Emma Saari

Improving the user experience of open application programming interface (API) from a digital marketing perspective

A case study in a global telecommunications company

School of Technology and Innovations
Master’s thesis in Computer Science
Master's Program in Information Systems

Vaasa 2023

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Emma Saari
Title of the thesis: Improving the user experience of open application programming interface (API) from a digital marketing perspective: A case study in a global telecommunications company
Degree: Master of Science in Economics and Business Administration
Subject: Information systems
Supervisor of work: Duong Dang
Completion year: 2023 **Pages:** 154

ABSTRACT:

Application programming interface (API) is a programming interface that allows different applications to share information, functionality, and other resources with each other. When creating an open-source application programming interface, customer feedback is important. Understanding end-user needs can help improve the interface and target marketing activities. It is known from previous studies that quality of experience (QoE) is the driver for open radio access networks (O-RAN) and that user experience (UX) is what affects system use and leads to actual usage.

This subject is relevant in this area of science since the software developer's perspective on open application programming interfaces is often disregarded, resulting in fewer studies. The objective of this thesis is to determine the necessary technological requirements to improve software developers' user experience and attract new customers. This research is based on an empirical study developing an open application programming interface for business-to-business (B2B) customers. The technology that is a central part of the study is programmable wireless network that allows to develop non-real time applications called xApps.

This is case study research, which uses both quantitative and qualitative research methods. Information gathered for this research is going to be collected via interviews and surveys from the target group. This thesis's limitation is that the analyzed target group has a limited business market. The empirical data for this research is gathered from a global telecommunication company, from publications about the industry, and surveys and interviews gathered from the target market group.

The three main findings of the thesis are related to how building trust among developers and partners is crucial, how developers need to be encouraged to learn something new and how developers that have more experience have fewer expectations. Previous research indicates that discovery about developers that have more experience have fewer expectations is new in this field of study. As a conclusion there are entry level blockages, telecommunication technology development, partner engagement, community, cost, and platform related killers that affect the motivation and prevent investing into an open API platform that specializes in xApps. To create a successful xApp platform the provider company needs to tackle these problems and highlight the possibilities that xApps offer.

KEYWORDS: API, qualitative research, quantitative research, user experience, software development, digital marketing

VAASAN YLIOPISTO
Tekniikan ja innovaatiojohtamisen akateeminen yksikkö
Tekijä: Emma Saari
Tutkielman nimi: Avoimen sovellusohjelmointirajapinnan käyttökokemuksen parantaminen digitaalisen markkinoinnin näkökulmasta: Tapaustutkimus globaalissa telekommunikaatio yrityksessä
Tutkinto: Kauppatieteiden maisteri
Oppiaine: Tietojärjestelmätiede
Työn ohjaaja: Duong Dang
Valmistumisvuosi: 2023 **Sivumäärä:** 154

TIIVISTELMÄ:

Sovellusohjelmointirajapinnan (API, Application programming interface) avulla eri sovellukset voivat jakaa tietoja, toimintoja ja muita resursseja keskenään. Avoimen lähdekoodin sovellusohjelmointirajapintaa luotaessa asiakaspalaute on tärkeää. Loppukäyttäjien tarpeiden ymmärtäminen voi auttaa parantamaan rajapintaa ja kohdentamaan markkinointia paremmin. Opinnäytetyön tavoitteena on selvittää, mitkä teknologiset vaatimukset ovat tarpeen ohjelmistokehittäjien käyttökokemuksen parantamiseksi ja uusien asiakkaiden houkuttelemiseksi. Aiemmista tutkimuksista tiedetään, että kokemuksen laatu (QoE) on avointen radioliityntäverkkojen (O-RAN) liikkeelle paneva voima ja, että käyttökokemus (UX) vaikuttaa järjestelmän käyttöön ja johtaa varsinaiseen käyttöön.

Aiheella on merkitystä tällä tieteen alalla, koska ohjelmistokehittäjän näkökulma avoimiin sovellusohjelmointirajapintoihin jää usein huomiotta, jolloin ohjelmistokehittäjän näkökulmasta tietoa löytyy vähemmän. Tutkimus perustuu empiiriseen tutkimukseen, jossa kehitetään avoimen sovelluksen ohjelmarajapintaa yritysten välisille asiakkaille (B2B). Tutkimuksen keskeisenä teknologiana on ohjelmoitava langaton verkko, jonka avulla voidaan kehittää ei-reaaliaikaisia xApp-sovelluksia.

Tämä tutkimus on tapaustutkimus, jossa on käytetty sekä kvantitatiivisia että kvalitatiivisia tutkimusmenetelmiä. Tutkimusta varten tietoa on kerätty kohderyhmän haastatteluilla ja kyselyillä. Tutkimuksen rajoituksena on, että analysoidulla kohderyhmällä on rajalliset liiketoimintamarkkinat. Empiirinen data tähän tutkimukseen on kerätty globaalista tietoliikenneyhtiöstä, alan julkaisuista sekä kohderyhmältä kerätyistä kyselyistä ja haastatteluista.

Opinnäytetyön kolme päähavaintoa ovat, että luottamuksen rakentaminen kehittäjien ja kumppaneiden keskuudessa on ratkaisevan tärkeää, kehittäjiä tulee kannustaa oppimaan uutta ja kokeneemmilla kehittäjillä on vähemmän odotuksia. Aiemmat tutkimukset osoittavat, että löytö siitä, että kokeneemmilla kehittäjillä on vähemmän odotuksia, on uusi tällä tutkimusalalla. Johtopäätöksenä lähtötason pullonkaulat, viestintäteknologian kehitys, kumppanuussitoutuminen, yhteisön, kustannusten ja alustan tappajat vaikuttavat motivaatioon ja heikentävät investointia xAppeihin erikoistuneeseen avoimeen API-alustaan. Menestyvän xApp-alustan luomiseksi yrityksen on puututtava näihin ongelmiin ja tuotava esiin xAppien tarjoamat mahdollisuudet.

AVAINSANAT: API, laadullinen tutkimus, määrällinen tutkimus, käyttäjäkokemus, ohