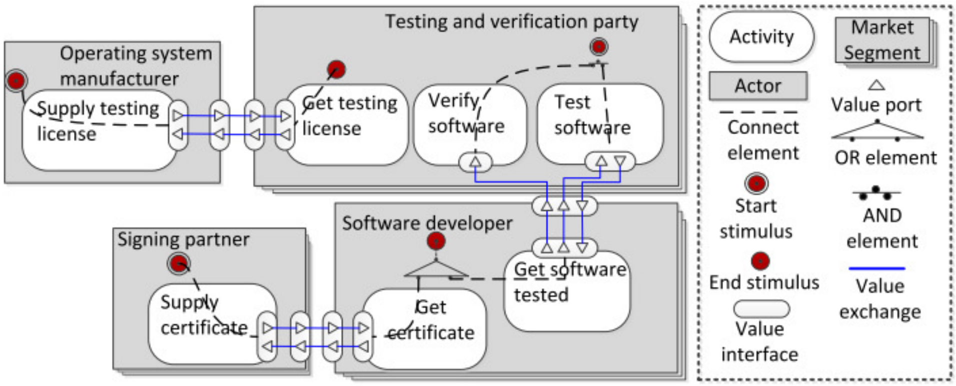


Figure 9: Example and notation of e3 value modelling (Sadi & Yu, 2015).

2.8.3 Service (System) Requirement Engineering Methods

Requirements engineering (RE) is considered a subfield of software engineering and is focused on finding and specifying requirements for software and software intensive systems (Vegendla, Duc, Gao, & Sindre, 2018). Loucopoulos & Karakostas (1995) defined requirements engineering as "… systematic process of developing requirements through an interactive co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats and checking the accuracy of the understanding gained". Although processes used for RE vary depending on the type of system under development the main activities are common to all processes: elicitation, analysis, specification (i.e. modelling), validation and management (Loucopoulos & Karakostas, 1995). Requirements engineering is considered one of the first steps in the realization of software systems, because a requirement is a specification of what should be implemented (Sommervile & sawyer, 1997).

Several requirements engineering methods have been proposed for service (eco)systems, which aim to elicit requirements of the system or services within such a system. Three have been identified and described below.

Meta model for service system requirements

Goal-oriented modelling has shown to be a suitable method for designing and developing service systems. However, goal modelling had not been systematically applied from a service dominant logic (S-D logic). Therefore, Lessard, Amyot, Aswad, & Mouttham (2020) developed a metamodel of service systems based on S-D Logic and derived a domain-specific profile of the goal-oriented requirements language (GRL) and a set of heuristics to elicit requirements for service systems. Analyses of requirements from the perspective of SD logic resulted in more comprehensive requirements through the identification of additional stakeholders and resources. Therefore, these requirements could lead to a more suitable design of a service system.

Lessard et al., (2020) first build a meta model of S-D logic derived from its core concepts (Vargo & Lusch, 2016). After adding two constructs specific to service systems, the metamodel that captures a service system was developed, as depicted in Figure 10.

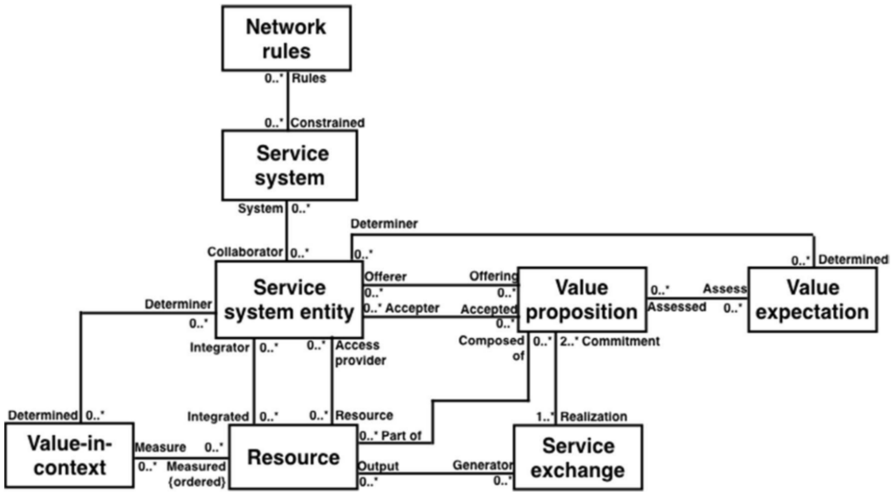


Figure 10: Service System Metamodel (Lessard et al., 2020)

To create requirements models from the service system model two steps were taken. First, a lightweight profile for the goal-oriented requirement language (GRL), part of the User Requirements Notation (URN) was proposed. GRL allows to reason about system requirements in terms of actor's intentions in relation to a system. Second, the service system metamodel constructs and relationships were mapped onto the GRL constructs and relationships. For example, the metamodel construct of 'network rules' became the GRL construct of a 'Goal' and the relationship of 'Network rules<>Service System' became GRL dependency links.

To instantiate the GRL model into requirements models, heuristics were identified for each GRL construct which comply with the service system metamodel. The information obtained by using the heuristics enables the creation of the requirement models.

Service Requirements Engineering Method

Immonen, Ovaska, Kalaoja, & Pakkala (2016) developed a service requirements engineering method specifically for digital service ecosystems. The digital service ecosystem is part of a service ecosystem, but only covers the digital part. The main difference with the concept of a software ecosystem is the lack of some technology (e.g. platform) underpinning the ecosystem (Immonen et al., 2016). The method is divided into three phases: service innovation (1), business analysis (3) and requirements analysis, negotiation and specification (3). The goal of the service innovation phase is to identify ideas for new services, scope and analyze them and transform them into service requirements. The goal of the business analysis phase is to identify which use cases have most business potential. The last phase aims to provide a complete requirement specification of the needed services which will be used as input for service architecture modelling.

The first phase of service innovation is further divided into two subphases: requirement elicitation and the requirements identification of services. Requirements elicitation defines what, how and from whom requirements should be elicited and guides the process of elicitation. Identification consists of identification, classification, merging and prioritizing service requirements. The last phase of requirements analysis, negotiation and specification is further divided into three interrelated and iterative subphases. The first subphase, analysis, determines the consistency, completeness and priority of requirements. The second subphase, negotiation, communicates the service requirements to business and technical stakeholders involved in the service development. The last subphase, specification, describes requirements using textual and graphical notations to make the requirements understandable and useful for all ecosystem members. This is done through to the use of a Use Case Analysis template.

Service Platform Requirements Engineering Method

Adali, Ozkan, Turetken, & Grefen (2021) used part of the method from Immonen et al. (2016) to extend the VP-BSIM. The goal of this extension was to enable the specification of service platform requirements. To do so, use cases are created per business service identified by the VP-BSIM. The first step of the extension reuses the subphase of requirements analysis by Immonen et al. (2016). In this subphase business services are identified that have potential for requirements specification. The second step of the extension reuses the subphase of requirement specification. In this subphase the service requirements are specified in textual or graphical format. Adali, et al. (2021) opted to also use a Use Case Analysis template. Therefore, the output of this extended VP-BSIM is a set of use case descriptions that describe the behavior of a service platform.

2.8.4 Software Ecosystem Modelling Methods

The software engineering field introduced and researched the concept of software ecosystems (SECOs). Manikas (2016) defined software ecosystems as the software and actor interaction in relation to a common technological infrastructure, that results in a set of contributions and influences directly or indirectly the ecosystem. The technological infrastructure of this software ecosystem often is a technology platform (Immonen et al., 2016; Manikas, 2016). Similarly to service systems, an important early step in the design of software ecosystems consists of requirements engineering (Vegendla et al., 2018). While no specific requirement elicitation tools and techniques exist specifically for SECO's (Vegendla et al., 2018), specific SECO modelling frameworks and techniques have been proposed. However, there is no widely adopted approach for SECO modelling and description (Jansen, Handoyo, & Alves, 2015; Pettersson & Andersson, 2016). Therefore, a few modelling techniques and frameworks are discussed.

Analysis and design of software ecosystem architectures

Christensen, Hansen, Kyng, & Manikas (2014) defined the concept of software ecosystem architecture to define organization, business and software aspects of a software ecosystem. With this concept, Christensen et al. (2014) wanted to model software ecosystems at an appropriate level of abstraction and showed how the concept could be used to both analyze existing software ecosystems and in the design of new software ecosystems. The three dimensions used by Christensen et al. (2014) followed the dimensions of (Campbell & Ahmed, 2010), but focused more on the structure of the underlying software ecosystem, instead of only the engineering process.

Christensen et al. (2014) defined the concept of software ecosystem architecture as the set of structures needed to reason about a software ecosystem, comprising of actor and software elements, the relations among them and their properties. Thus, software structure forms the core of a software ecosystem. Furthermore, since value creation is the main purpose of the ecosystem the business structure is relevant. Finally, the interaction and organization of actors and software needs to be governed, comprising the organizational structure. The different structures and examples of elements, relations and models to operationalize these structures are depicted in Table 4.

Table 4: Structures of a software ecosystem and their operationalization (Christensen et al., 2014)

Structure	Elements	Relations	Models
Organizational structure	Applications, platform, orchestrator, users, developers, boards, development projects, plans	Governed by, developed by, maintained by, connected to	Organizational structure, organization relationship & interaction, organization roles
Business structure	Products, services, partners, customers, resources	Channels, customer relationship, revenue stream	Business model canvas
Software structure	Modules, functions, services, nodes, developers	Depends on, used by, deployed on, developed by	Software architecture description

Organizational structures are described by actor and software elements that are related to governance. This includes aspects such as the involved actors, their roles and goals and the boundary of the ecosystem. Business structures are described by the business ontology of business model canvases as developed by Osterwalder (2004). Descriptions of the software structure consists of a set of architectural views (Kruchten, 1995). Using UML, the development view (i.e. how software is developed), functional view (i.e. how software behaves at runtime) and deployment view (i.e. how software is deployed to hardware) are defined. Furthermore, quality attribute scenarios (Bass, Clements & Kazman, 2013) are used to describe architectural requirements.

Software Ecosystem Meta-Model

Boucharas, Jansen, & Brinkkemper (2009) introduced the software ecosystem meta-model (SEM) consisting of two parts: product deployment context (PDC) diagrams and software supply network (SSN) diagrams. The modelling technique was introduced to help software vendors understand in which specific software system they are active and to use this ecosystem to their own strategic advantage. The PDC, exemplified in Figure 11, was created to provide a quick overview of a software product's architecture and dependencies. The PDC shows the hierarchy between products and components with a stack view.

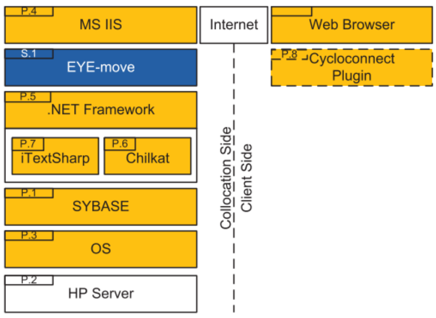


Figure 11: Example of Product Deployment Context Diagram (Boucharas et al., 2009)

The SSN diagram was proposed for describing and analyzing software ecosystems. This component, exemplified in Figure 12, describes the business relationships between participants of the ecosystem in terms of input and outputs flows between actors. The main elements of the SSN are actors, trade relationships, flows and gateways. Actors are organizations that participate in the software ecosystem (e.g. customer). Trade relationships connect two actors and are comprised of one or more flows. Flows represent a product or service between actors and consists of different types (e.g. product) and have a directionality. Gateways represent logical relationships between the flows and can be of the type 'OR' or 'XOR'. An SSN diagram consists of nodes and edges where nodes represent the participants and roles of the service ecosystem and edges represent the input and output flows between participants.

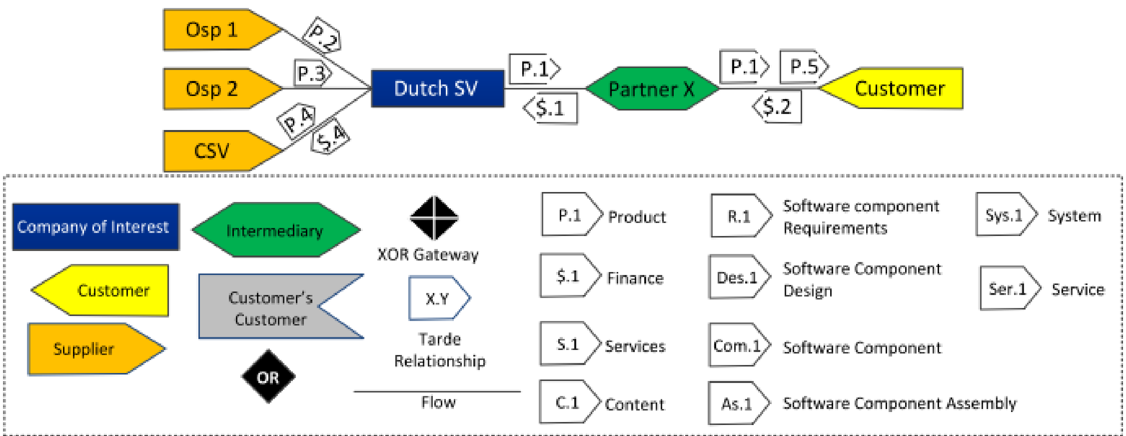


Figure 12: Example and notation of Software Supply chain Diagram (Boucharas et al., 2009)

2.8.5 Determining Digital Platform Openness

Although the platform design frameworks described earlier mention the aspect and importance of openness, they have not aided in the determination of the appropriate degree of openness of an instance of a digital platform by a platform owner. However, few approaches and methods exist that aim to help with this specific challenge.

A sequential innovation model to address openness trade-offs

Parker & Alstyne (2018) developed a mathematical model to characterize the optimal level of openness and intellectual property (IP) duration in a platform ecosystem for third-party developers with the aim of platform ecosystem growth. More specifically, it is an innovation model that addressed the trade-offs inherent in two decisions: (1) Closing a platform to increase the platform owner's ability to charge for access, while opening a platform increases the third-party developer's ability to innovate and (2) The longer third-party developers retain rights to their innovations the higher their income, but the shorter this intellectual property lasts the sooner new innovations can be built upon this previous innovation.

The model consists of three stages and includes a platform owner, third-party developer and end consumers as actors. Developers produce output using platform resources (e.g. API's, system developer toolkits (SDKs)). End users consume both the platform and the developers' output. The platform owner offers a one-time take-it or leave-it contract to third-party developers who can either participate on the platform or not. Through this contract, platform owners offer access to its IP, but in exchange also gains access to the developer's IP. In the first stage, platform owners open their technology, giving away IP, which developers can use to innovate with. Developers extend this IP and sell to end users and share revenues with the platform. Then the platform absorbs all IP extensions. In the second stage, the platform owner gives away all new IP extensions from the first stage. Developers act similar to stage 1 and the platform absorbs all cumulative IP extension. The timing of the model is depicted in Figure 13. During the model stages, the platform owner can change several parameters. For example, the platform owner can decide to change the expiration date of developer IP rights of innovations. Furthermore, platform owners can change the share of innovation profits by imposing royalties on their IP. Moreover the model is extended by including alternative organizational forms such as developers' decision to cooperate with other developers instead of accepting the platform owner's terms. This mechanism intended to model the control dimension of openness.

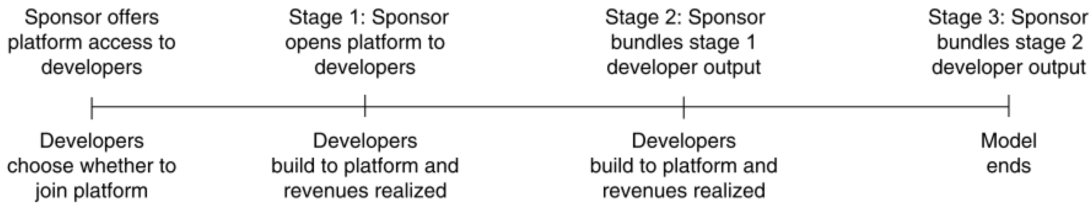


Figure 13: Platform Model Timing (Parker & Alstyne, 2018)

An integrated model of drivers, dimensions and outcomes of platform openness

Broekhuizen et al. (2021) introduced a research framework which contains drivers, dimensions and outcomes of digital platform openness identified by prior literature on transaction platforms. The framework, depicted in Figure 14, may help both researchers and practitioners to understand under which circumstances digital platforms are more likely to use a certain openness strategy and to predict the consequences of changing the platform's openness for the platform and its actors. Therefore, the authors claim it may guide decision making in the degree of platform openness.

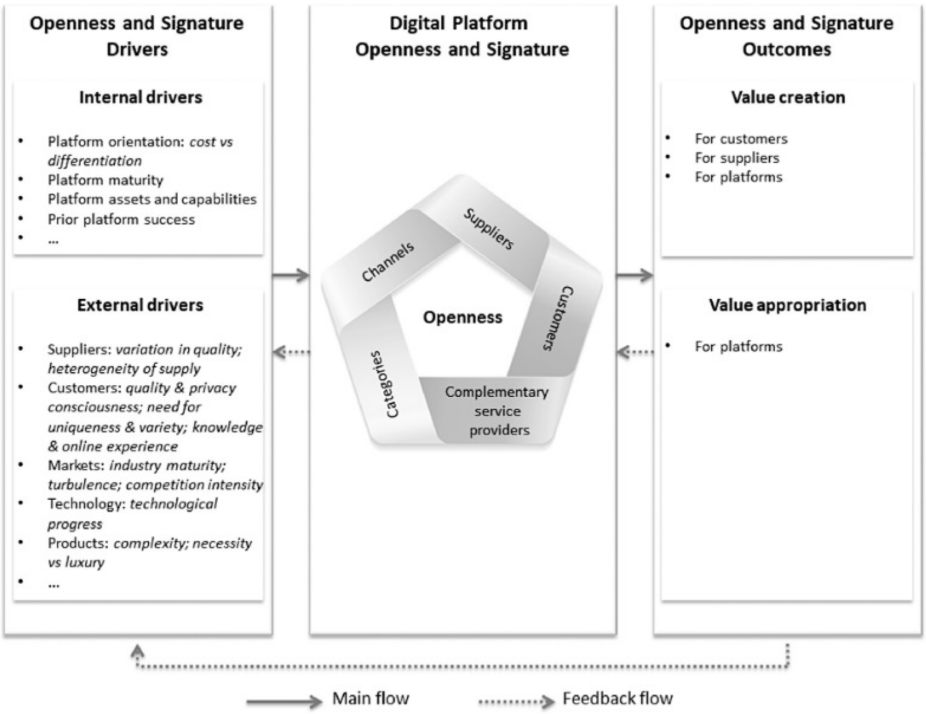


Figure 14: Research framework for drivers, dimensions and outcomes of platform openness (Broekhuizen et al., 2021)

Broekhuizen et al. (2021) defined a platform's openness signature as the platform's image or fingerprint for openness as perceived by its users. This platform signature consists of the five openness dimensions: three actor-based (suppliers, customers, complementary service providers) and two non-actor-based (categories and channels). The authors argue that besides making decisions on who is allowed on the platform and how much authority is given to actors, platform owners also need to decide on what categories of assortments are provided and through which channels of communication and distribution.

The framework identifies a set of drivers of platform openness for each openness dimension. Broekhuizen et al. (2021) state that these drivers may not deterministically define an ideal configuration of platform openness, but it may guide decision making based on the particular outcomes that platform owner desire. Therefore, the framework assumes that openness decisions relate to the platform's signature, which leads to outcomes in terms of creation of value for the platform's actors and the appropriation of this value. Therefore, practitioners could use this framework to identify opportunities and threats that emerge from a certain configuration of openness decisions across the openness dimensions to understand the resulting synergic and competitive effects, such that the desired platform signature can be established (Broekhuizen et al., 2021).

Broekhuizen et al. (2021) also created a taxonomy of openness signature configurations (Figure 15). However, the authors argue that a successful and appropriate configuration of dimensions also depends on the characteristics of the demand side of a platform (e.g. customer heterogeneity), supplier side of a platform (e.g. heterogeneity in quality), products or services provided by the platform (complexity) and platform characteristics (e.g. prior success).

Platform signature	Unified vs. diversified	
Openness decisions regarding...	Unified shopping platforms	Diversified shopping platforms
Examples	Amazon	OTTO
Supplier access	High	Low
Supplier authority	Low	High
Customer access	High	Low
Customer authority	Low	Low-high
Complementary service providers access	High	Low-high
Complementary service providers authority	Low	Low-high
Categories	High	High
Channels	Low-high	High

Figure 15: Part of the platform openness signature taxonomy for transaction platforms with examples (Broekhuizen et al., 2021)

2.9 Discussion and Conclusion of Review

Prior literature has called for research to help answer the question of how digital platforms are designed (De Reuver et al., 2018). However, the design of a digital platform is considered a complex undertaking. Two key design aspects are value proposition design and the degree of platform openness. Value propositions are deemed key determinants of the success and level of engagements between actors on digital platforms (Chandler & Lusch, 2015). From a service (eco)system perspective, value propositions are seen as invitations from actors to one another to engage in a service (Chandler & Lusch, 2015). In this context, value proposition design is seen as a systematic search process that actors in an ecosystem (e.g. platform owners) can perform to improve existing offerings, create new offerings, and reconfigure their ecosystems (Maglio & Spohrer, 2013). The ability of an actor in a service system to create and accept new offerings depends, among others, on the openness of a platform. Therefore, this too should be determined early as it is crucial to market success (Alstyne et al., 2016). Platform openness is defined by the level of restrictions on using, developing or commercializing functionality of a digital platform (West, 2003).

In line with the call from De Reuver et al., (2018) for design knowledge and design practices for practitioners, the existing body of academic work was identified, which discussed approaches, methods or frameworks to deal with the mentioned key design aspects of a digital platform. This ranged from general platform design frameworks, to literature specifically focused on value proposition design or platform openness. Furthermore, a few papers regarding requirements engineering, more specifically elicitation and modelling, were included since this is considered a common first step in the design of software systems.

First several digital platform design frameworks were identified. Hein et al. (2018) provided a design framework from the S-D logic perspective. However, the framework focused on the mobility service sector and is not directly generalizable to other domains. Furthermore, it does not contain guidance to help apply the framework. Tura et al. (2018) developed a more generic framework to platform owners aware of key design aspects. Similarly, Herman et al. (2020) developed frameworks which considered design aspects of multiple actor roles a digital platform. Although all the frameworks mention the key design aspects of value proposition (or value co-creation) design and platform openness, the concepts are treated in a limited capacity. For example, Hein et al. (2018) limits the openness discussion to the platform being private or public.

Several value proposition design methods were also identified. E^3 value ecosystem modelling (Gordijn & Akkermans, 2001) is able to capture relationships between actors in terms of activity flows and has previously been applied for modelling software ecosystems. The Service Dominant Business Model Radar (SDBM/R) (Turetken et al., 2019) is a tool specifically designed to engineer value propositions for complex digital innovations in a multi-actor setting and to align those actors. The VP-BSIM by Adali et al. (2021) used the input of the SDBM/R to translate value propositions into modular business services, bringing value propositions closer to realization.

Several papers were identified that can help to determine the appropriate degree of platform openness. Parker & Alstyne (2018) developed a mathematical model to characterize the optimal level of openness, focusing specifically on trade-offs inherent in the decisions to open or close a platform. However, their model reduced openness to the discussion of intellectual property. Broekhuizen et al. (2021) introduced a research framework which contains drivers, dimensions and outcomes of digital platform openness and could potentially help practitioners to identify opportunities and threats that emerge from a certain configuration of openness decisions. However, it focused only on transaction platforms and does not guide practitioners in the application of the framework. Sadi & Yu (2017ab) developed a method which defines openness requirements as a distinct class of non-functional requirements and used these as criteria to evaluate and select the optimal degree of openness via design alternatives that could implement functional requirements needed to open up a digital platform. However, its use is limited to platform openness towards the actor role of third-party developers on a digital platform.

The last few identified papers discussed elicitation and modelling methods for engineering service systems, digital service ecosystems and software ecosystems. Both service systems and software ecosystems are enabled by a technology such as a digital platform. Three service (system) requirements engineering methods were identified. Lessard et al. (2020) provided a meta model that captures concepts and relationships found in S-D logic and a set of heuristics to elicit requirements for service systems. Immonen et al. (2016) developed a method to define services in a digital service ecosystem and translates the services into requirements using textual and graphical notations. However, a digital service ecosystem assumes there is not central underlying technology (i.e. digital platform) to exchange and deliver the services. Finally, Adali et al. (2021) used elements of the method by Immonen et al. (2016) to specify requirements for a service platform. However, the use case analysis employed is relatively labor intensive.

Two papers in the domain of software ecosystem modelling were identified. Christensen et al. (2014) created a framework of three dimensions; organization, business and software to structure a software ecosystem and included tools such as the business model canvas (Osterwalder, 2004) which orchestrators of the ecosystem could use to operationalize the dimensions. Boucharas, Jansen, & Brinkkemper (2009) developed the software ecosystem metamodel (SEM) of which the software supply chain notation (SSN) can be used to define actors and their relations within the ecosystem. However, the SSN does not include a tool for modelling.

To conclude, few of the identified papers provide methods and approaches that can be easily used by practitioners to design value propositions or platform openness. Especially openness is treated in a very limited capacity by platform design frameworks. Moreover, methods and approaches that do exist, only treat the key design aspects individually. Therefore, the research gap identified by this literature review is the need for and lack of integrated methodological guidance on the design of digital platforms that accounts for the key aspects of the value proposition design and corresponding degree of platform openness.

3 Research design

This research adopts the design science research approach as the basis of its research design. The main objective of design science within information systems research is the creation and evaluation of IT artefacts intended to solve identified organizational problems (Hevner, March, Park, & Ram, 2004). According to Hevner et al. (2004), artefacts may include constructs, models and methods. This aligned with the objective of this research which was to develop a method to solve a problem within a specific domain.

This research implemented a design science research methodology (DSRM) proposed by (Peffers et al., 2007). This method was deemed suitable since practice orientation had a high importance for the designed artefact. The method is clearly defined in six steps as depicted in Figure 16.

The first step was to identify and motivate the research problem. For this research, the problem was identified through discussions with practitioners from an organization that wanted to take initiative to develop a digital platform, and a literature review. In the second step design objectives for the artifact were defined. In the third step the actual artifact was created by applying situational method engineering (SME) (Ralyté et al., 2003). In the fourth step the use of the artifact was demonstrated to solve one instance of the problem within a demonstration case company. In the fifth step the artifact was evaluated with the use of a focus group and a questionnaire which used the design evaluation criteria from the technology acceptance model (TAM) (Davis, Fred, 1985), and results of the demonstration and evaluation were compared to the design objectives to assess its fulfilment. Since design science research is an iterative method, the feedback from the evaluation is used to propose refinements to the artefact. In the last step, the artefact is communicated in the form of this thesis report and an accompanying presentation of the artefact, the design process and evaluation results to relevant audiences.

These steps are further elaborated in the next sections of this chapter. This chapter is concluded by a description of the demonstration case company whose context and input was used to demonstrate and evaluate the artifact.

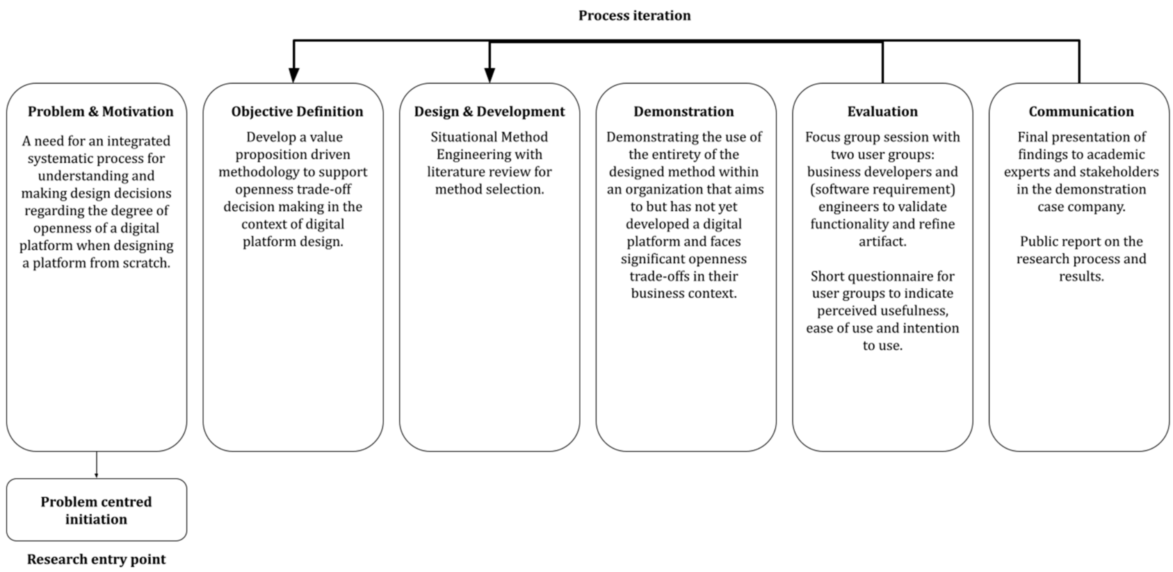


Figure 16: Design Science Research Methodology (DSRM) for this research, adapted from *(Peffers et al., 2007)*

3.1 Research Problem Identification and Motivation

The first step in the DSRM process by (Peffers et al., 2007) was to determine the research problem and motivation. The research problem was considered to *be the lack of an integrated method that could define the appropriate degree of platform openness during value proposition driven platform design*, as previously explained in Chapter 1. Furthermore, Chapter 2 includes a literature review which elaborated on the problem domain and included existing methods and frameworks that have tried to deal with the research problem.

3.2 Definition of design objectives

The second step was to determine the design objectives (DO) of the artifact that needed to be designed. These design objectives described how the designed artifact was expected to support solutions to the research problem (Peffers et al., 2007). Therefore the research objective was *to develop a digital platform design methodology that is value proposition driven and supports the determination of the appropriate degree of platform openness by making trade-offs explicit.* Three main design objectives were identified and are described below.

The first two design objectives were a direct result of discussions with practitioners and the literature performed and described in Chapter 2. The first design objective also allows the use of the method by practitioners (i.e. platform owners) who have not yet identified the design aspects and requirements of their digital platform under development.

DO1: The method should be value proposition driven.

DO2: The method should define explicit trade-offs related to a certain degree of openness.

DO3: The method should be useful, easy to use, and encourage intention to use the method.

The third and last design objective identified related to the use of the method by practitioners. This ensured that users of the method would be capable to execute the method without the use of external experts and generally aimed to ensure the acceptance of the method. For its formulation, the design evaluation criteria of the Technology Acceptance Model (TAM) developed by (Davis et al., 1989) were used.

3.3 Design and development

The aim of the demonstration and development step is to actually create the artifact (Peffers et al., 2007). To construct the actual artifact, situational method engineering (SME) (Ralyté et al., 2003) was used. SME supports the process of constructing or adapting a method to match the requirements of a given project situation.

Ralyté et al. (2003) consider three SME techniques: (1) to assemble method chunks, (2) to extend an existing method and (3) to generate a method by abstraction of a meta-model (i.e. paradigm-based method). Any SME technique consists of two main tasks: identifying the method engineering goal and then constructing a method that satisfies the set goal. The identified method engineering goal *was to extend the VP-BSIM to develop a digital platform design methodology that is value proposition driven and supports the determination of the appropriate degree of platform openness by making trade-offs explicit*. Moreover, the intention was not to create a method from scratch as enough relevant literature was identified in the literature review. Therefore, the extension-based strategy seemed most appropriate. More specifically, the extension-based pattern matching strategy was applied, since the required extension was initially not identified to be of a certain type. This strategy consisted of defining requirements which the extension should fulfil and identifying and applying the most fitting extension. All the steps are described in Figure 17. The requirements set for the extension were mostly based on the second design objective (DO2): the method extension should define explicit openness trade-offs (1), should be able to define multiple degrees of openness (2) and should use similar modelling languages (e.g. i-star) to make adoption easier (3). However, the literature review performed (Chapter 2) identified only one method related to determining platform openness. Therefore, the remaining effort focused on applying and matching this method with the VP-BSIM.

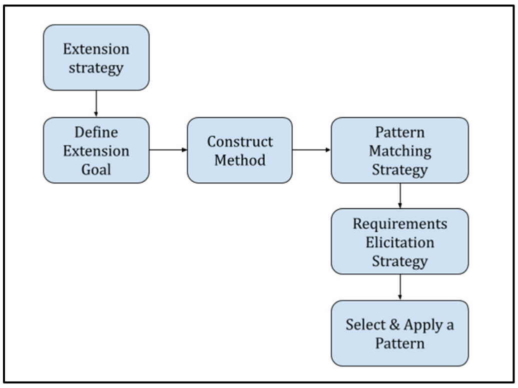


Figure 17: Extension based Situational Method Engineering Strategy used to design the artifact *(Ralyté et al., 2003)*

After following this extension-based strategy, two methods had been selected that met the stated design objectives. Design objective 1 (DO1) was met by starting with the value proposition business service identification method (VP-BSIM) by Adali et al. (2021) which translates value propositions of actors in a service ecosystem into modular business services which can be exchanged via a digital platform, bringing value propositions closer to realization. Therefore, the method allows platform design to be value proposition driven. Design objective 2 (DO2) was met by the method developed by Sadi & Yu (2017ab) which defines openness requirements as a distinct class of non-functional requirements (NFR) and used these as criteria to evaluate trade-offs and select the appropriate degree of openness via design alternatives that could implement functional requirements needed to open up a digital platform. However, there was still a mismatch between the methods since the method by Sadi & Yu (2017ab) started from requirements. Therefore, the business services from the VP-BSIM needed to be translated into platform requirements.

To translate the business services from the VP-BSIM into platform requirements, the concept of software services was used to add an additional method step. Software services support the execution of a business service with information technology (Kohlborn, et al., 2009). Software services expose functionalities that can be reused and composed based on business needs (Kohlborn, et al., 2009). Kohlborn, et al. (2009) developed a method to derive software services from business services. To minimize complexity of the artifact, a heuristic was developed by the author of this research, based on the paper by Kohlborn et al., (2009). The details of the heuristic are presented in Chapter 4. The software service identification method developed by Kohlborn, et al. (2009) consists of three main steps: preparation, identification and detailing. The heuristic focused mainly on the first two steps. The output of the first step are business services that are deemed suitable for software service enablement. The second step derives the software services by, among others, identifying potential service operations that can be automated. The third step is focused on verifying that existing software services do not overlap with newly identifed ones. This step was omitted since the designed artifact of this research assumed the creation of a digital platfrom from scratch, without existing infrastructure in place.

Finally, small adjustments were made to the extension from Sadi & Yu (2017ab) to streamline this method. This was done, since their method was described with a slight variation over two research papers. Their latest paper (Sadi & Yu, 2017a) has been used primarily. However, the various catalogues of non-functional openness requirements were taken from (Sadi & Yu, 2017b). These are presented in Appendix E. Furthermore, when applying the goal models (i.e. interdependency graphs) presented in their papers, the hurt/help links from the design alternatives towards the most-refined requirements have been left out. This was done to reduce visual complexity of the models and was carefully considered not to affect the performance of the model.

3.4 Demonstration

The aim of this step in the design science research method by (Peffers et al., 2007) was to demonstrate the use of the designed artifact to solve one instance of the problem. To this end, several steps were performed. First, an organization was selected that experienced the identified problem. This is also described as the focal organization, a company who has taken the initiative to develop a digital platform. Next, a demonstration instance of the method was executed by the researcher for this organizational context. This demonstration instance also helped to evaluate the designed method with practitioners employed by the selected organization (see section 3.5).

The organization was selected for two reasons: (1) Their desire to develop an open digital platform, while having limited understanding of the value of a digital platform for their business context and (2) the clear openness trade-offs that followed from their business context. The organizational context and choice for this organization was further elaborated on in section 3.6.

Demonstration of the designed method was performed as follows. In an online interview, the author of this research explained the purpose and elements of the input step of the designed method (SDBM/R) to participant 1 (Table 5) and let the participant describe their vision of the service (eco)system. After the interview, the author of this research filled in the SDBM/R based on the information from the interview. In a second interview, the filled in SDBM/R was verified with participant 1. The author of this report then constructed the SD and SR models from the first artifact method step and created the basis of the Service Domain - Capability matrix. Thereafter, an interview was held with participant 2 (Table 5). In this interview the completed input method step (SDBM/R) was presented and verified for a second time. Moreover, the capabilities of actors in the ecosystem were identified by participant 2 and the service operations were matched with the capabilities. From hereafter, the author of this report constructed the output of the remaining method steps without additional validation by the practitioners.

Table 5 shows the details of the participants of the demonstration. Since a formal project had yet to be assigned to the digital platform development, no formal project roles were identified.

Table 5: Details of the Participants of demonstration

Participant	Company	Company Function	Educational and/or Professional Background
1	GOAL3	CEO of the focal organization, responsible for overall company operations and product strategy	Tropical doctor, General practitioner
2	GOAL3	CMO of the focal organization, responsible for market research and business model strategy	Innovation Management

The scope of the demonstration was limited to one iteration of the designed method. This implied the creation and selection of one SDBM/R, generating multiple software services then working out the non-functional openness requirements and (evaluated) design alternatives for one selected software service. Therefore, the output of the demonstration was one selected design alternative that was judged to best balance the openness trade-offs when implementing one software service that was required for opening up the digital platform.

3.5 Evaluation and communication

The aim of the evaluation was to understand how well the artifact supports a solution to the problem (Peffers et al., 2007). During the empirical evaluation with practitioners the artifact's functionality and desirability was measured and compared to the previously stated design objective (DO3). The other design objectives (DO1 and DO2) were self-evaluated based by reflecting on the demonstration case (presented in Chapter 5).

The evaluation was performed ex post in a naturalistic setting since applicability to practice was deemed of high importance (Pries-Heje, Baskerville, & Venable, 2008). The evaluation methods used for this research consisted of a focus group complemented by a short questionnaire that was shared by the researcher after the focus group. Focus groups have been considered an effective method to evaluate the functionality of a designed artifact and to propose refinements for an artifact (Gibson & Arnott, 2007; Tremblay, Hevner, & Berndt, 2010).

The focus group session consisted of several steps. First, the designed method and results of the demonstration instance of the designed method were presented by the researcher. Secondly, the designed method and results were discussed. This discussion centered around open-ended questions to understand the motivations behind the desirability of the method and to propose refinements for the method. Details of the participants are presented in Table 6 below. All participants were employed by the demonstration case company, GOAL3. Further details of the focus group such as its procedure, setting and the results are presented in Chapter 6.

Table 6: Details of Participants of the focus group

Participant	**Company**	**Company Function**	**Educational and/or Professional Background**
1	GOAL3	CEO of the focal organization, responsible for overall company operations and product strategy	Tropical doctor General Practitioner
2	GOAL3	CMO of the focal organization, responsible for commercialization	Innovation Management
3	GOAL3	CTO of the focal organization, responsible for all product development	Embedded Systems Engineer, Industrial Automation
4	GOAL3	Lead software engineer, responsible for design and development of software products	Embedded Software Engineering Creative Technology
5	GOAL3	Head of research and innovation, responsible for the innovation roadmap	Mechanical Engineering Aerospace Engineering 8+ years' experience in healthcare technology

The short questionnaire shared after the focus group used the design evaluation criteria of the technology acceptance model (TAM) by (Davis et al., 1989). The TAM model has widely been accepted within information systems research to evaluate user acceptance of a designed artifact. The TAM constructs have also been acknowledged as appropriate to evaluate the 'adoption in practice' when the designed artifact is a method (Moody, 2003). According to Moody (2003) there are clear parallels between the adoption of information systems and methods since the decision made by the practitioner to use either one is deemed an individual choice and the result of reasoned action.

The TAM model measures the construct of 'intention to use' via two constructs: perceived usefulness and perceived ease of use. Perceived usefulness is defined by (Davis et al., 1989) as the degree to which a person believes that using a particular system would enhance his or her job performance. Perceived ease of use is defined as the degree to which a person expects the use of a particular system to be free of effort (Davis et al., 1989). In the context of this research, the idea behind the using the model constructs was that when the user of the method evaluated the method as useful and it took minimal effort to understand or apply the method, they are more inclined to use the designed method in the future.

The three constructs of TAM were operationalized using multiple 'items'. Similar to (Moody, 2003) the wording of each item was modified to accommodate a method as the design artifact. Participants of the questionnaire could express their level of agreement with each statement (i.e. item) on a 5-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). Furthermore, the question was asked whether their current role was business or engineering oriented to understand if the method would be perceived differently by these perspectives. The questionnaire and its results are presented in Chapter 6.

Based on the evaluation results multiple improvements of the artifact were developed. These are also presented in Chapter 6.

In the last step the designed method and results of the evaluated demonstration case have been presented to the case company and TU/e.

3.6 Demonstration case context

The demonstration case is performed within the company named GOAL3. This is a healthcare start-up based in Den Bosch, The Netherlands[1]. The aim of the startup is to contribute to the third Sustainable Development Goal set by the United Nations; 'Ensuring healthy lives and promote well-being for all at all ages'[2]. Therefore, their products and services are focused on markets of developing countries (i.e. low resource settings, low-middle income countries).

Their first product currently in development is a patient health monitoring device named IMPALA. This system registers vital signs of a patient such as heartrate and oxygen levels and communicates this information via displays to nurses and doctors present at the intensive care unit. This allows them to quickly diagnose and respond to changes in the health of a patient.

The management team of GOAL3 is convinced that to reach the high level of social impact the organization aims for and guarantee a sustainable business in the future, the adoption of a platform strategy and thus creation of a digital platform is essential. The organization is in the formative phase of their vision for this digital platform and aims to use the platform to accelerate innovation in healthcare via third-party developers such as public and private research groups (i.e. universities). This vision is aligned with the concept of an open digital platform aimed at (service) innovation. However, the business domain of healthcare brings many risks, for example, with regard to privacy and security of exchanged patient data and operational availability of (third-party) technology. Therefore, it is important for the organization to make the appropriate openness trade-offs when designing a digital platform. Furthermore, the organization has little experience in developing digital platforms and not everyone within the organization has been convinced of the value of a digital platform. Therefore, this organization is a good fit with this research to demonstrate the entirety of the designed method, including the value proposition design method steps (step 1-3).

[1] https://goal3.org/
[2] https://sdgs.un.org/goals/goal3

4 The Designed method

4.1 Method Overview

The goal of the designed method is to support the determination of an appropriate degree of platform openness via openness trade-off decision making in the context of value proposition driven digital platform design, turning design into a decision-making problem. The designed method is intended to be used by business developers and (software) requirement engineers and does not require a focal organization to have an existing operational digital platform. Therefore, the designed method starts from a co-created value proposition to identify which business services that enable the (co-created) value proposition(s) might be supported by a digital platform and are required to open up the digital platform, specifically towards third-party developers.

To achieve this goal, both a heuristic and method have been added to the VP-BSIM as explained in Chapter 3. The heuristic based on (Kohlborn et al., 2009) derives software services from the business services identified by the VP-BSIM. These software services are considered to be requirements (Gorschek & Wohlin, 2006) which together define a digital platform. The method by Sadi & Yu (2017ab) identifies non-functional openness requirements which are used evaluate and compare design alternatives which implement a software service (i.e. requirement) with a certain degree of openness. The method by Sadi & Yu (2017ab) evaluates these design alternatives via a formal goal model technique based on the i* language by (Horkoff & Yu, 2009). An overview of the designed method is depicted in Figure 18. All the steps of the designed method are described in this chapter.

The method starts with one or multiple service ecosystem value propositions envisioned by a focal organization. These ecosystem value propositions are generated by using the SDBM/R and serve as input for the VP-BSIM. The VP-BSIM consists of three main steps that are executed for each ecosystem value proposition. These steps are depicted by 1,2 and 3 in Figure 18.

The output of the VP-BSIM method is a set of business services that can be provided by a focal organization to realize their value proposition to the ecosystem. The businesses services are used to identify software services, which could support these business services. This identification follows a heuristic and is depicted by step 4 in Figure 18.

The output of the heuristic is a set of software services (i.e. requirements). For each individual software service, non-functional requirements and design alternatives to implement the software service are identified. The design alternatives are then evaluated on these non-functional requirements and compared to choose the most suitable design alternative. This depicted by steps 5,6,7 and 8 in Figure 18. The output of the designed method consists of a set of feasible design implementations of software services (i.e. requirements) which realize the desired value proposition of the focal organization with a certain degree of openness.

The designed method is intended to be used first by a focal organization to envision their digital platform and then to iteratively validate the value propositions and other (intermediate) outputs of the method such as the (non-) functional requirements and design alternatives, with other service ecosystem actors.

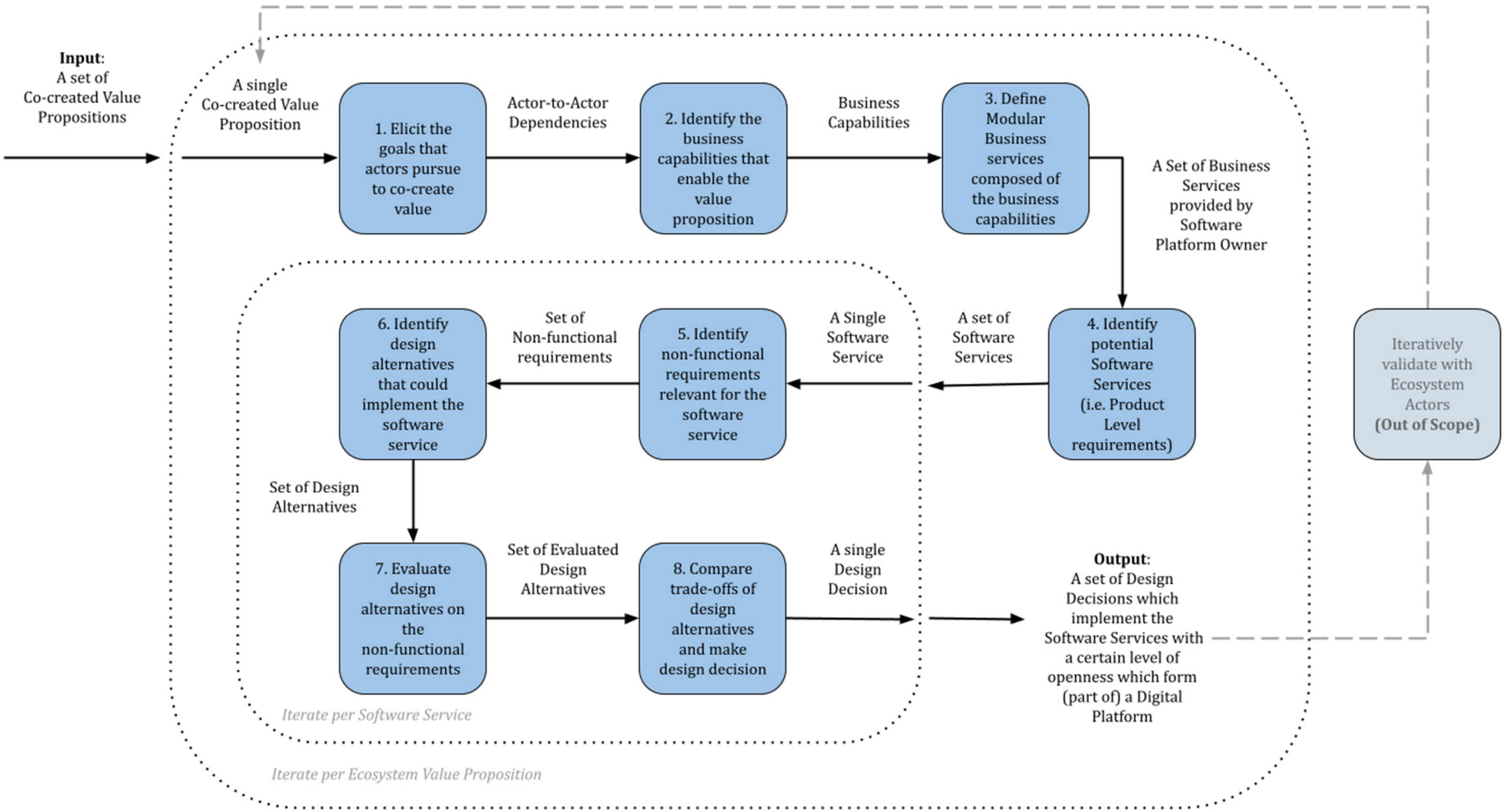


Figure 18: Method Overview

4.2 Input: Service System Value Propositions

The purpose of this preparatory step is to create the input that will be used for the VP-BSIM. This input consists of one or multiple co-created value propositions derived with the use of the SDBM/R (Turetken et al., 2019). If more than one co-created value propositions is identified, multiple SDBM radars are made. The construction of the SDBM/R is an iterative process but is explained as sequential design steps. The steps of the SDBM/R are intended by (Turetken et al., 2019) to be executed by representatives of all the identified actors. However, for the designed method, the steps will first be executed solely by the focal organization:

To construct a SDBM radar, the first step is to identify the co-created value in use of the service ecosystem and the actors who contribute:

A.1 Determine co-created value in use.

A.2 Determine actors of the service system (e.g. Customer, Focal organization, Core partner and Enriching partner)

The customer (i.e. end user) is the main beneficiary of the service system. The focal organization is the actor who initiates the service system and is often the digital platform owner. Core partners actively contribute to the service system, while enriching partners only enhance the value of the service system (Turetken et al., 2019).

A.3 Determine the value proposition of each actor taking part in the value co-creation.

A.4 Determine the activities each actor performs in the ecosystem to fulfil its contribution.

A.5 Determine the costs and benefits (i.e. resources) that each actor accrues through their activities to fulfil its value proposition.

The template depicted in Figure 19 will be used to help execute these steps.

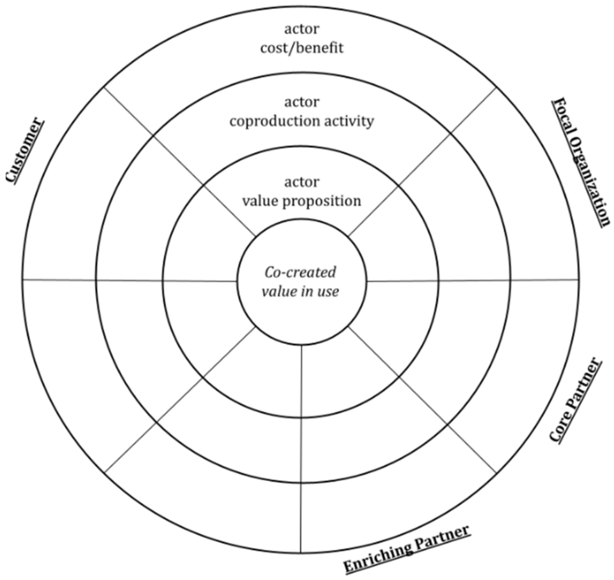


Figure 19: SDBM/R template adapted from (Turetken et al. 2019)

The output of this preparatory step is the SDBM/R per co-created value proposition.

4.3 Step 1: Elicit the Goals that Actors Pursue in the Service System to Co-Create Value

The purpose of the first step is to capture the dependencies among actors in the envisioned service ecosystem and the underlying motives of actors belonging to these dependencies (Adali, Ozkan, Gilsing, et al., 2021). The input for this step is one of the SDBM radars created in the preparatory step.

The sub steps are as follows:

1.1 Select a single SDBM/R

1.2 Create a SD model

1.2.1 Consider the actors described in the co-created value proposition as the actors for the SD model.

1.2.2 Determine the dependencies between actors.

1.2.3 Determine the type of the defined dependencies (e.g. goal, task or resource).

The first step is to choose a SDBM/R if multiple radars have been created in the preparatory step. With input of this chosen SDBM/R the Strategic Dependency (SD) model is created. The main concepts of this model are depicted in Figure 20. The creation of this SD model follows a few guidelines. First the co-created value proposition is modelled as a goal dependency between the customer and the focal organization (i.e. platform owner). Secondly, the cost and benefits from the SDBM/R are modelled as resource dependencies between actors. Thirdly, actor co-creation activities become task dependencies between actors.

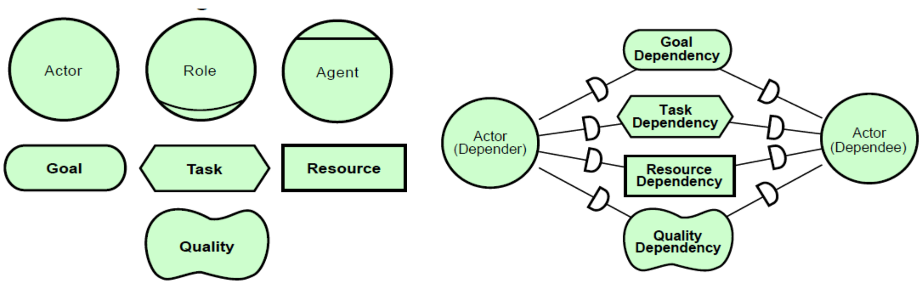


Figure 20: (Left) The main concepts used in SD and SR model and (Right) Dependency Types in an SD model based on the i* framework (Adali, Ozkan, Gilsing, et al., 2021)

After completing the SD model, more detail is added by extending it into an SR model.

1.3 Extend the SD model with a SR model

1.3.1 Determine the goals within the actor boundaries.

1.3.2 Determine the tasks to accomplish a given goal.

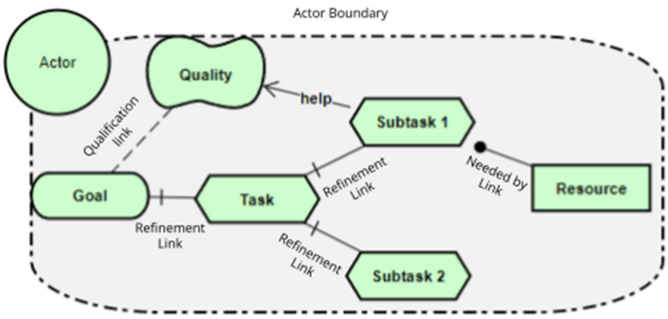
1.3.3 Determine the subtasks of the identified task(s).

1.3.4 Determine the resources that a task requires

1.3.5 Determine the quality requirements relevant for a goal, task or subtask.

The SR model (Figure 21) is also developed with the help of a few guidelines. First, goals within the actor boundary are given as actor value propositions in the SDBM/R. Secondly, tasks within the actor boundary are given as co-creation activities within the SDBM/R. Thirdly, quality elements within the actor boundary are given as adjectives in the SDBM/R. Furthermore, it is possible that further task and quality dependencies between actors are added during the creation of the SR model or more (sub)tasks are identified.

Figure 21: Representation of the Conceptual Elements and Dependency Types in an SR Model



(Adali, Ozkan, Gilsing, et al., 2021).

The output of this first step is the Strategic Dependency (SD) and Strategic Rationale (SR) model of a single co-created value proposition.

4.4 Step 2: Identify the Business Capabilities that Enable the Actor Value Proposition

The purpose of the second step is to identify the business capabilities of each actor that encapsulate the resource configurations necessary for that actor to fulfil its value propositions (Adali, Ozkan, Gilsing, et al., 2021). The input for this step are the SD and SR model created in step 1. To accomplish this step a template for the Service Domain – Business capability matrix is used. This template is depicted in Table A.1 in Appendix A.

The sub steps, when using the template are as follows:

2.1 Define a service domain for each first-level goal residing in the SR Model.

2.2 Define a service operation for all the tasks and subtasks refined under the first-level goal.

2.3 Identify the (future) business capabilities of all actors and create business capability descriptions.

2.4 Match the business capabilities to the service domains by marking the corresponding cell with an 'X' if the business capability facilitates the service operation under the service domain.

The output of this step is the Service Domain – Business Capability matrix according to the template and business capability descriptions according to the template in Table A.2 Appendix A.

4.5 Step 3: Define Modular Business Services Composed of the Business Capabilities

The purpose of the third and final step of the VP-BSIM is to compose the business capabilities of the service owner identified in the previous step into a set of modular business services (Adali, Ozkan, Gilsing, et al., 2021). The input for this step is the Service Domain – Business Capability Matrix. The first two sub steps (3.1 and 3.2) involve the service analysis with feature binding on the Service Domain – Business capability matrix.

The sub steps are as follows:

3.1 Perform feature binding analysis

3.1.1. Identify the service features

3.1.2. Identify the service feature binding times

3.1.3. Identify the service feature binding units

A service feature is a major functionality of the service that provides outcomes to the beneficiary of the service (Adali, Ozkan, Gilsing, et al., 2021). In the context of VP-BSIM, service features are given as business capabilities. A feature binding time refers to the point in time when the selected feature is bound to a service. In the VP-BSIM, two business capabilities share the same feature binding time when they share a service owner that facilitates the same service operation in the same service domain. A feature binding unit represents a set of business capabilities facilitating the same service operations at the same time.

3.2 Perform service analysis by determining the business services through composition of feature binding units according to the guidelines

3.3 Create a business service description for each business service.

During the service analysis, composed feature binding units are translated into business services. This is done by confirming their modularity according to the following guidelines:

1. The feature binding unit is self-contained
2. The feature binding unit is stateless
3. The feature binding unit is representative of a domain-specific service

Firstly, self-contained means that the business service should not need another business capability under another business service to execute. Secondly, stateless means that the business service should not require context or state information of another business capability under another business service. Finally, representative of a domain-specific service implies that it should provide an autonomous and unique business function.

The output of this step is a set of modular business service descriptions according to the template depicted in Table A.3 in Appendix A.

4.6 Step 4: Identify Potential Software Services Per Business Service Required for Openness

The purpose of the fourth step is to analyze the business services to identify which service operations performed by the focal organization should be supported by a digital platform and are required for openness towards third-party developers. The identified service operations are labeled software services, which are defined as product level requirements (i.e. goals) of the digital platform (Gorschek & Wohlin, 2006). For this step a heuristic is used based on the formal approach identified by (Kohlborn et al., 2009) as explained in Chapter 3.

The input of this step are the business services and their description as determined in the third step of the VP-BSIM. The heuristic consists of four sub steps and should evaluate each individual business service.

The sub steps are as follows:

4.1 Determine if the business service and specific service operations could be supported by a software service.

4.2 Identify software services required to open up the platform to third-party developers.

4.3 Create software service description table per software service.

4.4 Prioritize identified software services.

In sub step 4.1, the business services and specific service operations of the business services that could be supported by a software service should be identified. To do this, the software service identification template is used (depicted in Table A.4 in Appendix A). The template contains the identified business services and service operations in the VP-BSIM. If only a specific part of the service operation can be supported by a software service, the service operation needs to be decomposed. To complete this step, each identified software service gets a unique identification number (SS#).

In step 4.2 users of the method determine for each software service if it is specifically required for opening the platform for third-party developers. This is also noted in the software service identification table (last column). In step 4.3, each software service should be described via the software service description table depicted in Table A.5 in Appendix A. This contains information about the software service such as its functionality and the benefit the use of the software service would provide. Information about ecosystem actors and their relevant entities interacting with the software service are also to be defined. Finally in step 4.4, software services are given a priority for their future validation and development. This is also recorded in the software service description.

The output of step 4 is a set of prioritized software services and their descriptions that together could potentially function as a software platform.

4.7 Step 5: Identify Non-Functional Openness Requirements Per Software Service

Step 5 up and including step 8 have been adopted from Sadi & Yu (2017ab). Any modifications made to their method have been reported in Chapter 3. The purpose of the fifth step is to identify the relevant non-functional openness requirements for each software service. From here on, each main step of the method (Step 5 up and including Step 8) will be iterated for each individual software service. The input of this step is an individual software service and its description as defined in the software service description table in Step 4.

The sub steps are as follows:

- 5.1 Select a software service based on its priority.
- 5.2 Determine the business level openness requirements related to the software service.
  - 5.2.1 Select relevant business requirements from existing catalogue.
  - 5.2.2 Determine additional business level openness requirements if appropriate.
  - 5.2.3 Determine linking requirements.
- 5.3 Determine the system level openness requirements related to the software service.
  - 5.3.1 Select relevant system requirements from existing catalogue.
  - 5.3.2 Determine additional system level openness requirements if appropriate.
- 5.4 Determine general design concern openness requirements related to the software service.
  - 5.4.1 Select relevant general design concern requirements from existing catalogue.
  - 5.4.2 Determine additional general design concern system requirements if appropriate.
- 5.5 Create a NFR table with identified openness requirements.
- 5.6 Create Goal based non-functional openness requirement model.
  - 5.6.1 Place the software service and the related business requirements at the top.
  - 5.6.2 Decompose/refine system requirements and link to business requirements.
  - 5.6.3 Decompose/refine and link general design concerns
- 5.7 Determine the priority of identified openness requirements

The first step (5.1) is to select a software service to specify its non-functional requirements. This done based on the priorities given in step 4.

The aim of step 5.2 is to determine the relevant business level openness requirements related to implementing the software service. Business level openness requirements are the main motivations behind opening up a software platform. They are non-technical, social, business or organizational requirements that may influence the platform design indirectly. The business level openness requirements catalogue presented in Table E.1 of Appendix E and developed by (Sadi & Yu, 2017b) can be used in step 5.2.1 to select relevant requirements. Discussion of the ecosystem context and domain in which the software service will operate may lead to additional openness requirements being identified in step 5.2.2. The business level openness requirements may only need to be identified once and be reused for the evaluation of other software services. In Step 5.2.3 openness requirements are identified that link the business and system level requirements.

Step 5.3 follows the same pattern as step 5.2 but identifies the relevant system level openness requirements related to implementing the software service. System level openness requirements are technical, related to the quality of the software design and directly influence design decisions. The system level openness requirements catalogue presented in Table E.2 of Appendix E and developed by (Sadi & Yu, 2017b) can be used in step 5.3.1 to select relevant requirements.

Step 5.4 also follows the same pattern as step 5.2 but identifies relevant general design concerns related to implementing the software service. The general design concern requirements catalogue presented in Table E.3 of Appendix E and developed by (Sadi & Yu, 2017b) can be used in step 5.4.1 to select relevant requirements.

Each non-functional requirement should be accompanied by an entity description that shows which element of the digital platform is involved, such as the third-party content, or data. These entities follow from the software service description tables previously developed in Step 4 and depicted in Table A.5 Appendix A.

In step 5.5 the identified non-functional requirements are shortly described in the NFR table as depicted in the template Table A.6 in Appendix A. Then, in step 5.7, add a priority to each requirement in both the NFR table and the goal model depicted in Figure 22, choosing from non-critical, critical (!) and very critical (!!). The determination of priority is based on consensus within the focal organization.

In step 5.6, create a model for the non-functional requirements based on the i* goal modelling language as depicted in Figure 22. In step 5.6.1, start the model with formulating the software service as a 'goal'. Then connect the business level openness requirements as 'soft goals' below the software service via 'qualification' links. In Step 5.6.2, connect the system level openness requirements, also as soft goals, to the business level requirements via 'help' links. Also decompose/refine the system level requirements by relating them with 'help' links. The same steps should be repeated for the general design concern openness requirements in step 5.6.3, as depicted in Figure 22.

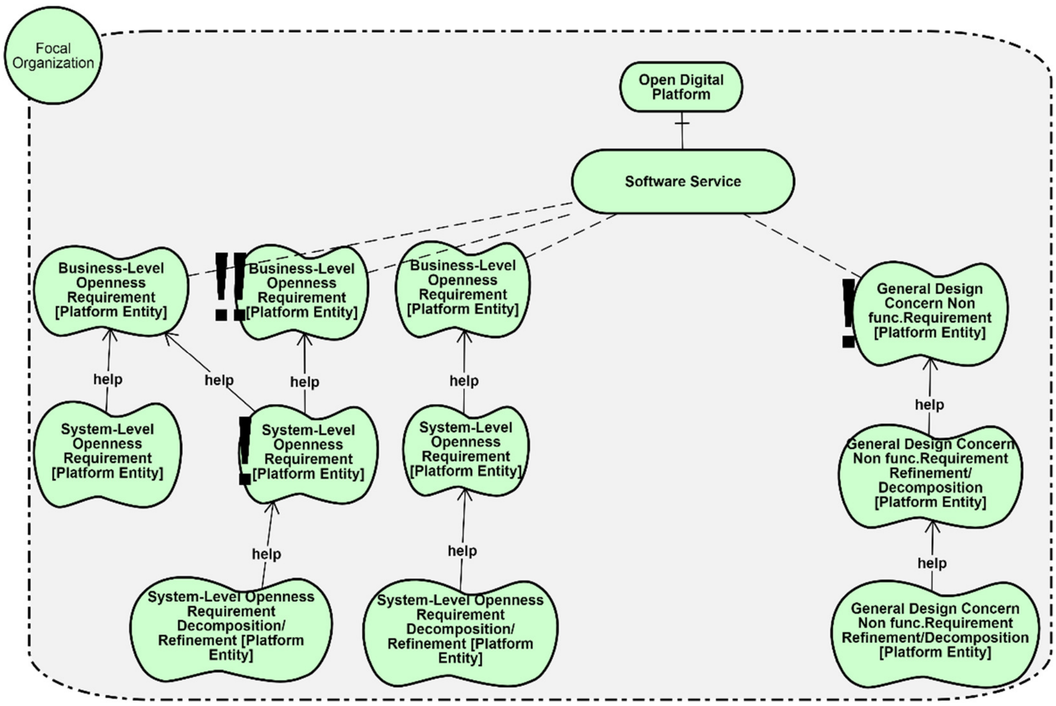


Figure 22: Example Goal model of the non-functional (openness) requirements of an individual software service based on the i* framework.

The output of step 5 consists of two parts. First, an NFR table with the identified requirements and a short description of their relation to the software service. Second, a goal-based overview for an individual software service consisting of relevant and refined business, system and generic design concern openness requirements.

4.8 Step 6: Identify Design Alternatives to Implement the Software Service

The purpose of the sixth step is to identify design alternatives that could implement an individual software service defined in step 4. The inputs for this step are the software services as defined in step 4 and the goal model as defined in step 5.

The sub steps are as follows:

6.1 Determine alternative design alternatives.

6.2 Create table with design alternative descriptions.

6.3 Modify goal model with design alternatives.

The aim of step 6.1 is to identify which design alternatives could implement the software service. At least two alternatives should be defined which have a different evaluation outcome on at least one non-functional requirement. The next steps are used to specify the design alternatives. In step 6.2 a Design Alternative Table as depicted in the template Table A.7 of Appendix A is created. A short description of how it would implement the software service is given.

In Step 6.3, the goal model from Step 5 is modified to include the design alternatives. The design alternatives are each modelled as 'task' elements and connected to the software service via 'OR-refinement' links to indicate that each alternative could possibly implement the service.

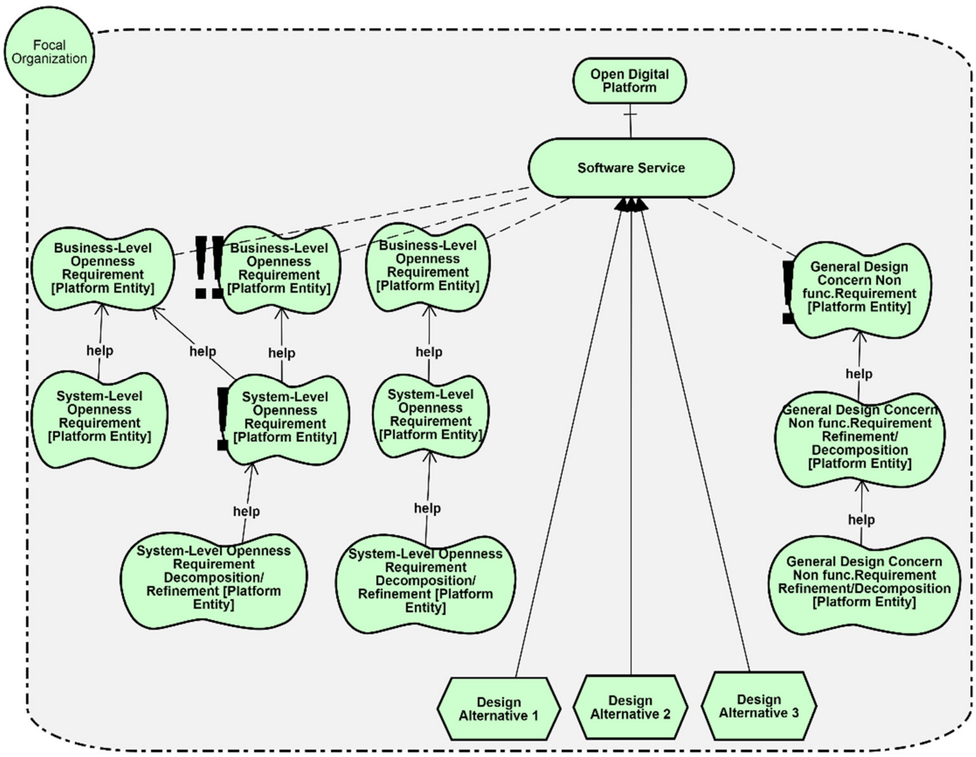


Figure 23: Example Modified goal model including the design alternatives of a software service.

The output of this sixth step consists of two parts. First a Design Alternative description table that and second the modified goal model from step 5 with an addition of the design alternatives (Figure 23).

4.9 Step 7: Evaluate the Design Alternatives

The purpose of the seventh step is to evaluate the design alternatives based on the non-functional requirements defined in step 5. The input of this step are the Design Alternative table and goal model defined in Step 6 and the non-functional requirements belonging to the software service defined in step 5.

The sub steps are as follows:

7.1 Evaluate Design Alternatives on the most refined non-functional requirements.

7.1.2 Create table to systematically evaluate design alternatives.

7.1.2 Modify goal model with initial evaluation labels.

7.2 Propagate evaluation labels.

7.3 Resolve evaluation labels.

The evaluation procedure will follow the i* based guidelines outlined by (Horkoff & Yu, 2009) for the assigning, propagating and resolving of evaluation labels. In step 7.1 the most refined non-functional requirements in the goal model are evaluated first. The most refined NFR evaluation table is created such as the template depicted in Table A.8 of Appendix A. Here a label (Satisficed(S), Partially Satisficed(PS), Conflict(C), Partially Denied (PD) or Denied(D)) is assigned to the design alternative for each non-functional requirement after consensus within the focal organization. The goal model is then modified to reflect these initial evaluation labels as depicted in Figure 24. The (Partially) Satisfied label (S, PS) represents the presence of evidence which is (insufficient) sufficient to satisfy the requirement. Partially denied and denied (PD, D) have the same definition with respect to negative evidence. Conflict (C) indicated the presence of both positive and negative evidence. Note that the assumption was that enough information (i.e. evidence) was available to make these judgements, therefore no 'unknown' label was permitted. A guiding question to reflect and choose a label is to think determine "what is the effect of using the design alternative on the non-functional requirement?".

In step 7.2 the initial evaluation labels are propagated upwards to the other non-functional requirements automatically if possible. This is done via the procedure described in Table 7. The incoming source label results in an automatic evaluation for the non-functional requirement.

Table 7: Automatic Propagation guidelines for evaluation labels

Source Label		**Contribution Link Type**
	Name	**Help**
✓	Satisficed	✓•
✓•	Partially Satisficed	✓•
⚡	Conflict	⚡
✗•	Partially Denied	✗•
✗	Denied	✗•

In step 7.3, the issue of non-functional requirements with multiple incoming labels is resolved. The multiple incoming labels are stored in a 'label bag'. These combined labels result in a single evaluation label for the non-functional requirement under consideration. This procedure is described in Table 8. For the cases not covered in this table, human judgement of the participants of the method is used based on their domain knowledge.

Table 8: Resolving guidelines for evaluation labels.

Label Bag Contents	Resulting Label
1. The bag has only one label. Ex: {✗•} or {✓•}	the label: ✗• or ✓•
2. The bag has multiple full labels of the same polarity, and no other labels. Ex: {✓, ✓, ✓} or {✗, ✗}	the full label: ✓ or ✗
3. All labels in the bag are of the same polarity, and a full label is present. Ex: {✓•, ✓, ✓•} or {✗, ✗•}	the full label: ✓ or ✗
4. The human judgment situation has already occurred for this element and the answer is known	the known answer
5. A previous human judgment situation for this element produced ✓ or ✗, and the new contribution is of the same polarity	the full label: ✓ or ✗

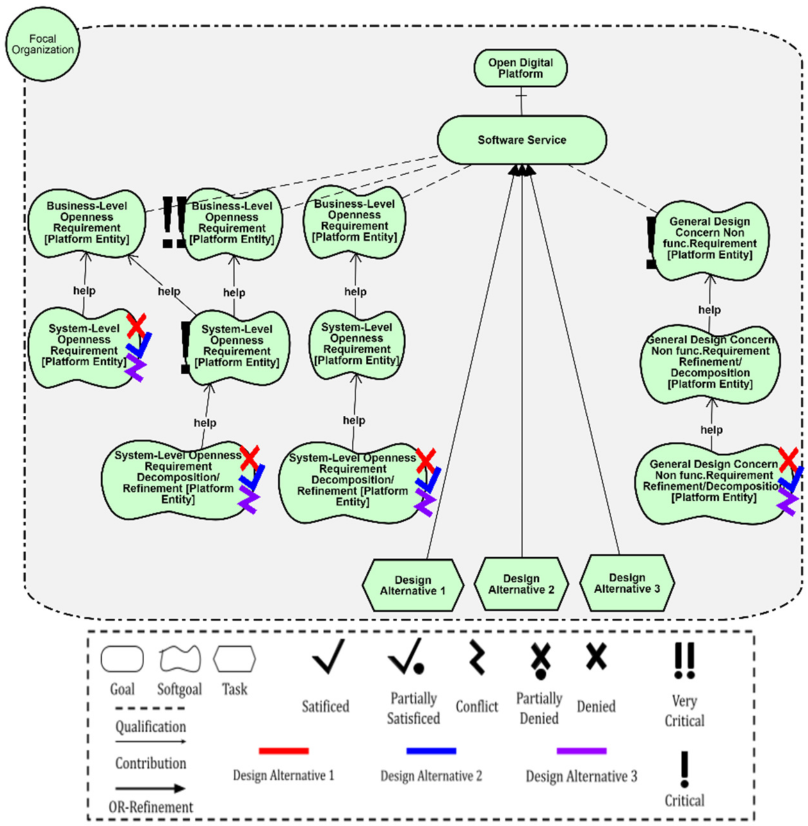


Figure 24: Example Goal model with most refined requirements evaluated.

The output of this step consists of two parts. First the goal model with evaluated Design Alternatives (Figure 24) and secondly the table which describes the rationale for the evaluation.

4.10 Step 8: Compare the Trade-offs between Design Alternatives and Make Design Decision

The purpose of the eight and last step is to compare the evaluated design alternatives, understand de trade-offs being made, especially between the openness requirements and general design concern requirements, and make a design decision by choosing a design alternative that would be most suitable to implement the software service. The input of this step is the evaluated goal model defined in step 7.

The sub steps are as follows:

8.1 Create a trade-off table with the highest priority non-functional requirements.

8.2 Reason between the trade-offs.

8.3 Make a decision by selecting a design alternative.

In step 8.1 a highest priority fulfilment table is created such as depicted in the template in Table A.9 of Appendix A. This table contains a selection of the highest priority non-functional requirements from the goal model (indicated by 'critical' and 'very critical'). The evaluation of the non-functional requirements also follow from the previously developed goal model. Therefore the evaluation labels follow the same patterns as before (Satisficed(S), Partially Satisficed(PS), Conflict(C), Partially Denied (PD) or Denied(D)).

The table is the starting point for the reasoning between acceptable trade-offs in step 8.2. Finally, in step 8.3, based on consensus between the participants, a design alternative is selected.

The output of this eight step is a single design alternative that would be able to implement the software service with an appropriate degree of openness.

5 Demonstration

For the demonstration phase of the design science research methodology, the method as described in Chapter 4 was executed for the demonstration case company described in Chapter 3.6. As input for the designed method, the service ecosystem value proposition for the case company was defined. Thereafter, the eight steps of the designed method were executed. As described in Chapter 3.4, the method was executed in consultation with two employees from the demonstration case company, which acted as the focal organization.

5.1 Input: Service System Value Propositions

The SDBM/R of the service ecosystem envisioned by the focal organization is depicted in Figure 25. First, the co-created value in-use as envisioned by the focal organization is determined (Step A.1). This has been formulated as 'High quality, cost-effective evidence-based healthcare'.

The second step is to define all actors that contribute to this service ecosystem (Step A.2). The first actor is considered to be end-user and main beneficiary of the service ecosystem: the healthcare provider. The focal organization is the service ecosystem facilitator. In this demonstration case, this role is provided by GOAL3. Furthermore, several core partners have been identified: research groups (e.g. universities), implementation & training partner's, Ministry of health, and insurance companies.

After the actors have been identified the first ring of the SDBM/R describes the contributions to the co-created value proposition of every individual actor (Step A.3). The healthcare provider ensures patients receive appropriate healthcare at the right time. The research groups provide new healthcare knowledge that healthcare provider can apply, the Ministry of Health and insurance company ensure new healthcare knowledge is approved and the implementation & training partner ensure that the healthcare provider receives and is able to apply new healthcare knowledge. The focal organization facilitates the service exchanges between actors in the service ecosystem. The second ring in the SDBM/R describes the activities that should be performed by the actors to enable their contribution (Step A.4). The healthcare provider makes clinical decisions during diagnoses and treatment of patients that are based on healthcare knowledge and collects patient health data in the process. This healthcare data is distributed by the focal organization to research groups. In turn, research groups use the received healthcare data to develop new healthcare knowledge.

In this SDBM/R, healthcare knowledge is primarily aimed at clinical decision support. Healthcare providers use clinical workflows when diagnosing and treating patients. Clinical workflows are the tasks and decisions aimed at managing a patient and can be represented by a clinical flowchart. During this clinical workflow, decisions have to be made based on certain (patient) parameters. For example, at a certain patient heartrate the decision is made to administer drugs to the patient to lower blood pressure. In this context, new healthcare knowledge refers to an entire new clinical workflow, or adjustments within existing clinical workflows such as certain parameters, aimed to improve patient health outcomes.

After new healthcare knowledge has been developed it is distributed by the focal organization to the Ministry of Health and Insurance Company. The Ministry of Health determines and approves healthcare policy, which should be understood as deciding which healthcare knowledge, based on proven effectiveness, may be used by healthcare providers. Insurance companies determine which healthcare knowledge to finance and monitor the expenditure of healthcare providers. Eventually, approved healthcare knowledge is implemented at the healthcare provider by the implementation & training partner. This partner trains healthcare provider employees and provides technical support.

The third and last ring of the SDBM/R describes the cost and benefits of each actor when contributing to the service ecosystem (Step A.5). As explained in Chapter 4, these can be both tangible and intangible in nature. In this service ecosystem, the healthcare provider receives healthcare knowledge and as a result can experience improved resource allocation in terms of, for example, employees and inventory by making more effective clinical decisions leading to among others shortened patient stays. In return, the healthcare provider sends healthcare data and a facilitation fee to the focal organization. Besides receiving healthcare data and knowledge the focal organization benefits by creating an 'ecosystem lock-in'. Research groups receive healthcare data and also provide a facilitation fee to the focal organization. The Ministry of Health and the insurance companies benefit by improved access to health data to develop policy and monitor care and spending. Implementation & training partners benefit by receiving a fee from the focal organization.

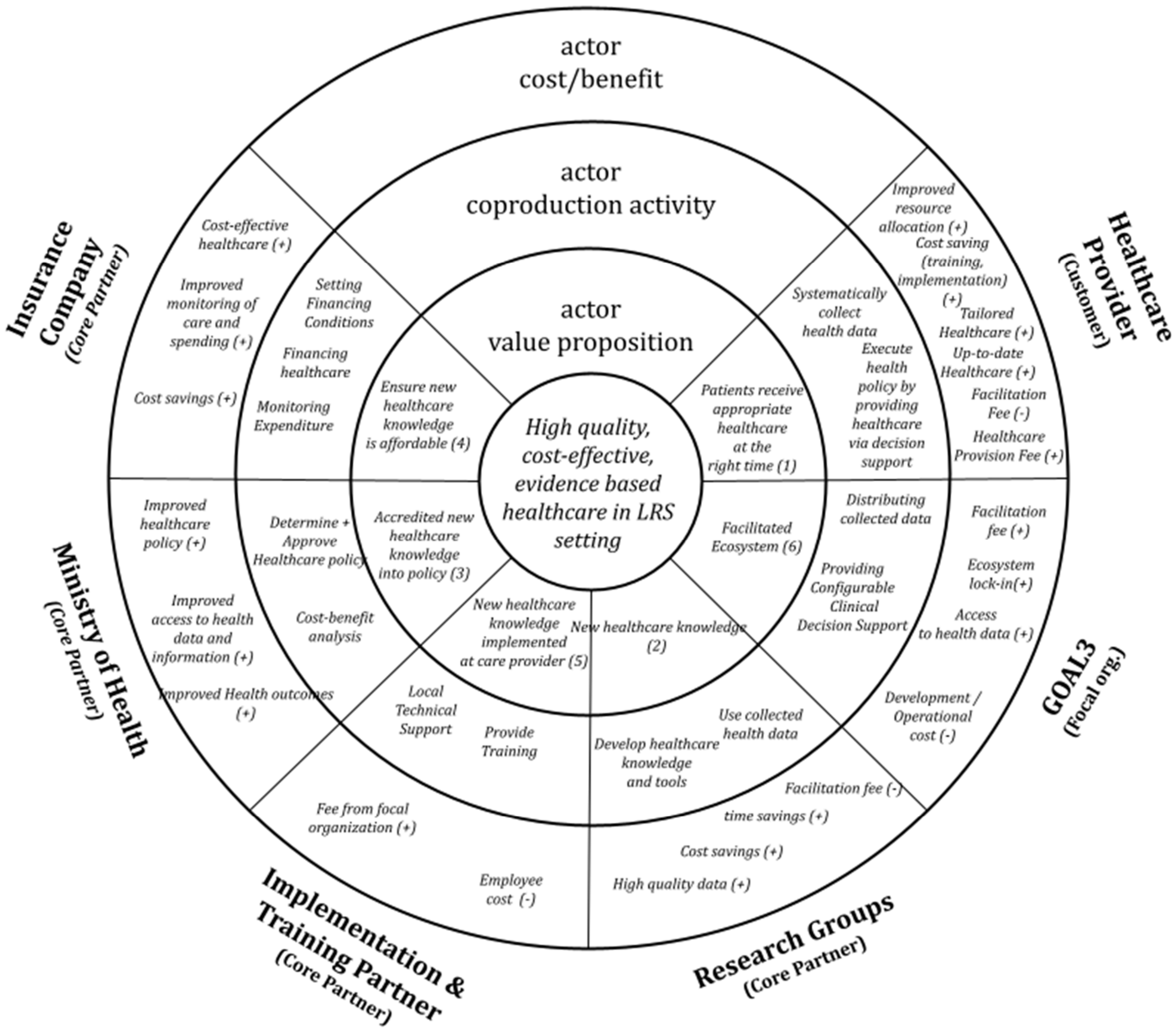


Figure 25: SDBM/R for the Service Ecosystem Envisioned by the Focal Organization

5.2 Step 1: Elicit the Goals that Actors Pursue in the Service System to Co-Create Value

After the value proposition of the service ecosystem and its contributors was defined, the SDBM/R was translated into an SD model to understand the relationships between actors (Step 1.2). Only one SDBM/R was created, so the selection of a SDBM/R has been omitted (Step 1.1). The created SD model is depicted in Figure 26. The actors in the SD model were illustrated as roles in the SD model (Step 1.2.1). The next step was to identify the dependencies and the type of dependencies between the actors (Step 1.2.2 and 1.2.3). For this the guidelines described in Chapter 4.3 were applied. The co-created value proposition was modeled as a goal dependency between the focal organization and the main beneficiary of the service ecosystem; the healthcare provider. Furthermore, actor co-creation activities became task dependencies between actors. Finally, tangible cost and benefits, such as healthcare data, were modeled as resource dependencies (i.e. exchanged resources) between actors.

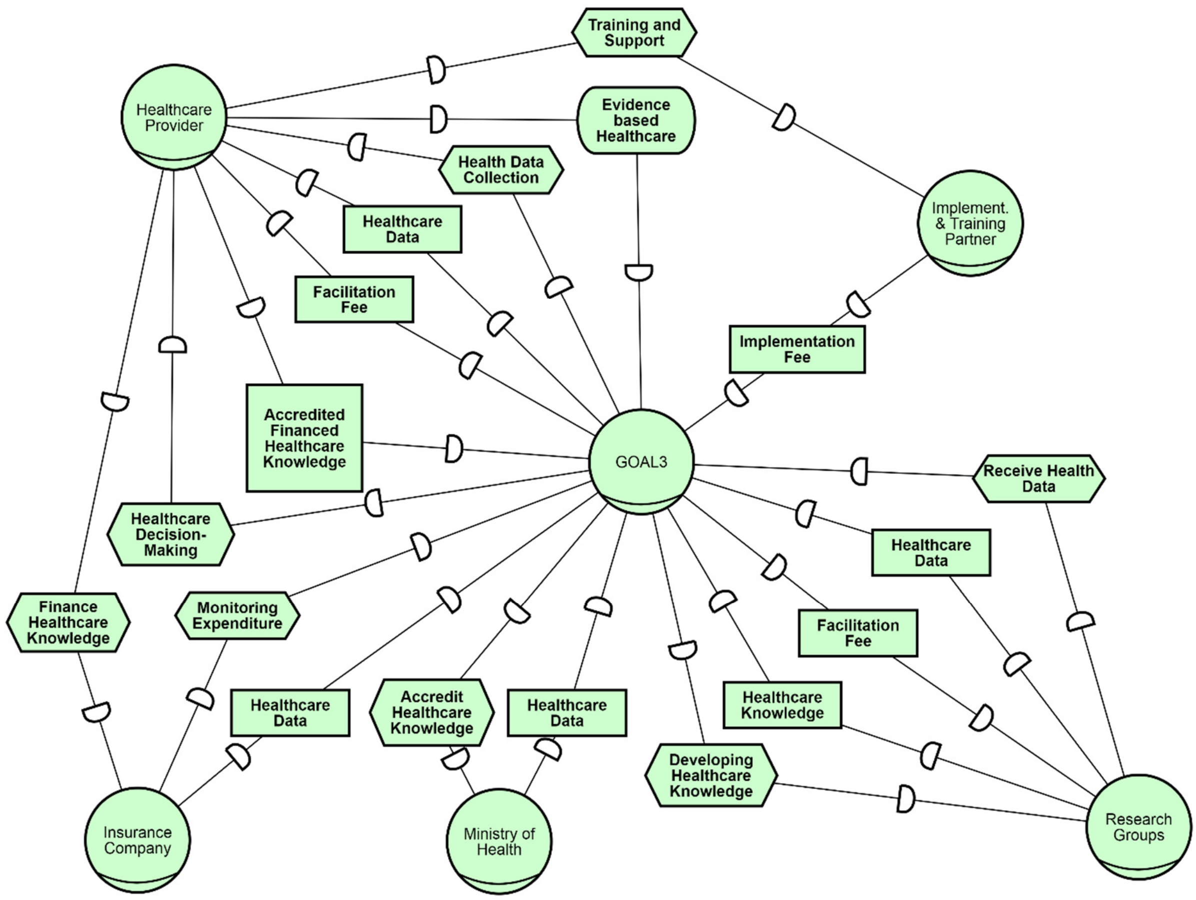


Figure 26: SD Model of the service ecosystem

After completing these steps, the SD model was extended into the SR model (Step 1.3). The SR model is depicted in Figure 27. Again the guidelines were used. The first step was to define the goal for each actor based on their contribution to co-created value proposition defined earlier in the first ring of the SDBM/R (Step 1.3.1). These goals are achieved by the tasks and subtasks described by the value co-creation activities defined in the second ring of the SDBM/R (Step 1.3.2 and 1.3.3). During these steps, new actor tasks were identified or existing tasks further refined, showing the explorative nature of the designed method.

For the focal organization these tasks were related to the goal of facilitating the service ecosystem. This role is not limited to storing and distributing healthcare data, but also to guide the creation of newly approved and affordable healthcare knowledge and eventual implementation at the healthcare provider. This starts with identifying research gaps which require new healthcare knowledge. For example, a high mortality rate under certain (patient) conditions is noticed in the healthcare data. The next step would then be to attract research groups (e.g. universities) to execute research that will lead to new healthcare knowledge. The focal organization (or research group) translates this new healthcare knowledge into new or adjusted (i.e. updated) clinical workflows that help the healthcare provider with clinical decision making. When a clinical workflow has been developed or adjusted, the implementation & training partner is able to teach staff (e.g. nurses, doctors) of the healthcare provider the new tasks and procedures of the (updated) clinical workflow. The clinical workflows may be adjusted to requirements or conditions of different nationalities in which the service ecosystem operates, which makes the clinical decision support provided by the focal organization configurable.

Finally, resource elements were added mainly to depict their origin or potential ownership as many tasks required similar resources for different purposes (Step 1.3.4) and quality elements were added within a few of the actor boundaries to emphasize requirements of actors (Step 1.3.5).

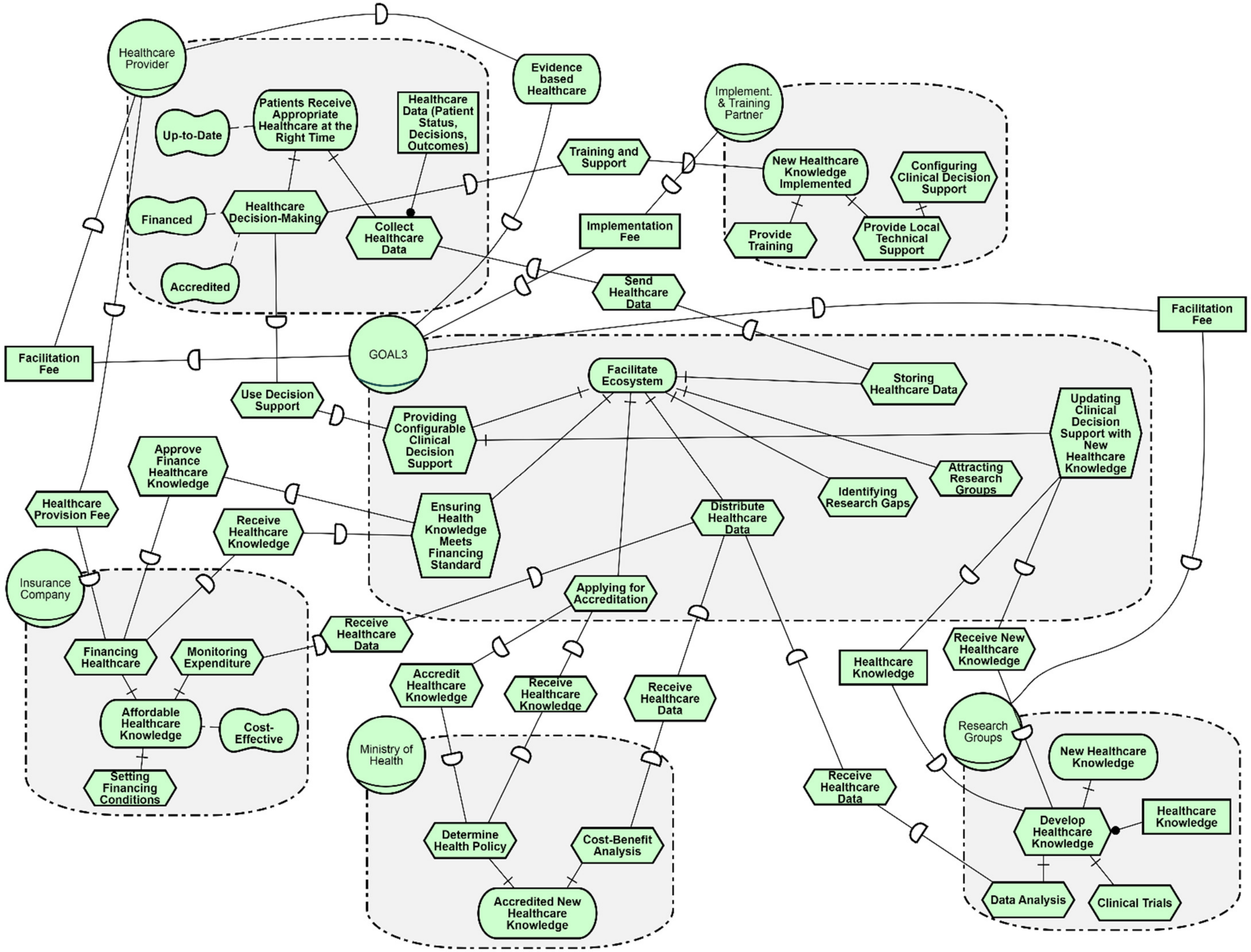


Figure 27: SR model of the service ecosystem

5.3 Step 2: Identify the Business Capabilities that Enable the Actor Value Proposition

In the second step of the method, a service domain – business capability matrix is created to identity which business capabilities actors need to enable their actor value propositions in the service ecosystem (Step 2.1). For the demonstration case company, the matrix is depicted in Table 9.

The service first-level goals and (sub) tasks under the first-level goal were directly translated from the SR model (Figure 27) to the service domains and service operations of the matrix (Step 2.1.1 and Step 2.1.2). The next step was to identify the (future) business capabilities (Step 2.1.3). To validate the selection, an informal interview was held with the CMO (chief marketing officer) of the focal organization. The business capabilities were described in a formal format depicted in Tables B.1, B.2, B.3, B.4, B.5 of Appendix B. For demonstrative purposes only the capabilities of the focal organization were specified. Furthermore, the business process elements of the table have not been specified due to time constraints. The last step was to match the business capabilities to the service operations in the matrix by marking the corresponding cell with an 'X'.

Table 9: Service Domain - Business Capability Matrix of the service ecosystem

Service Domains	Service Operations	Business capabilities																	
		GOAL3 (Focal Organization)					Healthcare Provider					Research Groups	Ministry of Health		Insurance Company		Implementation & Training Partner		
		Data Hosting	Data Analytics	Research Project Management	Product Development	Sales & Delivery (Incl. prod. delivery)	Diagnoses & Treatment Planning	Treatment & Monitoring	Resource Allocation	Documentation	Learning / Adapting	Clinical Trials	Health Policy Making	Data Analytics	Risk Assessment	Payment Processing	Field Support	Implementation Context Understanding (Change Management)	Sales context Understanding
Ecosystem Facilitated	Storing Healthcare Data	X																	
	Distributing Healthcare Data	X																	
	Identifying Research Gaps		X	X															
	Attracting Research Groups			X															
	Applying for Accreditation			X															
	Ensuring Health Knowledge Meets Financing Standard			X															
	Providing Configurable Clinical Decision Support				X	X												X	X
	Updating Clinical Decision Support with New Healthcare Knowledge	X			X														
Patients Receiving Appropriate Healthcare at the Right time	Making Healthcare Decisions						X	X	X		X								
	Collecting Healthcare Data									X		X							
New Healthcare Knowledge creation	Processing Healthcare Data											X							
	Developing Healthcare Knowledge											X							
Accredit New Healthcare Knowledge	Executing Cost-Benefit Analysis													X					
	Determining Health Policy												X						

(Table 9 Continued)

Service Domains	Service Operations	Business capabilities																	
		GOAL3 (Focal Organization)					Healthcare Provider					Research Groups	Ministry of Health		Insurance Company		Implementation & Training Partner		
		Data Hosting	Data Analytics	Research Project Management	Product Development	Sales & Delivery (Incl. prod. delivery)	Diagnoses & Treatment Planning	Treatment & Monitoring	Resource Allocation	Documentation	Learning / Adapting	Clinical Trials	Health Policy Making	Data Analytics	Risk Assessment	Payment Processing	Field Support	Implementation Context Understanding (Change Management)	Sales context Understanding
Affordable Healthcare Knowledge	Setting Financing Conditions														X				
	Financing Healthcare															X			
	Monitoring Healthcare Expenditure														X				
New Healthcare Knowledge Implemented	Configuring Clinical Decision Support																	X	
	Providing Training										X						X	X	
	Providing Local Technical Support																X	X	

5.4 Step 3: Define Modular Business Services Composed of the Business Capabilities

In the third step of the method the business services were defined. For demonstration purposes this was limited to specification of business services for the focal organization.

First the feature binding analysis was performed (Step 3.1). For this step, an initial selection was made of business capabilities needed to execute the same service operations in the same service domain at the same time (Step 3.1.2 and 3.1.1.3).

In step 3.2 the service analysis was conducted to ensure the modularity of the identified business services. However, since the granularity or level of abstraction of both service operations and business capabilities was deemed high, the analysis performed was lenient.

The business services are depicted in Table 10. Each color represents the combination of business capabilities and service operations which together form a coherent business service.

Finally, the business services were described in a formal format in Table C.1, C.2, C.3, C.4 and C.5 in Appendix C (Step 3.3).

Table 10: Business Services of Focal Organization in the service ecosystem

Service Domains	Service Operations	Business capabilities				
		GOAL3 (Focal Organization)				
		Data Hosting	Data Analytics	Research Project Management	Product Development	Sales & Delivery
Ecosystem Facilitated	Storing Healthcare Data	X				
	Distributing Healthcare Data	X				
	Identifying Research Gaps		X			
	Attracting Research Groups			X		
	Applying for Accreditation			X		
	Ensuring Health Knowledge Meets Financing Standard			X		
	Providing Configurable Clinical Decision Support				X	X
	Updating Clinical Decision Support with New Healthcare Knowledge	X			X	

- **Data Hosting** Business Service
- **Research Gap Identification** Business Service
- **Health Knowledge Initiation** Business Service
- **Health Knowledge Institutionalization** Business Service
- **Clinical Decision Support** Business Service
- **Sales & Delivery (HW)** Business Service

5.5 Step 4: Identify Potential Software Services Per Business Service Required for Openness

In the fourth step, the business services were analyzed to identify the service operations performed by the focal organization which should be supported by a digital platform and were required for openness towards third-party developers. The identified service operations were labeled software services and treated as product level requirements (i.e. goals) of the digital platform.

In Step 4.1 the software service identification table was created. This is depicted in Table 12. In the third, fourth and fifth column service operations were decomposed if needed and it was determined if the service operation should be executed as part of or supported by the digital platform in the form of a software service.

In Step 4.2 the software services were selected that were deemed to be required to open up the digital platform to third-party developers (i.e. research groups in the demonstration case). 'Storing' and 'distributing' healthcare data (SS-1 and SS-2) were considered required since this data could function as input for research groups to develop new healthcare knowledge, for example during a clinical trial. 'Providing an overview of identified research gaps' (SS-4) was also deemed required for openness, since this would give research groups a starting point for executing their research. Similarly, 'Providing Research Project Collaboration Environment' (SS-5) allows research groups to share progress during their research. Finally, 'Updating Clinical Decision Support with New Healthcare Knowledge' (SS-7) allowed new healthcare knowledge to be delivered to the healthcare provider (i.e. end user).

For demonstrative purposes only software service SS-7 'Updating Clinical Decision Support with New Healthcare Knowledge' has been specified and described in Table D.1 of Appendix D (Step 4.3). After the selection and definition of software services, a priority was given to the software service to indicate the urgency for development of the service (Step 4.4).

Table 11: Software Identification Table of the service ecosystem

Focal Organization					
Business Service	**Service Operations**	**Executable via Software Service (Yes/ Partially/ No)**	**(Optional) Decomposed Service Operation**	**(Optional) Executable via Software Service (Yes/ Partially/ No)**	**Required for Openness Towards Third-Party Developer (Yes/No)**
Data hosting	Storing Healthcare Data	Yes (SS-1)	-	-	Yes
	Distributing Healthcare Data	Yes (SS-2)	-	-	Yes
Research Gap Identification	Identifying Research Gaps	Partially	Analyze Existing Research Literature	No	No
			Identifying Anomalies in Healthcare Data	Yes (SS-3)	No
Health Knowledge Initiation	Attracting Research Groups	Partially	Providing Overview of Identified Research Gaps	Yes (SS-4)	Yes
			Providing Research Project Collaboration Environment	Yes (SS-5)	Yes
			Facilitate Research Funding	No	-
Health Knowledge Institutionalization	Applying for Accreditation	No	-	-	-
	Ensuring Health Knowledge Meets Financing Standard	No	-	-	-
Clinical Decision Support	Providing Configurable Clinical Decision Support	Yes (SS-6)	-	-	No
	Updating Clinical Decision Support with New Healthcare Knowledge	Yes (SS-7)	-	-	Yes
Sales & Delivery (HW)	Providing Configurable Clinical Decision Support	No	-	-	No

5.6 Step 5: Identify Non-Functional Openness Requirements Per Software Service

In the fifth step, the relevant non-functional openness requirements were determined for a single software service. SS-7 'Updating Clinical Decision Support with New Healthcare Knowledge was selected to be further specified in the remaining method steps (Step 5.1). The non-functional openness requirements accompanying this software service are depicted in Table 12 and Figure 31.

First the business level openness requirements were selected from the catalogue (Step 5.2.1). The catalogue is presented in Table E.1 in Appendix E. No additional business level openness requirements were determined (Step 5.2.2). Furthermore, a few so called linking requirements were determined for the identified business level openness requirements (Step 5.2.3). A similar procedure was used for the system level openness requirements and general design concerns related to openness (Step 5.3 and 5.3). Their respective catalogues are presented in Table E.2 and E.3 in Appendix E. The non-functional requirements and their relevancy for the software service were all described in Table 12 (Step 5.5).

Table 12: Non-functional Openness Requirements (NFR) for the Software Service

SS-7: Updating Clinical Decision Support with New Healthcare Knowledge		
Business level Openness requirement	Priority (-, critical, very critical)	Description
Market-related Time to Market [Third party content]	-	A market related aim of opening up the digital platform is to reduce the time to market for new healthcare knowledge developed by research groups (third party content).
Market-related New Markets [Platform]	Critical (!)	A market related aim of opening up the digital platform is to enter new markets by adding new healthcare knowledge of different healthcare domains (neonatal, maternity, etc.)
Market-related Community [Platform]	Critical (!)	A market related aim of opening up the digital platform is to create a community of research groups which can develop new healthcare knowledge (content) to bring to the platform.
Customer-related Stickiness [Platform]	-	One customer related aim of opening up the digital platform is to increase the difficulty of switching to another digital platform (by the customer) by providing decision support in multiple healthcare domains (neonatal, maternity, etc.)
Product-related Third Party Content Variety [Platform]	-	A product related aim of opening up the digital platform is to increase healthcare knowledge variety. By basing research on identified research gaps (needs), redundancy of similar research is reduced and resources of research groups can be used more effectively.
Finance-related Revenue stream [Platform]	Very Critical (!!)	A finance related aim of opening up the digital platform is to generate income with the platform by asking a facilitation fee from research groups (third party provider) and/or the customer/end-user.
Linking Network size [Platform]	-	This linking requirement bridges the gap between business and system level openness requirements. The network size (i.e. number of customers and third-party providers) influences market related aspects such as the revenue stream generated by opening up the platform. Network size can be stimulated by for example making it easier to adopt the platform.
Linking	-	This linking requirement bridges the gap between business and system level openness requirements. Innovative third-party content is required to reach

Innovative Third-Party Content [Platform]		new markets (i.e. new healthcare domains) and can be realized by among others the extensibility of the platform.
Linking Adoptability [Platform]	-	This linking requirement bridges the gap between business and system level openness requirements. Easier adoption of the platform should reduce the time to market of research executed by research groups (third party content). This can be realized for example by providing easy access to healthcare data which research groups use for their clinical trials.
System level openness requirement	**Priority** (-, critical, very critical)	**Description**
Configurability [Platform]	-	The clinical decision support and its (third party) content should be configurable for different nations or regions in which it will be deployed. This is deemed important because customers (i.e. healthcare providers) may have different needs or regulations.
Extensibility [Platform]	-	The clinical decision support should be easily extended by integration of third-party content.
Evolvability [Platform]	Critical (!)	The platform must be able to evolve its architecture and interfaces. The platform will inevitably undergo changes, adding more features and more options to open up to third party providers (i.e. research groups) for example by allowing them to build additional applications for clinical decision support.
Transparency [Platform]	-	The clinical decision support should be transparent in its functions, components and in the processing of third-party content. This is important for third party content providers (i.e. research groups) to be able to develop and test their content before it is deployed at healthcare providers.
Deployability [Third Party content]	Very critical (!!)	Third party content provided by the clinical decision support must be easily deployable. This includes being independent from other third-party content. In other words, newly developed content (i.e. healthcare knowledge) must not (negatively) interfere with the working of the existing clinical decision support (i.e. other clinical workflows).
General design concern requirement	**Priority** (-, critical, very critical)	**Description**
Security [Platform]	Very critical (!!)	The software service must include security protocols to ensure that new healthcare knowledge from a third party is not malicious.
Accountability [Third Party Content]	Critical (!)	The digital platform must make sure third-party content functions and performs as the third party intended. Since the use of the clinical decision support involves life or death situations this is deemed critical.
Proprietary Ownership [Platform]	-	It should be clear who owns (part of) the platform in terms of intellectual property rights of the application. This ensures responsibility for updating and supporting the clinical decision support.
Proprietary Ownership [Third Party Content]	-	It should be clear who owns the third-party content. This might be a requirement from third party content providers (i.e. research groups) before making their content (i.e. new healthcare knowledge) available to the platform.
Performance [Platform]	Very Critical (!!)	The software service must allow the digital platform to perform according to the expected and agreed upon service standards common for the healthcare domain.

In Step 5.6 a goal model was created to show the links between and refinements of the chosen non-functional requirements. This goal model is depicted in Figure 28. Adoptability[Platform], Network size [Platform] and Innovative Third-Party Content [Platform] are treated as linking requirements between the business level openness requirements and the system level openness requirements. To exemplify the model: the software service has a great impact on the Adoptability of the platform by third-party developers (i.e. research groups), since it is related to delivering their content to the end user (i.e. healthcare provider). Adoptability of the platform could be refined into Deployability of third-party content and Transparency of the workings of the platform.

Because of the delivery role of the software service, Adoptability directly influenced the Time to Market of Third-Party Content. Furthermore, Adoptability was deemed to influence the (growth of the) Network Size of the platform.

Finally, in Step 5.7 a priority was given to the non-functional requirements based on their importance and described in both Table 12 and the goal model (Figure 28). For example, New Markets [Platform] and Community [Platform] were deemed critical drivers to create early growth of the user-base of the digital platform via openness. A more specific example for the software service was the general design concern Security[Platform]. This was considered very critical, since throughout the process of content delivery, malicious actors may interfere. This is not allowed to happen because lives of patients depend on correct clinical workflows being implemented and used by the healthcare provider.

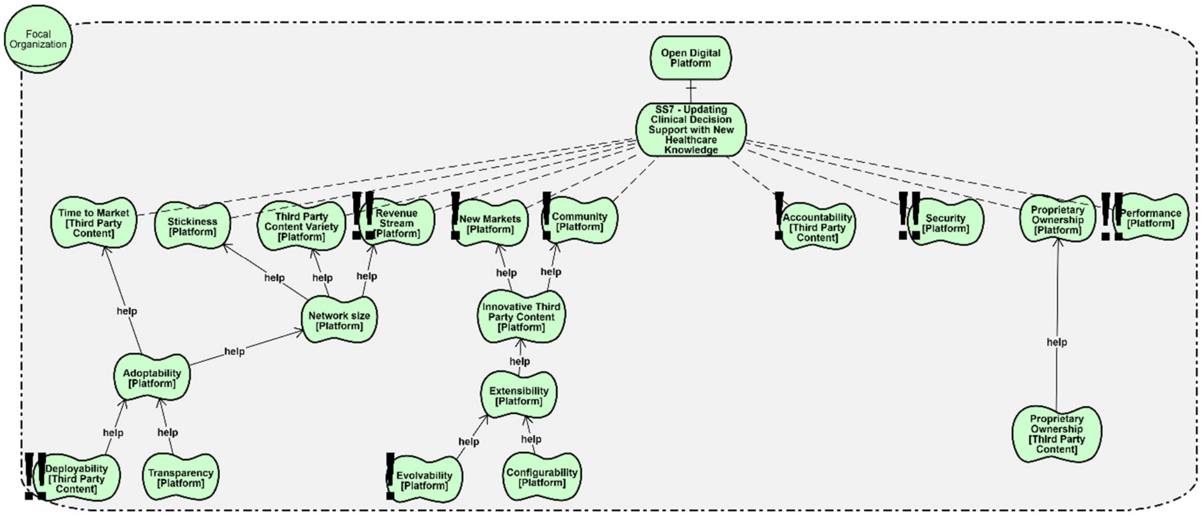


Figure 28: Goal model of non-functional requirements for the software service

5.7 Step 6: Identify Design Alternatives to Implement the Software Service

In the sixth step, the design alternatives were identified that could implement the software service 'SS-7: Updating Clinical Decision Support with New Healthcare Knowledge. The design alternatives are presented in Table 13 (Step 6.2). Each design alternative is characterized by a different degree of openness to implement the software service and was given a description (Step 6.1). Finally the goal model of Figure 28 with the non-functional requirements was modified to include the design alternatives (Step 6.3). This is depicted in Figure 29 (see next step) by the task elements.

Table 13: Design Alternatives for the software service

SS7- Updating Clinical Decision Support with New Healthcare Knowledge
DA1: 'Sideloading' (Open)
This design alternative only concerns the transfer of third-party content (i.e. in a file format) between two (local) devices. In this design alternative there is no controlled source from which files are coming. The technical details are mainly limited to enabling certain file formats to be imported in digital platform (i.e. the clinical decision support).
DA2: 'Third-Party Knowledge Library' (Semi-Open) (Example Noviguide - https://www.noviguide.com/)
This design alternative involves an online 'content store' which is a digital environment provided by the platform owner from where third-party content can be downloaded and imported in the clinical decision support by the healthcare provider or implementation & training partner. The role of the platform owner is limited to, for example, screening the third-party content for security reasons. An example in the healthcare domain is provided by the company Noviguide. Their clinical decision support software includes a 'clinical knowledge library' which contains knowledge developed by (third party) health experts in the form of clinical workflows which are to be followed when providing healthcare.
DA3: 'In-house' (Closed) (Example LogicNets - https://logicnets.com/)
This design alternative involves clinical decision support updates being created and performed directly by the platform owner. In this implementation, third party content providers (i.e. research groups) develop new healthcare knowledge, but the platform owner transforms this knowledge into useable artifacts for clinical decision support (e.g. changed clinical parameters or new clinical workflows). An example in the healthcare domain is provided by the company LogicNets. Their clinical decision support software only allows the creation and deployment of clinical workflows by an 'administrative/developer role'. No sideloading or third-party store is allowed or used to import third party content on the digital platform.

5.8 Step 7: Evaluate the Design Alternatives

In the seventh step, the design alternatives for the software service were evaluated based on the non-functional openness requirements identified in Step 5.

In Step 7.1 the design alternatives were evaluated on the most refined non-functional openness requirements. This was done with the help of Table 14 (Step 7.1.2). The (Partially) Satisfied label (S, PS) represents the presence of evidence which is (insufficient) sufficient to satisfy the requirement. Partially denied and denied (PD, D) have the same definition with respect to negative evidence. Conflict (C) indicated the presence of both positive and negative evidence. Note that the assumption was that enough information (i.e. evidence) was available to make these judgements, therefore no 'unknown' label was permitted.

Reasoning of label choices for a few of the labels are as follows. For the most open design alternative 'Sideloading', several labels were assigned because of the lack of control the platform owner has to verify (the use of) third-party content. However, labels were also more nuanced. For example, Deployability was labeled conflicting since it would be more difficult for the platform owner to test how content interacts with each other to guarantee performance, but if sideloading is done locally, it could be more usable in low resource settings where there is a lack of connectivity to deliver third-party content to the end-user (i.e. healthcare provider). Furthermore, Evolvability of the platform is impacted both positively and negatively. Positively, due to lower complexity of the delivery of content, and negatively, perhaps due to compatibility of older standards which might not guarantee that sideloaded content still functions as intended.

For the most closed design alternative 'In-house' most requirements were satisfied since it offered the platform owner most control. Exceptions were Transparency[Platform] and Evolvability[Platform]. Third-party content providers could experience this design alternative of delivery as less transparent as it leaves them with little control over implementation at the end-user (i.e. healthcare provider), however this would then need to be discussed at an earlier stage in the development of third-party content. Evolvability of the platform might be impacted negatively because all development would need to be executed by the platform owner. For example, if new data formats or standards are introduced, the platform owner would need to verify the standards for all content themselves instead of merely demanding these new standards to be used by third parties.

The design alternative of 'Knowledge Library' is the middle-of-the-road alternative, with labels judged to be between the other design alternatives.

Table 14: Design Alternative Evaluation on most refined non-functional requirements

SS7 – Updating Clinical Decision Support with New Healthcare Knowledge								
Design Alternative	*Deployability [Third Party Content]*	*Transparency [Platform]*	*Evolvability [Platform]*	*Configurability [Platform]*	*Accountability [Third Party Content]*	*Security [Platform]*	*Proprietary Ownership [Third Party Content]*	*Performance [Platform]*
DA1-'Sideloading' (Open)	C	C	C	PD	D	D	C	C
DA2-'Knowledge Library' (Semi-open)	PS	PS	PS	PS	S	PS	S	C
DA3-'In-house' (Closed)	PS	C	C	S	S	S	S	PS

S: satisficed / PS: partially satisficed / PD: partially denied / D: denied / C: conflict / U: unknown

Next, the goal model was modified with the initial evaluation labels (Step 7.1.2). This is depicted in Figure 29. The following method steps then evaluated the rest of the non-functional requirements, resolving issues if necessary according to the propagation and resolving rules of (Horkoff & Yu, 2009) explained in Chapter 4.8 (Step 7.2 and 7.3). Most labels were propagated automatically without explicit human judgement.

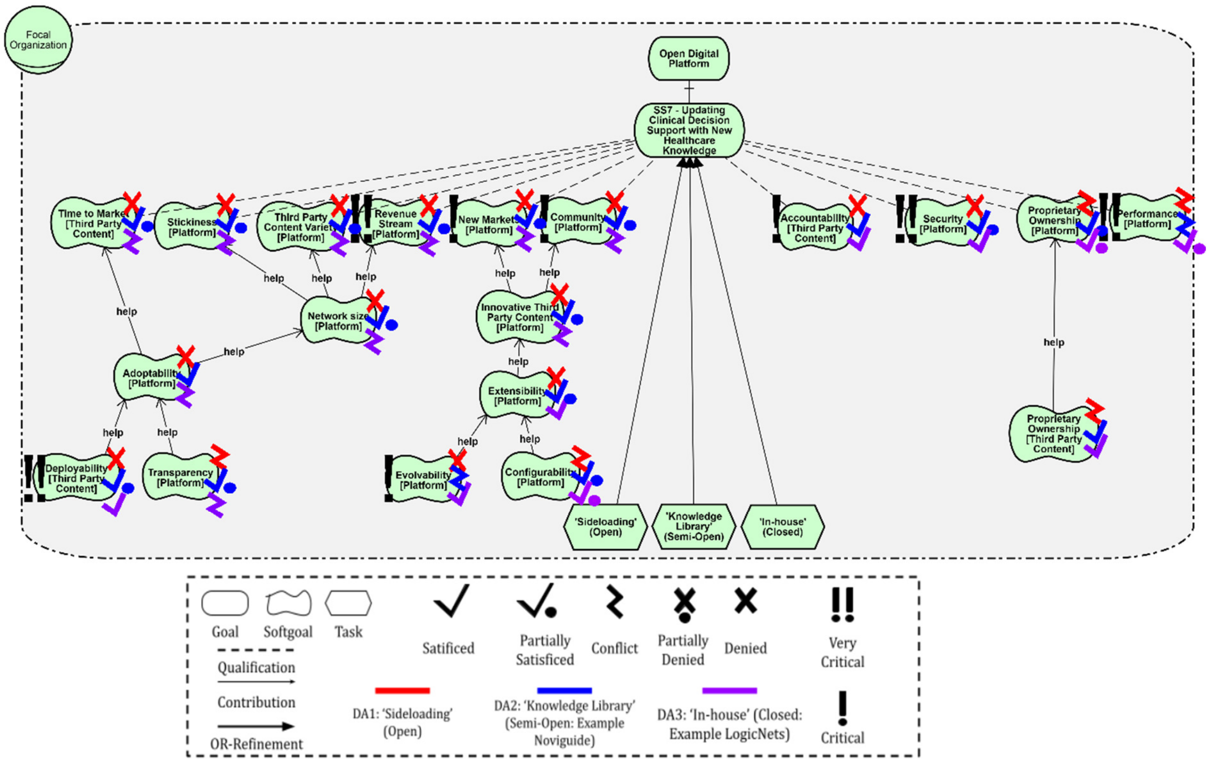


Figure 29: Evaluated Design Alternatives of Software Service

5.9 Step 8: Compare the Trade-offs between Design Alternatives and Make Design Decision

In the eighth and last step of the designed method, the trade-offs are compared between the most critical openness requirements and the general design concerns and a decision is made about which design alternative should be used. In step 8.1 a trade-off table is created with the highest priority non-functional requirements and their evaluation label from the goal model presented in Figure 29 from the previous method step. This is depicted in Table 15. For additional clarification, the labels have been provided with a color code.

The next step was to reason between the trade-offs that each design alternative makes and to identify the most suitable alternative (Step 8.2 and Step 8.3). The design alternative 'sideloading' performs poorly on both the critical openness requirements and the general design concerns. Therefore, this design alternative was not further considered. The differences between the other alternatives were less evident. The main trade-off seemed to be between platform owner control and the development freedom for the third-party content provider (i.e. research groups from universities). This could result in higher revenue streams for the platform owner and new markets (i.e. healthcare domains) being served by the digital platform when the 'Knowledge Library' alternative is chosen. While the 'in-house' alternative would ensure a higher level of security and performance of the platform. Furthermore, due to the complexity and criticality of the products and services, the platform owner may be incentivized to restrict access and lower the level of authority (i.e. reduce openness) towards third-party actors if the platform owner is liable for incidents and misuse (Broekhuizen et al., 2021). Moreover, third-party content providers (i.e. research groups from universities) may not be familiar with this type of digital platform and/or this type of collaboration with direct translation of their research into clinical workflows that will be implemented in practice at healthcare providers. Therefore, a more nuanced view would be to argue that during a 'startup' period of the platform the 'in-house' alternative could be implemented, while eventually transitioning towards the 'Knowledge library' or even continued use of both types of delivery in parallel. Parallel use of delivery methods might also be required because of the low resource setting in which the digital platform needs to operate. Healthcare providers that have access to connectivity could be easier served with delivery methods that rely on connectivity such as the 'Knowledge library' alternative.

Table 15: Trade-off Comparison for each design alternative

SS7 – Updating Clinical Decision Support with New Healthcare Knowledge								
	Business Level Openness Requirements			System Level Openness Requirements		General Design Concerns		
	Revenue stream [Platform]	New Markets [Platform]	Communit y [Platform]	Deployability [Third Party Content]	Evolvability [Platform]	Accountability [Third Party Content]	Security [Platform]	Performance [Platform]
Priority	Very Critical (!!)	Critical (!)	Critical (!)	Very Critical (!!)	Critical (!)	Critical (!)	Very Critical (!!)	Very Critical (!!)
DA1-'Sideloading' (Open)	D	D	D	C	C	D	D	C
DA2-'Knowledge Library' (Semi-Open)	PS	PS	PS	PS	PS	S	PS	C
DA3-'In-house' (Closed)	C	C	C	PS	C	S	S	PS

S: satisficed / PS: partially satisficed / PD: partially denied / D: denied / C: conflict / U: unknown

6 Evaluation

Following the demonstration, this chapter describes the empirical evaluation of the designed method and the assessment of the design objectives. The evaluation of an artifact is used to assess how well the method supports the solution to the problem (Peffers et al., 2007). Section 6.1 describes the empirical evaluation procedure and results. Section 6.2 assesses the design objectives defined in Section 3.2. Finally, Section 6.3 proposes improvements for the designed method based on the evaluation results.

6.1 Empirical Evaluation Protocol and Results

The empirical evaluation evaluated the artifact on the third design objective: *The method should be useful, easy to use, and encourage intention to use the method (DO3).* However, first the evaluation protocol and results are presented. The evaluation consisted of a focus group session with practitioners and a short questionnaire. The focus group was held with participants from different roles within the demonstration case company. This company functioned as the focal organization actor in the service ecosystem and would be the main user of the method. The focus group consisted of five participants. Details of the participants were given in Section 3.5. The selection of the participants was done such that all skills deemed necessary to execute the designed method were represented. This ranged from business developers to (software) requirement engineers. The setting of the focus group was online via group videocall with a planned duration of about 90 minutes.

The focus group agenda consisted of the following subjects:

1. Introduction to the research
2. Explanation of method steps with demonstration results
3. Discussing the method (steps) via open-ended questions
4. Introduction to the questionnaire

At the start of the focus group, participants were informed about the goal and agenda of the meeting. Next, the research was introduced by explaining the goal of the research and the designed artifact including an overview of the method steps. Following this, the method steps were explained individually, accompanied by results of the demonstration (Chapter 5) to provide examples. During this phase of the focus group, short questions were asked of the participants about the demonstration examples to test their understanding of the method steps. For example, by asking if they could identify additional actors in the SDBM/R (Input Step). The focus group was closed with an evaluation of the artifact. This consisted of a few general questions about the designed method to get their first impressions and identify potential improvements. The questions and all the important participant quotes in response are provided in Appendix E.

For the first discussion question, the researcher asked the participants about their first impressions of the method. This gave both positive and negative responses. Some participants immediately recognized the value of the method such as Participant 1 who explained ".. To me it feels like a pretty logical framework" and even seeing multi-use of the method such as Participant 5 who responded that it could ".. also be used to assess a platform that already exists". Other participants were overwhelmed such as Participant 2 who stated "Wow, quite extensive, maybe challenging to use, but for platform design this might be the right way to go through it thoroughly".

For the second discussion question, the research asked the participants about the strong and weak elements of the method. Starting with the strong elements, Participant 5 considered it ".. a good approach to start, I like that you attempt to structure this process" and Participant 1 stated their approval by saying "What I like about it is it brings together the business aspects of 'what do you want to do' with development" and "It structures the process and makes coming to decisions explicit and it helps to guide that process". Furthermore, Participant 5 liked that "it gave us a chance to think about it in a way we normally would not have". However, while others recognized strong elements, they also saw hurdles to adopting the method. For example, Participant 4 stated "There are a lot of tools, sub steps and terminology used in the method which takes time to digest and understand and get everybody on the same page, that seems like an entry hurdle"… "But I think that if this 'understanding' phase has been completed it would be a very strong method to visualize and communicate with each other". Participant 1 countered this by saying "I agree, but every iteration we use this method it would take a shorter time".

For the third and last discussion question, the researcher asked the participants what they would like to add to or change from the method. Here, participants mentioned several possible flaws of the method. For example, Participant 3 noted "Now we went through one design decision, but how to deal with multiple decisions influencing each other? What if there are conflicting requirements?". Participant 5 agreed by stating "The openness trade-offs are complex with many aspects to it .. if you approach a design element on its own you would decide differently than if you consider the platform as a whole". Participants also noted on the complexity of the method such as Participant 5 stating "There are too many steps, further simplification needs to happen if that's possible.. In practice you would probably never use such a complicated tool but something more simple" and "To me design comes down more to gut feeling and vision you have". Furthermore, minor remarks were made about individual method steps such as Participant 5 stating about the SDBM/R that "I liked the radar step, but a platform might use these actors in different ways ... For example in Ghana, the government is the healthcare provider, the insurance company and the training partner. Then it's just GOAL3 and the government … How do you design your platform per situation? Or should it account for all of them?". However, Participant 1 also noted that for true reflection on this question ".. we need to go through these steps ourselves, to test how it [the method] works and if it works".

Furthermore the short questionnaire depicted in Table 16 was presented and the participants were expected to complete this questionnaire after the focus group. As described in Section 3.5, this questionnaire used the design evaluation criteria from TAM (Davis et al., 1989) to measure the utility of the designed method. Questions were adapted to fit the artefact. The questionnaire consisted of 8 items for perceived usefulness, 3 for ease of use and 2 for the intention to use. Before filling in the questionnaire, the participants were asked to determine if their role within the company mainly involved business or engineering tasks. The table also includes the responses of the questionnaire. Under the each of the possible answers (1-5) the count of responses is included (e.g. 1x means one response). Furthermore, in the last three columns, the average scores are noted per question. The averages are also given for participants from a specific user group (e.g. B-average means the average score of business developers).

Results of the questionnaire seem aligned with the questions discussed in the focus group, showing variation in responses. Furthermore, the average perceived usefulness score was 3,4 while the average ease of use and intention to use were 3,1 and 3,2 respectively.

Table 16: The questionnaire including results.

							Averaged Results Per User Group		
	Question	Totally Disagree (1)	Disagree (2)	Neither agree nor disagree (3)	Agree (4)	Totally agree (5)	B-Average	E-Average	Total Average
Perceived Usefulness	I found the method to systematically stimulate and facilitate discussion of digital platform design.				4x	1x	4	4.3	4.2
	I found the method helpful in clarifying the value of an (open) digital platform.			1x	4x		4	3.7	3.8
	I found the method helpful in determining the appropriate degree of openness of software services (i.e. product level requirement) of a digital platform.			4x	1x		3	3.3	3.2
	• I found the method useful to determine which software services (i.e. product level requirement) could potentially be included in the (open) digital platform.		2x	1x	1x	1x	2.7	3.3	3.2
	• I found the method useful to determine which non-functional (openness) requirements would be relevant for a certain software service (i.e. product level requirement).			3x	2x		3	3.7	3.4
	• I found the method useful to understand the openness trade-offs that different design alternatives impose on the software service (i.e. product level requirement).			3x	2x		3.3	3.7	3.4
	I found certain methods steps to be missing when trying to determine the appropriate degree of openness of a software service (i.e. product level requirement) of a potential digital platform.		2x		3x		3.3	3.3	3.2
	I found certain method steps to be redundant trying to determine the appropriate level of openness of a software service (i.e. product level requirement) a potential digital platform.	1x		2x	2x		2.7	2.7	3
Perceived Ease of Use	I found the steps of the method clear and understandable		1x	2x	2x		3.3	3	3.2
	I found the method easy to apply in our business context (i.e. healthcare in low resource setting)		1x	3x		1x	3.7	2.7	3.2
	Learning to use the method will not be hard for me	1x	1x	1x	2x		3	2.7	2.8
Intention to use	I found the method to fit with our organization and way of working		1x	1x	3x		3.7	3.3	3.4
	I intend to use (part of) the method when I need to make decisions during the design an open digital platform		1x	3x	1x		3.3	3	3

B-Average = Average score for participants with business-oriented role or background

E-Average = Average score for participants with engineering-oriented role or background

Total Average = Average score over all participants

6.2 Assessment of artifact design objectives

Based on the demonstration case and the empirical evaluation, the design objectives have been assessed. As stated in Section 3.2, the designed artifact needed to satisfy these design objectives to the best of its ability to be able to help solve an instance of the problem. The first two design objectives (DO1 and DO2) have mainly been self-assessed, while the third design objective (DO3) was explicitly evaluated with practitioners as described earlier in Section 6.1.

DO1: The method should be value proposition driven.

To satisfy this objective, step 1 up and including step 3 of the designed method were adapted from the VP-BSIM of Adali et al. (2021). This was demonstrated in Chapter 5. The VP-BSIM steps ensured that focal organizations could use the designed method without having a prior digital platform concept or operational platform in place. Furthermore, the VP-BSIM ensured that the desired co-created value proposition was considered throughout the method. Every software service (i.e. product level requirement) of the digital platform was linked to a business service and eventually to the overall co-created value.

Although not explicitly evaluated during the focus group, some comments could be related to this first design objective. For example, participants of the focus group noted that they "liked the SDBM/R step" (Participant 5), but that it was missing nuance since the same actors could have different roles in different situations. Moreover, the granularity of the defined business capabilities and service operations was relatively high which made the application of the service analysis rules by the researcher, challenging. Furthermore, a general comment from participants was that there are too many steps in the designed method, and it would benefit from simplification. For example, Participant 5 stated "There are too many steps, further simplification needs to happen if that's possible.. In practice you would probably never use such a complicated tool but something more simple" and "To me design comes down more to gut feeling and vision you have". Furthermore, not all participants fully understood the value proposition concepts as they deemed them outside of their expertise. For example, Participant 4 stated ".. it only started to make sense when talking about requirements". This participant later acknowledged that if they took time to dive into the method "..it would be a strong method to visualize and communicate with each other". Participant 1 agreed with Participant 4 that "it requires a lot of different backgrounds to align on a topic", but also liked this aspect of the method and mentioned that it might be required to make their ideas tangible.

Based on the observations by the author of this research and the related comments by participants of the focus group, this design objective is considered largely fulfilled by the author.

DO2: The method should define explicit trade-offs to related to a certain degree of openness.

To satisfy this objective, step 5 up and including step 8 of the designed method was adapted from Sadi & Yu (2017ab). This was demonstrated in Chapter 5. The method by Sadi and Yu (2017ab) used non-functional requirements relevant to openness of a digital platform to represent openness trade-offs of implementing a single software service (i.e. product level requirement). The non-functional requirements were identified on the business and system level. The business level consisted of strategic, organizational level requirements while the system level was more preoccupied with technical limitations of the environment in which the platform would be deployed. This allowed for trade-offs within and between the business and system level.

A self-reflection by the author of this research on this second design objective is that the method steps were very sensitive to human judgement. Results could be impacted by the interpretation of design alternatives due to their high-level conceptualization, the selection of non-functional requirements, the importance of requirements and the judgement of fulfillment. Furthermore, the judgement of fulfillment of non-functional requirements was dependent on the group composition of users of the method.

Although not explicitly evaluated during the focus group, some of the participant's comments could also be related to this second design objective. For example, participants acknowledged that different elements of the platform could have different degrees of openness, with Participant 3 stating that "Not everything might require openness, for example, payment processing elements may need to be designed more closed". Furthermore, participants of the focus group seemed to understand how each design alternative identified in Step 6 could be considered a different degree of openness. Moreover, although the refinement and coherence between non-functional requirements seemed logical according to the participants, the Participant 3 noted that the priority distinction between to non-functional requirements may not be precise enough and it was doubted if refined requirements should have equal weight over the label of their 'parent' when applying the automatic propagation procedure. Finally, while the decision-making process was understood for a single software service by the focus group participants, there were doubts if decisions would be influenced by analyzing the platform as a whole. For example, Participant 5 stated that "Openness trade-offs are complex with many aspects to it and when you design an element on its own you might decide differently than if you consider the platform as a whole" and Participant 3 asked "How to deal with multiple design decisions influencing each other? What if there are conflicting requirements?".

Based on the observations by the author of this research and the related comments by participants of the focus group, this design objective is considered largely fulfilled by the author.

DO3: The method should be useful, easy to use, and encourage intention to use the method.

To assess this design objective the results from the questionnaire in Table 16 were used. Each of the scores on the design evaluation criteria of TAM is shorty discussed. Overall the method was received with mixed results by the participants. This seemed to be influenced by two main aspects. First, the limited amount of time taken for the focus group with participants stating they were "overwhelmed, I did not expect this many steps" (Participant 5) or felt they needed "To go through these steps ourselves "(Participant 1) to test and fully understand it. Secondly, the limited experience of participants with a platform design process, making participants more cautious in their judgement. This could be observed in Table 16 by the questions regarding redundancy or missing method steps, which scored relatively neutral and showed minimal differences (3 vs 3.2).

The average score for perceived usefulness is 3.4. This relatively neutral score is aligned with the diversity of comments made by participants such as "it's a good approach to start, I like that you attempt to structure this process.." (Participant 5) and "It gave us a chance to think about it in a way we normally would not have" (Participant 5), but also that it still left them with fundamental unanswered questions about the platform. These last comments indicate that the openness discussion might need to take place at a later stage in the design process, especially when it involves technical specifications. The method steps adapted from the VP-BSIM were scored most positively. Especially the first few steps of the VP-BSIM are relatively friendly towards practitioners since it could be perceived as less abstract. However, participants stated that some of the method steps did not consider the complete complexity of the dynamics between actors and the changes in dynamics when a platform would be deployed in different countries.

The average score for ease of use is 3,1. This score is also aligned with comments made by participants during the focus group, especially those with an engineering role or background. They had more difficulty with applying the method to the focal organization business context.

The average score for intention to use is 3.2. This too was expected and seemed to have been influenced by the two aforementioned aspects regarding limited time to understand the method and limited experience with platform design.

6.3 Directions for Improvement

The results of the evaluation and assessment of the design objectives showed several opportunities to improve the designed method.

First, from the focus group and questionnaire results could be derived that step 4 of the designed needed improvement. It was either not clear enough which activities of the service ecosystem could be enabled by a digital platform and/or how these platform design elements formed a coherence. This step is critical for the designed method, since it links the VP-BSIM by Adali et al. (2021) with the method from Sadi & Yu (2017a). Options for improvement range from refining the existing step or to entirely redesign this step based on an additional literature review. When refining the existing step a few improvement directions could be implemented. First, the service identification template (Table A.4 Appendix A) and its application could have a better description. This makes it easier for method users to understand and apply. Second, another sub step could be added in which users of the method create consensus on which digitalized or automated service operations could form one coherent digital platform.

Second, participants of the focus group noted that the outcome of the decision-making process of which design alternative to choose to implement the software service could be influenced by considering all the design decisions to make as a whole. This would show if there are any conflicts between the individual design choices. The current method only treats design decisions individually. This is a challenging issue, since the designed method relies on decomposition to simplify the design process. Furthermore, software services are supposed to be based on the business services and these are meant to be self-contained. However, an improvement to the method could be to add an additional step after step 8. This additional step could focus on two aspects. The first aspect would be to consider the chosen design alternatives for multiple software services and reflect on a high level about their compatibility. If two design choices are deemed not compatible a new design alternative needs to be chosen for either or both. The second aspect would be to compare the non-functional requirements and corresponding evaluation labels in the goal models of the chosen design alternatives. The user of the method could then set a threshold for the combined evaluation of this non-functional requirement. For example, if multiple software services consider security as a critical non-functional requirement, the chosen design alternatives for these software services are not allowed to all be labeled as denied.

Third, some participants of the focus group noted that the designed method contained too many steps. This reduced the ease of use for the method since a single iteration of the method would take too much time and too many meetings to align and make the design decisions. Therefore, the issue was that the method was deemed not lightweight enough. The challenge here is that the reduction of the number of method steps might decrease the robustness, reliability and traceability of the method. Possible improvements without drastically changing the method are as follows. First, the method-overview (Figure 18) and method steps more clearly where the method could be split up and executed in isolation and under which conditions. For example, the method is value proposition driven, which implies that when the user of the method already has a general idea of the digital platform concept, the input step and/or Step 1 up and including Step 3 could be skipped. However this also impacts step 4 as this uses business services and service operations as input. Second, the method (steps) should give a description on how each is iterative and thus do not need to be perfectly executed at once, saving time. The method steps are currently explained linearly. Third, the individual method steps should more clearly show their relationship and relevance to the eventual trade-off decision making. This could convince users of their necessity. Finally, in support of the previous improvement, the method-overview should (Figure 18) show which specific tools and inputs/outputs (i.e. templates and models) are used or developed within each method step.

7 Conclusion

Digital platforms have become a promising technology applied in a multitude of business domains and contexts. These platforms enable value co-creation for a complex network of actors, also known as the service ecosystem. The design of a digital platform is a complex task consisting of many concerns. Openness is considered a major concern during the design of a digital platform, especially when the platform is intended for service innovation with the help of third-party developers. In such a case, design decisions affect the design and development of the platform offering (i.e. value propositions). Therefore, it is crucial to determine the appropriate degree of openness early in the platform's design process when value propositions are designed.

Previous literature has described the benefits and risks of openness and prescribed which design elements may expand or restrict openness, and various trade-offs that have to be made, but not many design artifacts exist that guide practitioners in determining platform openness early in the platform design. Literature from service system engineering provided methods useful for the value proposition design and defining the value co-creation of a complex network in which a platform operates. Furthermore, no methods besides Sadi & Yu (2017ab) were identified which focused on determining the appropriate degree of openness. However, their method assumed that either a digital platform was in place and operational or that all functionality of the platform is understood and known before making openness decisions. Thus, no coherent integrated method existed that guided the practitioner in determining the appropriate degree of openness during the design of a digital platform from scratch (i.e. its value proposition design).

Adali et al. (2021) created the Value Proposition driven-Business Service Identification Method (VP-BSIM) to define business services from a set of value propositions. This was considered a good starting point for designing value propositions of the service ecosystem and bringing them into practice. However, the method did not identify which business services or service operations could be enabled by a digital platform and was not suitable yet to determine the appropriate degree of openness. Therefore, the objective of this research was *to develop a digital platform design methodology that is value proposition driven and supports the determination of the appropriate degree of platform openness by making trade-offs explicit.*

To achieve this, the service (eco)system perspective based on S-D logic was applied and a design science research methodology was followed (Peffers et al., 2007). During the design and development phase, the VP-BSIM (Adali, et al., 2021) was extended with the adapted NFR approach (Sadi & Yu, 2017ab) by adopting a situational method engineering method (Ralyté et al., 2003).

The final designed artifact is a method consisting of eight main steps and an input step. In the input step, the co-created value of the service ecosystem is determined by applying the Service Dominant Business Model Radar (SDBM/R) from Turetken et al. (2019). This is used as input for the VP-BSIM. The first three main steps follow from the VP-BSIM. In the first main step (1), the goals that actors pursue in the service ecosystem and the dependencies between actors to achieve these goals are determined via Strategic Dependency (SD) and Strategic Rationale (SR) models. In the second main step (2), business capabilities required to enable the goals and tasks actors pursue in the service ecosystem are defined. The SR model is translated into the Service Domain – Business Capability Matrix and (future) business capabilities of actors are matched to service domains and service operations (i.e. tasks) performed in the service ecosystem. In the third main step (3), coherent modular business services are composed of the business capabilities and service operations. A service analysis is performed combined with a feature binding analysis to ensure the business services are modular.

The fourth main step (4) is the designed heuristic based on Kohlborn et al. (2009) to determine software services which form design requirements of the digital platform. To do so, each service operation of the identified business services is examined and determined if they could be supported by information technology. This is followed by a step to determine if the software service is required for openness towards an actor using the platform (e.g. third-party developers).

The fifth up and including the eight main step of the designed method follow from Sadi & Yu (2017ab). In the fifth main step (5), non-functional openness requirements are determined for a single software service and depicted in a goal model (i.e. interdependency graph). In the sixth main step (6), multiple design alternatives are determined which are able to implement a software service (i.e. requirement). Each identified design alternative needs to be characterizable by a different degree of openness. In the seventh main step (7), the design alternatives are evaluated by assessing if they meet the non-functional openness requirements determined in the fifth main step. For this step the goal model evaluation procedure from Horkoff & Yu (2009) is applied by Sadi & Yu (2017ab). In the eighth and last main step (8), the scores on the most critical non-functional openness requirements are compared between the design alternatives to reason about the openness trade-offs. The chosen design alternative implements the software service (i.e. requirement) with a degree of openness judged to be appropriate by the platform owner.

The designed method was demonstrated with a single iteration of the method steps with input from the demonstration case company. Evaluation of the designed method consisted of a focus group and a questionnaire that followed the design evaluation criteria from the Technology Acceptance Model (TAM). The focus group was organized to show and discuss the method with employees of the demonstration case company who are the intended users of the method. Results of this evaluation were mixed. Not all of the design objectives set for the artifact were fully met and results from the questionnaire have indicated that additional design and development cycles of the design science research method are needed to improve the artifact.

7.1 Research implications

This research focused on the early steps in the formation of digital platforms and took an explicit design perspective to platforms. More specifically, it aimed to create a design artifact that could guide practitioners in determining the appropriate degree of openness of a digital platform concept. To achieve this, the VP-BSIM by Adali et al. (2021) was extended with the method proposed by Sadi & Yu (2017ab). To the best of the author of this research's knowledge, no prior research existed to explicitly support the practitioner in these activities in an integrated method at a similar stage of platform design.

Other research implications are related to the method by Sadi & Yu (2017ab). Through this research, the method they designed has received another round of validation. Moreover, their method has been applied from a broader perspective. First, the method has been exposed to a non-embedded platform. Non-embedded platforms may contain a higher-level complexity from a socio-technical view as more actors could be involved. Second, the method has been applied with input from a higher level of abstraction; software services defined as product level requirements instead of using functional requirements. A higher level of abstraction was deemed useful for a lightweight method. Based on the limited demonstration and evaluation, the method by Sadi and Yu (2017ab) seems to be able to handle the higher level of abstraction. The designed method still allows for decomposing these higher-level requirements into functional requirements.

Finally, the designed method and demonstration case might contribute to the creation of design knowledge for business to business (B2B) digital platform's, since this area of research has been underexposed compared to consumer oriented platforms, according to Hein et al. (2018). The designed method should be neutral to such types of platforms and only requires third-party providers to exist as actor roles in the service ecosystem.

7.2 Practical implications

The designed method allows practitioners to systematically reason about and make decisions based on the trade-offs of different degrees of openness towards third-party providers on the digital platform under development. Furthermore, it allows practitioners to use the method without having either a digital platform concept or a deployed and operational digital platform. Therefore, the method provides practitioners with an explicit moment for reflection on openness during the early design phase of a digital platform.

The aim of defining the appropriate degree of openness is for practitioners to design a digital platform that has a better fit with the current and future intentions of the platform owner (i.e. focal organization) and other actors of service ecosystem. Therefore, for example, the appropriate degree of openness stimulates service innovation on the platform.

Another implication is the bridging of business and technical perspectives on platform design. This results in defining a more appropriate degree of openness. The designed method bridges these perspectives in at least two ways. First, it allows practitioners from both business and engineering backgrounds to define the co-created value of the service ecosystem and decide on which activities should be enabled by the digital platform. Second, when defining the non-functional requirements both business level and system level requirements are included, which shows their relationship. Furthermore, results of the trade-off evaluation can be used for other, for example, monetary cost-benefit analysis performed by business developers and may even lead to platform owners outsourcing certain design elements of the platform. Therefore, the designed method can help with a potential make or buy (i.e. join) decision (Hein et al., 2020).

Another practical implication comes from the use of decomposition in the designed method. The method uses decomposition in various method steps to turn design into a decision-making problem. This reduces the complexity of platform design. Furthermore, it allows platform owners to decide on different degrees of openness for other design elements of a digital platform. For example, design elements related to payment processing may be designed with a lower degree of openness due to security and privacy requirements.

Finally, the designed method has shown that deciding on the appropriate degree of openness is a qualitative task. Therefore, the results are influenced by human judgement at various moments. For example, when deciding on which design alternatives to pick, which non-functional requirements to include, what the priority of these requirements are and how the requirements are eventually evaluated. Since this is not avoidable, the designed method includes multiple steps that allow for a description or argumentation. This supports traceability and should make it easier to reflect on previously made decisions, or to communicate these decisions either within the focal organization or eventually with other actors of the service ecosystem.

7.3 Limitations and future research

The main limitations of this research stem from the choice to make the author of this research perform the demonstration (1), the scope of the demonstration (2) and the selection of participants of the focus group (3). Regarding the first limitation, the demonstration case was built to help solve an instance of the problem with the designed artifact and to use as an example to guide the evaluation focus group. However, although this demonstration case used input from practitioners employed by the demonstration case company, most output was generated by the author of this research. This decision was made due to time constraints for both the author of this research and the demonstration case company. This decision combined with limited experience with the application domain on part of the author is a validity threat to this research. Furthermore, this hindered the evaluation within the focus group due to superficiality of the demonstration.

The second limitation concerns the scope of the demonstration. The designed method was demonstrated for only one software service (i.e. requirement). Therefore, it is not well understood if the method is suitable and applicable or other software services. Moreover, the method was intended to be used iteratively by validating method step outputs with relevant service ecosystem actors. However, this was not executed in the demonstration case due to time constraints and remains to be validated.

The third limitation concerns the evaluation. Participants for the focus group were selected based on their employment by the demonstration case company and on the skills needed to execute the designed method. While it was deemed positive that the participants did not have a complete preconceived idea of their service ecosystem and digital platform resulting in demonstrating and discussing the entirety of the designed method, their limited experience with a platform design process was not. This limitation could be considered a major validity threat to this research and resulted in feedback for improvement that was limited in detail. To increase the robustness of the method a first suggestion for future research is to evaluate the method with experienced digital platform designers. Furthermore, the online setting for the focus group might have limited the depth of the discussion during the evaluation. However, it also enabled the researcher to record individual input of participants.

This research also presented several other opportunities for future research. First, the designed method may be useful to apply to different types of actors in the service ecosystem. The designed method currently focused on openness between third-party developers and the platform owner. In this case, the catalogues developed by Sadi & Yu (2017b) may need to be extended or their relevancy validated for other actors.

Second, future research should look to use the designed method within the wider openness discussion. The designed method currently only considers tangible, behavioral platform design elements in the form of software services (i.e. product level requirements) as input for the openness discussion. However, non-tangible boundary resources related to, for example, governance rules and regulations that determine when or how third-party developers might participate on a platform are not specifically included in the designed method, even though they are important for defining the degree of openness of a digital platform.

The third and fourth opportunities for future research were mentioned in the evaluation of Chapter 6 as improvement directions. These were presented to enhance the overall overview over the platform design during the use of the method. The lack of platform design overview was caused by the designed method relying on decomposition to reduce the complexity of the trade-off decision making process. The first improvement direction aimed to help establish the coherence between platform design elements by adding a sub step to method step 4 that would force the users of the method to reach consensus on which activities are to be performed by the platform or other information systems. The second improvement direction aimed to establish coherence between chosen implementation design alternatives for each platform design element (i.e. software service, product level requirement). Suggestions were to create an additional step after step 8, which reflects on the compatibility between chosen design alternatives based on their description and a comparison of the evaluation labels on shared non-functional openness requirements.

Finally, a suggestion in the focus group was made to use the designed method to assess a deployed, operational digital platform. The designed method may then assist business developers and (software) requirement engineers to validate the decisions they have previously made with regard to their value proposition and/or degree of openness.

Bibliography

Adali, O. E., Ozkan, B., Gilsing, R., & Grefen, P. (2021). A Method to Transform Value Propositions of a Service System into Business Services. In *European Conference on Information Systems (ECIS 2021): Human Values Crisis in a Digitizing World* (Vol. 77, pp. 1–16). Retrieved from https://aisel.aisnet.org/ecis2021_rp/77

Adali, O. E., Ozkan, B., Turetken, O., & Grefen, P. (2021). Identification of Service Platform Requirements from Value Propositions: A Service Systems. In *IFIP International Federation for Information Processing 2021* (pp. 311–322). Saint-Etienne: Springer International Publishing. https://doi.org/10.1007/978-3-030-85969-5

Amit, R., & Zott, C. (2015). CRAFTING BUSINESS ARCHITECTURE: THE ANTECEDENTS OF BUSINESS MODEL DESIGN. *Strategic Entrepreneurship Journal*, *9*, 331–350. https://doi.org/10.1002/sej.1200

Baldwin, C. Y., & Woodard, C. J. (2009). The Architecture of Platforms: A Unified View. *Platforms, Markets and Innovation*, 19–44.

Basole, R. C. (2009). Structural Analysis and Visualization of Ecosystems : A Study of Mobile Device Platforms. In *Fifteenth Americas Conference on Information Systems*. San Francisco.

Benlian, A., Hilkert, D., & Hess, T. (2015). How open is this platform? The meaning and measurement of platform openness from the complementors' perspective. *Journal of Information Technology*, *30*, 209–228. https://doi.org/10.1057/jit.2015.6

Böhmann, T., Leimeister, J. M., & Möslein, K. (2014). Service Systems Engineering A Field for Future Information Systems Research Service. *Business & Information Systems Engineering*, *2*, 73–79. https://doi.org/10.1007/s12599-014-0314-8

Bonina, C., & Eaton, B. (2020). Cultivating open government data platform ecosystems through governance: Lessons from Buenos Aires, Mexico City and Montevideo. *Government Information Quarterly*, *37*(May), 1–15.

Bonina, C., Koskinen, K., Eaton, B., & Gawer, A. (2021). Digital platforms for development: Foundations and research agenda. *Information Systems Journal*, (December 2020), 1–34. https://doi.org/10.1111/isj.12326

Boucharas, V., Jansen, S., & Brinkkemper, S. (2009). Formalizing Software Ecosystem Modeling. In *IWOCE'09* (pp. 41–50). Amsterdam.

Boudreau, K. (2010). Open Platform Strategies and Innovation: Granting Access vs. Devolving Control. *Management Science*, *56*(10), 1849–1872.

Bresnahan, B. T., & Greenstein, S. (2014). Mobile Computing: The Next Platform Rivalry. *The American Economic Review*, *104*(5), 475–480.

Broekhuizen, T. L. J., Emrich, O., Gijsenberg, M. J., Broekhuis, M., Donkers, B., & Sloot, L. M. (2021). Digital platform openness: Drivers, dimensions and outcomes. *Journal of Business Research*, *122*(July 2018), 902–914. https://doi.org/10.1016/j.jbusres.2019.07.001

Brozovic, D., & Tregua, M. (2022). The evolution of service systems to service ecosystems : A literature review. *International Journal of Management Reviews*, (December 2021), 1–21. https://doi.org/10.1111/ijmr.12287

Campbell, P. R. J., & Ahmed, F. (2010). A Three-Dimensional View of Software Ecosystems. In *ECSA 2010* (pp. 81–84). Copenhagen.

Chandler, J. D., & Lusch, R. F. (2015). Service Systems: A Broadened Framework and Research Agenda on Value Propositions, Engagement, and Service Experience. *Journal of Service*

Research, *18*(1), 6–22. https://doi.org/10.1177/1094670514537709

Chesbrough, H., & Rosenbloom, R. S. (2007). *The Role of the Business Model in Capturing Value from Innovation: Evidence from Xerox Corporation's Technology Spinoff Companies*.

Christensen, H. B., Hansen, K. M., Kyng, M., & Manikas, K. (2014). Analysis and design of software ecosystem architectures - Towards the 4S telemedicine ecosystem. *Information and Software Technology*, *56*(11), 1476–1492. https://doi.org/10.1016/j.infsof.2014.05.002

Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2000). *Non-Functional Requirements in Software Engineering*. (R. Basili, Victor, Ed.) (1st ed.). Springer Science+Business Media New York.

Davis, Fred, D. (1985). *A Technology Acceptance Model for Empirically Testing New End-User Information Systems: Theory and Results*. MIT.

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User Acceptance of Computer Technology : A Comparison of Two Theoretical Models. *Management Science*, *35*(8), 982–1003.

De Reuver, M., Sørensen, C., & Basole, R. C. (2018). The digital platform: A research agenda. *Journal of Information Technology*, *33*(2), 124–135. https://doi.org/10.1057/s41265-016-0033-3

Eaton, B., Elaluf-Calderwood, S., Sorensen, C., & Yoo, Y. (2015). DISTRIBUTED TUNING OF BOUNDARY RESOURCES: THE CASE OF APPLE'S IOS SERVICE SYSTEM. *MIS Quarterly*, *39*(1), 217–244.

Eisenmann, T., Parker, G., & Van Alstyne, M. (2008). *Opening Platforms: How, When and Why?*

Fehrer, J. A., Woratschek, H., & Brodie, R. J. (2018). A systemic logic for platform business models. *Journal of Service Management*, *29*(4), 546–568. https://doi.org/10.1108/JOSM-02-2017-0036

Franco-bedoya, O., Ameller, D., Costal, D., & Franch, X. (2017). Open source software ecosystems : A Systematic mapping. *Information and Software Technology*, *91*, 160–185. https://doi.org/10.1016/j.infsof.2017.07.007

Fürstenau, D., Auschra, C., Klein, S., & Gersch, M. (2018). A process perspective on platform design and management: evidence from a digital platform in health care. *Electronic Markets*, *29*(4), 581–596. https://doi.org/10.1007/s12525-018-0323-4

Gawer, A. (2014). Bridging differing perspectives on technological platforms: Toward an integrative framework. In *74th Annual Meeting of the Academy of Management, AOM 2014* (Vol. 43, pp. 423–428). https://doi.org/10.5465/AMBPP.2014.278

Gawer, A., & Cusumano, M. A. (2014). Industry platforms and ecosystem innovation. *Journal of Product Innovation Management*, *31*(3), 417–433. https://doi.org/10.1111/jpim.12105

Ghazawneh, A., & Henfridsson, O. (2013). Balancing platform control and external contribution in third-party development: The boundary resources model. *Information Systems Journal*, *23*(2), 173–192. https://doi.org/10.1111/j.1365-2575.2012.00406.x

Ghazawneh, A., & Henfridsson, O. (2015). marketplaces. *Journal of Information Technology*, *30*, 198–208. https://doi.org/10.1057/jit.2015.16

Gibson, M., & Arnott, D. (2007). The use of focus groups in design science research. In *ACIS 2007 Proceedings - 18th Australasian Conference on Information Systems* (pp. 327–337). https://doi.org/10.1007/978-1-4419-5653-8_10

Gordijn, J., & Akkermans, H. (2001). Designing and Evaluating E-Business Models. *IEEE Intelligent Systems*, (August), 11–17.

Gorschek, T., & Wohlin, C. (2006). Requirements Abstraction Model. *Requirements Engineering*, *11*, 79–101. https://doi.org/10.1007/s00766-005-0020-7

Hein, A., Scheiber, M., Böhm, M., Weking, J., Rocznik, D., & Krcmar, H. (2018). TOWARD A DESIGN FRAMEWORK FOR SERVICE PLATFORM ECOSYSTEMS. In *Twenty-Sixth European Conference on Information Systems (ECIS2018)*. Portsmouth, UK.

Hein, A., Schreieck, M., Riasanow, T., Setzke, D. S., Wiesche, M., Böhm, M., & Krcmar, H. (2020). Digital platform ecosystems. *Electronic Markets*, *30*(1), 87–98. https://doi.org/10.1007/s12525-019-00377-4

Henfridsson, O., & Bygstad, B. (2013). The Generative Mechanisms of Digital Infrastructure Evolution. *MIS Quarterly*, *37*(3), 907–931.

Herman, H., Grobbelaar, S. S., & Pistorius, C. (2020). The design and development of technology platforms in a developing country healthcare context from an ecosystem perspective. *BMC Medical Informatics and Decision Making*, *20*(1), 1–24. https://doi.org/10.1186/s12911-020-1028-0

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Informations Research. *Management Information Systems Quarterly*, *28*(1), 75–105.

Homscheid, D., Kilian, T., & Schaarschmidt, M. (2015). Offen versus geschlossen – Welchen Zusammenhang gibt es zwischen Apple iOS- und Android-App-Entwicklern? In *12th International Conference on Wirtschaftsinformatik*. Osnabrück.

Horkoff, J., & Yu, E. (2009). Evaluating goal achievement in enterprise modeling - An interactive procedure and experiences. In *PoEM 2009* (Vol. 39 LNBIP, pp. 145–160). https://doi.org/10.1007/978-3-642-05352-8_12

Immonen, A., Ovaska, E., Kalaoja, J., & Pakkala, D. (2016). A service requirements engineering method for a digital service ecosystem. *Service Oriented Computing and Applications*, *10*(2), 151–172. https://doi.org/10.1007/s11761-015-0175-0

Isckia, T., De Reuver, M., & Lescop, D. (2018). Digital innovation in platform-based ecosystems: An evolutionary framework. In *MEDES 2018 - 10th International Conference on Management of Digital EcoSystems* (pp. 149–156). https://doi.org/10.1145/3281375.3281377

Jansen, S., Handoyo, E., & Alves, C. (2015). Scientists' needs in modelling software ecosystems. In *ACM International Conference Proceeding Series* (Vol. 07-11-Sept). https://doi.org/10.1145/2797433.2797479

Ketonen-Oksi, S., Jussila, J. J., & Kärkkäinen, H. (2016). Social "media-based" value creation and business models. *Industrial Management and Data Systems*, *116*(8), 1820–1838. https://doi.org/10.1108/IMDS-05-2015-0199

Kohlborn, T., Korthaus, A., Chan, T., & Rosemann., M. (2009). Identification and analysis of business and software services-a consolidated approach. *IEEE Transactions on Services Computing*, *2*(1), 50–64. https://doi.org/10.1109/TSC.2009.6

Kruchten, B. (1995). The 4 + 1 View Model of Architecture. *IEEE Software*, 42–50.

Lessard, L., Amyot, D., Aswad, O., & Mouttham, A. (2020). Expanding the nature and scope of requirements for service systems through Service-Dominant Logic: the case of a telemonitoring service. *Requirements Engineering*, *25*(3), 273–293. https://doi.org/10.1007/s00766-019-00322-z

Lusch, R. F., & Nambisan, S. (2015). SERVICE INNOVATION: A SERVICE-DOMINANT LOGIC

PERSPECTIVE. *MIS Quarterly*, *39*(1), 155–176.

Maglio, P. P., & Spohrer, J. (2013). A service science perspective on business model innovation. *Industrial Marketing Management*, *42*(5), 665–670. https://doi.org/10.1016/j.indmarman.2013.05.007

Manikas, K. (2016). Revisiting software ecosystems R esearch: A longitudinal literature study. *The Journal of Systems and Software*, *117*, 84–103. https://doi.org/10.1016/j.jss.2016.02.003

Moody, D. L. (2003). The Method Evaluation Model : A Theoretical Model for Validating Information Systems Design Methods. In *ECIS 2003*.

Ondrus, J., Gannamaneni, A., & Lyytinen, K. (2015). The impact of openness on the market potential of multi-sided platforms: a case study of mobile payment platforms. *Journal of Information Technology*, *30*, 260–275. https://doi.org/10.1057/jit.2015.7

Osterwalder, A. (2004). *The Business Model Ontology : a proposition in a design science approach*. University of Lausanne.

Otto, B., & Jarke, M. (2019). Designing a multi-sided data platform: findings from the International Data Spaces case. *Electronic Markets*, *29*(4), 561–580. https://doi.org/10.1007/s12525-019-00362-x

Parker, G., & Alstyne, M. Van. (2018). Innovation , Openness , and Platform Control. *Management Science*, *64*(7), 3015–3021.

Parker, G., Alstyne, M. Van, & Jiang, X. (2017). PLATFORM ECOSYSTEMS: HOW DEVELOPERS INVERT THE FIRM. *MIS Quarterly*, *41*(1), 255–266.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems2*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Pettersson, O., & Andersson, J. (2016). A Survey of Modeling Approaches for Software Ecosystems. In *ICSOB 2016* (pp. 79–93). https://doi.org/10.1007/978-3-319-40515-5

Poniatowski, M., Lüttenberg, H., Beverungen, D., & Kundisch, D. (2021). Three layers of abstraction: a conceptual framework for theorizing digital multi-sided platforms. *Information Systems and E-Business Management*, (0123456789). https://doi.org/10.1007/s10257-021-00513-8

Pries-Heje, J., Baskerville, R., & Venable, J. (2008). Strategies for design science research evaluation. In *16th European Conference on Information Systems, ECIS 2008*.

Ralyté, J., Deneckère, R., & Rolland, C. (2003). Towards a Generic Model for Situational Method Engineering. In *CAiSE 2003* (pp. 95–110).

Saarikko, T. (2016). Platform Provider by Accident: A Case Study of Digital Platform Coring. *Business and Information Systems Engineering*, *58*(3), 177–191. https://doi.org/10.1007/s12599-016-0426-4

Sadi, M. H., & Yu, E. (2017a). Accommodating Openness Requirements in Software Platforms: A Goal-Oriented Approach. In *CAiSE* (Vol. 10253 LNCS, pp. 44–59). Springer International Publishing AG. https://doi.org/10.1007/978-3-319-59536-8

Sadi, M. H., & Yu, E. (2017b). Modeling and analyzing openness trade-offs in software platforms: A goal-oriented approach. In *REFSQ* (pp. 1–17). https://doi.org/10.1007/978-3-319-54045-0_3

Schreieck, M., Krcmar, H., & Wiesche, M. (2016). Design and Governance of Platform Ecosystems – Key Concepts and Issues for future research. In *Twenty-Fourth European Conference on Information Systems (ECIS)*. Istanbul.

Setzke, David, S., Böhm, M., & Krcmar, H. (2019). Platform Openness: A Systematic Literature Review and Avenues for Future Research. In *14th International Conference on Wirtschaftsinformatik* (pp. 891–905). Siegen.

Spagnoletti, P., Resca, A., & Lee, G. (2015). A design theory for digital platforms supporting online communities: a multiple case study. *Journal of Information Technology*, *30*, 364–380. https://doi.org/10.1057/jit.2014.37

Storbacka, K. (2011). A solution business model: Capabilities and management practices for integrated solutions. *Industrial Marketing Management*, *40*(5), 699–711. https://doi.org/10.1016/j.indmarman.2011.05.003

Thomas, L. D. W., Autio, E., & Gann, D. M. (2014). Architectural leverage: Putting platforms in context. *Academy of Management Perspectives*, *28*(2), 198–219. https://doi.org/10.5465/amp.2011.0105

Tiwana, A. (2015). Platform desertion by app developers. *Journal of Management Information Systems*, *32*(4), 40–77. https://doi.org/10.1080/07421222.2015.1138365

Tiwana, A., Konsynski, B., & Bush, A. A. (2010). Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research*, *21*(4), 675–687. https://doi.org/10.1287/isre.1100.0323

Tremblay, M. C., Hevner, A. R., & Berndt, D. J. (2010). Focus Groups for Artifact Refinement and Evaluation in Design Research. *Communications of the Association for Information Systems*, *26*, 599–618. https://doi.org/10.17705/1cais.02627

Tura, N., Kutvonen, A., & Ritala, P. (2018). Platform design framework: conceptualisation and application. *Technology Analysis and Strategic Management*, *30*(8), 881–894. https://doi.org/10.1080/09537325.2017.1390220

Turetken, O., Grefen, P., Gilsing, R., & Adali, O. E. (2019). Service-Dominant Business Model Design for Digital Innovation in Smart Mobility. *Business and Information Systems Engineering*, *61*(1), 9–29. https://doi.org/10.1007/s12599-018-0565-x

Ulrich, K. T. (1995). The Role of Product Architecture in the Manufacturing Firm. *Research Policy*, *24*(3), 419–440.

Vargo, S. L., & Lusch, R. F. (2004). Evolving to a New Dominant Logic for marketing. *Journal of Marketing*, *68*(January), 1–17.

Vargo, S. L., & Lusch, R. F. (2016). Institutions and axioms : an extension and update of service-dominant logic. *Journal of the Academic Marketing Science*, *44*, 5–23. https://doi.org/10.1007/s11747-015-0456-3

Vegendla, A., Duc, A. N., Gao, S., & Sindre, G. (2018). A systematic mapping study on requirements engineering in software ecosystems. *Journal of Information Technology Research*, *11*(1), 49–69. https://doi.org/10.4018/JITR.2018010104

West, J. (2003). How open is open enough? Melding proprietary and open source platform strategies. *Research Policy*, *32*(7), 1259–1285. https://doi.org/10.1016/S0048-7333(03)00052-0

WHO. (2021). *Global strategy on digital health 2020-2025*. Geneva.

Appendices

Appendix A: Method Templates

Table A.1: Service Domain - Business Capability Template

Service Domains	Service Operations	Business Capabilities		
		Service Ecosystem actor 1		Service Ecosystem actor 2
		Capability X	Capability Y	Capability Z
Service Domain 1	*Service Operation 1*	*X*		
	Service Operation 2		*X*	

Table A.2: Business Capability Template

The Name of the Business Capability:		
Description:		
Capability Owner:		
Goals Addressed:		
Dependencies to Other Capabilities:		
Business Processes	**Tangible Resources**	**Intangible Resources**

Table A.3: Business Service Description Template

The Name of the Business Service:	
Service Description	
Service Owner:	
Business Service Operations / Delivery Mechanism:	
Type of the Business Service:	
Used Resources:	
Strategic Importance	
Business Goals Addressed:	
Customer Goals Addressed:	
Context Relations	
Connected Value Proposition(s):	

Connected Business Capability:	
Connected Business Services:	

Table A.4: Software Service Identification Template

Focal Organization					
Business Service	**Service Operations**	**Executable via Software Service (Yes, Partially, No)**	**(Applies if decomposition required) Decomposed Service Operation**	**(Applies if decomposition required) Executable via Software Service (Yes, Partially, No)**	**Software Service Required to Open Up Platform to Third Party? (Yes, No)**
Example Business Service	*Example Service Operation performed by Business Service*	*Partially*	*Example decomposition*	*Yes (SS-1)*	
			Example decomposition	*No*	
	Example Service Operations performed by Business Service	*No*			
Example Business Service	*Example Service Operation performed by Business Service*	*Yes (SS-2)*			

Table A.5: Software Service Description Template

The Name of the Software Service (SS-#):	
Priority (High/Medium/Low):	
Software Service Description	
Functionality Description:	
Software Service Owner:	
Linked Business Service(s)	
Benefits for using Software Service / Digital Platform	
Involved Ecosystem Actors and their relevant (Software) Entities	

Table A.6: Non-Functional Openness requirements (NFR) Template for an individual Software Service

SS-#: Name of the software service		
Business level Openness requirement	Priority (non-critical, critical, very critical)	Description
Example Business Requirement [Example Entity]		*[Explanation of relevancy]*
System level openness requirement	Priority (non-critical, critical, very critical)	Description
Example System Requirement [Example Entity]		*[Explanation of relevancy]*
General design concern openness requirement	Priority (non-critical, critical, very critical)	Description
Example General design requirement [Example Entity]		*[Explanation of relevancy]*

Table A.7: Design Alternative Template

SS#-Name of Software Service
DA#: Design Alternative Name
Description of design alternative
DA#: Design Alternative Name
Description of design alternative

Table A.8: Evaluation of refined non-functional openness requirements Template

SS#-Name of Software Service			
Design Alternative	*Decomposed/Refined non-functional requirement*	*Decomposed/Refined non-functional requirement*	*Decomposed/Refined non-functional requirement*
DA#- Design Alternative	*Evaluation label*	*Evaluation label*	*Evaluation label*
DA#- Design Alternative	*Evaluation label*	*Evaluation label*	*Evaluation label*
DA#- Design Alternative	*Evaluation label*	*Evaluation label*	*Evaluation label*

Table A.9: Evaluation of highest priority requirements Template

SS#-Name of Software Service				
Requirements	*General Design concern*	*General Design Concern*	*Business-Level Openness Requirement*	*System-Level Openness Requirement*
Priority	*Critical/Very Critical*	*Critical/Very Critical*	*Critical/Very Critical*	*Critical/Very Critical*
DA#-Design Alternative name	*Evaluation Label*	*Evaluation Label*	*Evaluation Label*	*Evaluation Label*
DA#-Design Alternative name	*Evaluation Label*	*Evaluation Label*	*Evaluation Label*	*Evaluation Label*
DA#-Design Alternative name	*Evaluation Label*	*Evaluation Label*	*Evaluation Label*	*Evaluation Label*

Appendix B: Business capability descriptions

Table B.2: Business Capability Data Hosting

The Name of the Business Capability:	Data Hosting	
Description:	The ability to host healthcare data	
Capability Owner:	Focal Organization	
Goals Addressed:	Provide Evidence Based healthcare	
Dependencies to Other Capabilities:		
Business Processes	**Tangible Resources**	**Intangible Resources**

Table B.3: Business Capability Data Analytics

The Name of the Business Capability:	Data Analytics	
Description:	The ability to analyze healthcare data	
Capability Owner:	Focal Organization	
Goals Addressed:	Provide Evidence Based healthcare	
Dependencies to Other Capabilities:		
Business Processes	**Tangible Resources**	**Intangible Resources**

Table B.4: Business Capability Research Project Management

The Name of the Business Capability:	Research Project Management	
Description:	The ability to manage research projects that lead to new healthcare knowledge	
Capability Owner:	Focal Organization	
Goals Addressed:	Provide Evidence Based healthcare	
Dependencies to Other Capabilities:		
Business Processes	**Tangible Resources**	**Intangible Resources**

Table B.5: Business Capability Product Development

The Name of the Business Capability:	Product Development
Description:	The ability to develop the platform (software) and hardware on which the platform operates

Capability Owner:	Focal Organization	
Goals Addressed:	Provide Evidence Based healthcare	
Dependencies to Other Capabilities:		
Business Processes	**Tangible Resources**	**Intangible Resources**

Table B.6: Business Capability Sales & Delivery

The Name of the Business Capability:	Sales & Delivery (HW)	
Description:	The ability to sell and deliver the hardware on which (part of) the platform operates	
Capability Owner:	Focal Organization	
Goals Addressed:	Provide Evidence Based healthcare	
Dependencies to Other Capabilities:		
Business Processes	**Tangible Resources**	**Intangible Resources**

Appendix C: Business Service descriptions

Table C.1: Data Hosting Business Service

The Name of the Business Service:	BS1: Data Hosting
Service Description	
Service Owner:	Focal Organization (GOAL3)
Business Service Operations / Delivery Mechanism:	Storing Healthcare data Distributing Healthcare data
Type of the Business Service:	STANDARD
Used Resources:	TBD
Strategic Importance	
Business Goals Addressed:	Facilitating Ecosystem
Customer Goals Addressed:	Provide Evidence Based healthcare
Context Relations	
Connected Value Proposition(s):	Facilitated Ecosystem
Connected Business Capability:	Data Hosting
Connected Business Services:	

Table C.2: Research Gap Identification Business Service

The Name of the Business Service:	BS2: Research Gap Identification
Service Description	
Service Owner:	Focal Organization (GOAL3)
Business Service Operations / Delivery Mechanism:	Identifying Research Gaps
Type of the Business Service:	UNIQUE
Used Resources:	TBD
Strategic Importance	
Business Goals Addressed:	Facilitating Ecosystem
Customer Goals Addressed:	Provide Evidence Based healthcare
Context Relations	
Connected Value Proposition(s):	Facilitated Ecosystem
Connected Business Capability:	Data Analytics
Connected Business Services:	

Table C.3: Health Knowledge Initiation Business Service

The Name of the Business Service:	BS3: Health Knowledge Initiation
Service Description	
Service Owner:	Focal Organization (GOAL3)
Business Service Operations / Delivery Mechanism:	Attracting Research Groups
Type of the Business Service:	STANDARD
Used Resources:	TBD
Strategic Importance	
Business Goals Addressed:	Facilitating Ecosystem
Customer Goals Addressed:	Provide Evidence Based healthcare
Context Relations	
Connected Value Proposition(s):	Facilitated Ecosystem
Connected Business Capability:	Research Project Management
Connected Business Services:	

Table C.4: Health Knowledge Institutionalization Business Service

The Name of the Business Service:	BS4: Health Knowledge Institutionalization
Service Description	
Service Owner:	
Business Service Operations / Delivery Mechanism:	Applying for Accreditation Ensuring Health Knowledge Meets Financing Standard
Type of the Business Service:	STANDARD
Used Resources:	TBD
Strategic Importance	
Business Goals Addressed:	Facilitating Ecosystem
Customer Goals Addressed:	Provide Evidence Based healthcare
Context Relations	
Connected Value Proposition(s):	Facilitated Ecosystem
Connected Business Capability:	Research Project Management
Connected Business Services:	

Table C.5: Clinical Decision Support Business Service

The Name of the Business Service:	BS5: Clinical Decision Support
Service Description	
Service Owner:	Focal Organization (GOAL3)
Business Service Operations / Delivery Mechanism:	Providing Configurable Clinical Decision Support Updating Decision Support with New Healthcare Knowledge
Type of the Business Service:	STANDARD
Used Resources:	Digital Platform, New Healthcare Knowledge
Strategic Importance	
Business Goals Addressed:	Facilitating Ecosystem
Customer Goals Addressed:	Provide Evidence Based healthcare
Context Relations	
Connected Value Proposition(s):	Facilitated Ecosystem
Connected Business Capability:	Data Hosting Product Development
Connected Business Services:	

Table C.6: Sales & Delivery (HW) Business Service

The Name of the Business Service:	BS6: Sales & Delivery (HW)
Service Description	
Service Owner:	Focal Organization (GOAL3)
Business Service Operations / Delivery Mechanism:	Providing Configurable Clinical Decision Support
Type of the Business Service:	UNIQUE
Used Resources:	TBD
Strategic Importance	
Business Goals Addressed:	Facilitating Ecosystem
Customer Goals Addressed:	Provide Evidence Based healthcare
Context Relations	
Connected Value Proposition(s):	Facilitated Ecosystem
Connected Business Capability:	Sales & Delivery
Connected Business Services:	

Appendix D: Software Service descriptions

Table D.1: Software Service SS-7

The Name of the Software Service (SS-7):	Updating Clinical Decision Support with New Healthcare Knowledge
Priority (High/Medium/Low):	High
Software Service Description	
Functionality Description:	This software service (i.e. product level requirement) is concerned with enabling the integration of health knowledge developed by research groups (i.e. third-party content) into the digital platform. Therefore this functionality aims to deliver third party content to the healthcare provider (i.e. customer).
Software Service Owner:	Focal Organization (GOAL3)
Linked Business Service(s)	Clinical Decision Support
Benefits for using Software Service / Digital Platform	(+) Reduced 'time to market' for healthcare knowledge (+) Easier access to market for healthcare knowledge
Involved Ecosystem Actors and their relevant (Software) Entities	Focal Organization (GOAL3) [Platform, Healthcare Data] Research Group [Third Party Content]

Appendix E – Non-Functional Openness Requirements Catalogue

Table E.1: Business Level Openness Requirements Catalogue

Market Related objectives	**Description**	**Reference**
Market Reach Market Presence New Markets Standardized Market Adoptability Time to Market	A main reason for opening up software platforms is to expand market reach, open up new markets and communities for a platform, increase the adoption of a platform among various users and developer communities and reduce time to market of new and innovative features,	Popp, K. M. (2010). Goals of Software Vendors for Partner Ecosystems–A Practitioner´ s View. In Software Business (181-186). Springer Berlin Heidelberg. Bosch, J. (2012). Software ecosystems: Taking software development beyond the boundaries of the organization. Journal of Systems and Software, 85(7), 1453-1454. Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., & Robinson, W. (2011). The brave new world of design requirements. Information Systems, 36(7), 992-1008.
Customer-Related Objectives	**Description**	**Reference**
Attracting New Customers Developing New Customer Communities Stickiness of the Platform Customer Retention	Growing the network size of complementary applications hardens switching to a different platform, thus increases the stickiness of a platform. Moreover, growing the variety of platform offerings increases attractiveness of the platform for new and potential users and increases value of the core product to existing users.	Popp, K. M. (2010). Goals of Software Vendors for Partner Ecosystems–A Practitioner´ s View. In Software Business (181-186). Springer Berlin Heidelberg. Bosch, J. (2012). Software ecosystems: Taking software development beyond the boundaries of the organization. Journal of Systems and Software, 85(7), 1453-1454.
Product-Related Objectives	**Description**	**Reference**
Co-Innovation and Open-Innovation Variety of Software Vendor's Offerings (Third Party)	Innovative features play an important role in the success of a platform, specifically in knowledge intensive domains. Via growing the network size of developers, the platform owners can benefit from emerging external innovations	Popp, K. M. (2010). Goals of Software Vendors for Partner Ecosystems–A Practitioner´ s View. In Software Business (181-186). Springer Berlin Heidelberg.
Financial-Related Objectives	**Description**	**Reference**
Revenue Stream Sharing Cost of Innovation Decreasing Total Cost of Ownership	Collaborating with partners in ecosystems shares the cost of innovation and decreases the total cost of ownership for commodity and innovative functionality	Popp, K. M. (2010). Goals of Software Vendors for Partner Ecosystems–A Practitioner´ s View. In Software Business (181-186). Springer Berlin Heidelberg. Bosch, J. (2012). Software ecosystems: Taking software development beyond the boundaries of the organization. Journal of Systems and Software, 85(7), 1453-1454.
Network-effect Related Objectives	**Description**	**Reference**
Customer and Partner Ecosystem Gravity Community Building	Third party developers play an important role in the success of an open platform through their contributions and innovations. A larger pools of developers will provide more innovative output. Thus platform developers aim to attract and engage a large number of developers to contribute and develop applications to their platforms. Factors such as the degree of openness, low entry barriers of both monetary and technical nature, and the network size of a platform	Koch, S., & Kerschbaum, M. (2014). Joining a smartphone ecosystem: Application developers' motivations and decision criteria. Information and Software Technology, 56(11). Popp, K. M. (2010). Goals of Software Vendors for Partner Ecosystems–A Practitioner´ s View. In Software Business (181-186). Springer Berlin Heidelberg. Bosch, J. (2012). Software ecosystems: Taking software development beyond the boundaries of the organization.

	influence the choice of external developers to join a platform.	Journal of Systems and Software, 85(7), 1453-1454.

Table E.2: System Level Openness Requirements Catalogue

Requirement	Description	Reference
Accessibility of Functionalities and services	An open software platform needs to be accessible to third party applications and have access to the features and services of third-party applications. The ease of access to and from a software platform is an important quality requirement for opening up a platform. Accessibility can be categorized in four levels.	[Anvaari, M., & Jansen, S. (2010). Evaluating architectural openness in mobile software platforms. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (85-92).
Accessibility of Data		
Accessibility of Platform structure (i.e. features and components)		
Source code		
Extensibility	An Open platform needs to be extended and complemented by other software applications and components over time. Extensibility quality identifies how easy a new application or feature can be added to a platform. Various quality criteria contribute to the extensibility (below)	
Composability	Open and seamless integration of external modules is an important requirement for a platform. Factors such as decoupling third-party applications from each other, eliminating the need for development synchronization, and independent development, integration, and validation of third-party applications contribute to the composability of an open platform. Carefully decoupled components with well defined interfaces enable third-party developers to modify their applications without disrupting the overall correctness. Platform interfaces should decouple the platform organization from the third-party applications. Achieving this objective, allows the platform owner to release new version of the platform or new components without disabling the externally developed applications operating on top of the platform	Bosch, J., & Bosch-Sijtsema, P. (2010). From integration to composition: On the impact of software product lines, global development and ecosystems. Journal of Systems and Software, 83(1), 67-76. Eklund, U., & Bosch, J. (2014). Architecture for embedded open software ecosystems. Journal of Systems and Software, 92, 128-142.
Deployability	Third-party applications must be possible to be deployed independently of each other, and the platform behavior must not depend on the order in which applications are deployed	Eklund, U., & Bosch, J. (2014). Architecture for embedded open software ecosystems. Journal of Systems and Software, 92, 128-142.
Stability	Open software platforms and their APIs need to be sufficiently stable over time to provide a stable infrastructure for third-party applications	Eklund, U., & Bosch, J. (2014). Architecture for embedded open software ecosystems. Journal of Systems and Software, 92, 128-142.
Configurability	Open software platforms must support variability in configuring the platform and third-party applications to enable customized products be developed	Eklund, U., & Bosch, J. (2014). Architecture for embedded open software ecosystems. Journal of Systems and Software, 92, 128-142.
Evolvability	In open software platforms, new functionality are continuously added and the size of the platforms continuously grows. To deal with the growth, it is required to proactively refactor platform architecture and standardize platform interfaces.	Bosch, J. (2010). Architecture challenges for software ecosystems. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (pp. 93-95).

Decentralizability and Distributability	The functionalities of an open software platform need to be distributed among several applications, and platform components need to operate in a decentralized environment. Thus, the ease to operate in a decentralized environment is an important quality requirement for an open software platform.	Scacchi, W. (2007). Free/open-source software development: Recent research results and methods. Advances in Computers, 69, 243-295.
Interoperability	An open software platform requires to easily cooperate and interact with third-party applications. Mechanisms are required to coordinate and facilitate the interactions between the platform and third-party applications and to resolve conflicts that arise in coordination	Boudreau, K. (2010). Open platform strategies and innovation: Granting access vs. devolving control. Management Science, 56(10), 1849-1872. Scacchi, W. (2007). Free/open-source software development: Recent research results and methods. Advances in Computers, 69, 243-295. Bosch, J. (2010). Architecture challenges for software ecosystems. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (pp. 93-95).
Reusability	An open software platform and its components need to be used and re-used in the development of other software features and applications. The ease to do so is an important design quality in an open platform.	
Modifiability	To use the platform in the development of other applications and software features, the platform or some parts of its functionalities or structures may need to be modified and customized. Thus, the platform should provide mechanisms that enables easy modification of some features.	Cataldo, M., & Herbsleb, J. D. (2010). Architecting in software ecosystems: interface translucence as an enabler for scalable collaboration. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (65-72).
Transparency	To be complemented and extended by third-party applications, the platform structure, functionalities, and behavior need to be visible and transparent to external applications to various degrees	Cataldo, M., & Herbsleb, J. D. (2010). Architecting in software ecosystems: interface translucence as an enabler for scalable collaboration. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (65-72).

Table E.3: General Design Concerns Requirements Catalogue

Requirement	Description	Reference
Operational Security Integrity Confidentiality Privacy	The end-users use a composition of the core of platform and various external applications developed on top of it. Security concerns arise as possible defective or malicious code in external applications may disable the overall system. Mechanisms are required: (1) to guarantee the integrity of platform services and data in the presence of access by third-party applications;(2) to preserve the confidentiality and privacy of the end-users' information and platform data when opening up a platform to third-party developers; and (3) to ensure safe and correct operation of features and services developed by multiple parties.	Knauss, E., Yussuf, A., Blincoe, K., Damian, D., & Knauss, A. (2016). Continuous clarification and emergent requirements flows in open-commercial software ecosystems. Requirements Engineering, 1-21. Eklund, U., & Bosch, J. (2014). Architecture for embedded open software ecosystems. Journal of Systems and Software, 92, 128-142. Bosch, J. (2010). Architecture challenges for software ecosystems. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (pp. 93-95). Scacchi, W., & Alspaugh, T. A. (2013). Processes in securing open architecture software systems. In Proceedings of

		International Conference on Software and System Process. Baresi, L., Di Nitto, E., & Ghezzi, C. (2006). Toward open-world software: Issue and challenges. Computer, 39(10), 36-43.
Controllability Maintainability Centralizability	The development and maintenance of an open platform and its complementary applications is shared among various parties. In this setting, mechanisms are required to manage software enhancements, extensions, and architectural revisions in decentralized projects. Moreover, rules are required to govern and control the applications network.	Boudreau, K. (2010). Open platform strategies and innovation: Granting access vs. devolving control. Management Science, 56(10), 1849-1872. Ghazawneh, A., & Henfridsson, O. (2013). Balancing platform control and external contribution in third-party development: the boundary resources model. Information Systems Journal, 23(2), 173-192. Scacchi, W. (2007). Free/open source software development: Recent research results and methods. Advances in Computers, 69, 243-295.
Reliability Trust Accountability	In open software platforms, parties providing and consuming a software service are easily exposed to cheaters. Therefore, mechanisms are required to guarantee trustworthiness and accountability of third-party services and functionalities.	Baresi, L., Di Nitto, E., & Ghezzi, C. (2006). Toward open-world software: Issue and challenges. Computer, 39(10), 36-43. Scacchi, W. (2007). Free/open source software development: Recent research results and methods. Advances in Computers, 69, 243-295.
Proprietary Ownership	The ownership and intellectual property rights of the applications, components and data produced by external developers is a critical concern in open software platforms. Mechanisms are required to ensure responsibility and commitment to updating and supporting third-party modules. Moreover, the alignment of component licenses need to be checked in the usage and composition of open software components and modules at build time and deployment	Knauss, E., Yussuf, A., Blincoe, K., Damian, D., & Knauss, A. (2016). Continuous clarification and emergent requirements flows in open-commercial software ecosystems. Requirements Engineering, 1-21. Bosch, J. (2010). Architecture challenges for software ecosystems. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (pp. 93-95). Baresi, L., Di Nitto, E., & Ghezzi, C. (2006). Toward open-world software: Issue and challenges. Computer, 39(10), 36-43.

Appendix F – Evaluation Focus Group Participant Response Quotes

- What is your first impression of the method?

Participant 2: "Wow, quite extensive, maybe challenging to use, but for platform design this may be the right way to go through it thoroughly"

Participant 5: "Quite overwhelming, I did not expect this many steps" … "This needs time to digest a bit .. its presented at a high level which makes it challenging to make it tangible"

Participant 4: "I do not have a strong business background, so it only started to make sense when talking about requirements "…. "Everything before this, seemed quite complex to me" "I don't think we had enough time to explain it such that I could completely follow it"

Participant 1: "To me it feels like a pretty logical framework"

Participant 5: "Seems like a tool that could also be used to assess a platform that already exists"

- What are the strong and weak elements of the method?

Participant 5: "Its a good approach to start, I like that you attempt to structure this process, but there are simplifications and assumptions embedded and these should be made explicit"

Participant 5: "Also, in each step we can spend perhaps months"

Participant 4: "When you apply this method it seems that you need have to have very clearly in your mind what the goal is, and I am not sure if the method provides ways to form or refine that"… "That is an area that we struggle with as a company"

Participant 4: "There are a lot of tools, sub steps and terminology used in the method which takes time to digest and understand and get everybody on the same page, that seems like an entry hurdle"… "But I think that if this 'understanding' phase has been completed it would be a very strong method to visualize and communicate with each other"

Participant 1:" I agree, but every iteration we use this method it would take a shorter time"

Participant 1: "What I like about it is that it brings together the business aspects of 'what do we want to do' with development" "It structures the process, makes coming to decisions explicit, and it helps to guide that process"

Participant 1: "A weakness may be that we need all the people in this videocall and perhaps even more to go through the whole method" "It requires a lot of different backgrounds to align on a topic" … "But, perhaps this not be a weakness, because it helps to make the idea tangible and get the ball rolling"

Participant 5: "it gave us a chance to think about it in a way we normally would not have" "But we still have fundamental questions about the platform that we have not had answered"

Participant 3: "Openness touches so many aspects of a platform, but how do we start small? It seems like a lot of work for the first few steps".. "But I see that not everything might require openness, for example, payment processing elements may need to be designed closed"

- What would you like to add to or change from the method?

Participant 5: “I liked the radar step, but a platform might use these actors in different ways” “For example in Ghana, the government is the healthcare provider, the insurance company and the training partner. Then it’s just GOAL3 and the government” .. “How do you design your platform per situation? Or should it account for all of them? ”

Participant 5: “The openness trade-offs are complex with many aspects to it … if you approach a design element on its own you would decide differently than if you consider the platform as a whole”

Participant 3: “Now we went through one design decision, but how to deal with multiple design decisions influencing each other? What if there are conflicting requirements”? “How to weigh these to each other?”

Participant 3: “Also, the priority setting of critical and very critical might not be precise enough”

Participant 5: “There are too many steps, further simplification needs to happen if that’s possible.” .. “In practice you would probably never use such a complicated tool but something more simple” … “To me design comes down more to gut feeling and the vision you have”

Participant 5: “However, I am not entirely sure if steps are missing, I do not have too much experience with platform design” .. “Evaluating with organizations who have more experience with the platform design process might make the method more robust”

Participant 1: “To truly understand the method we need to go through these steps ourselves, I think…to test how it works, and if it works”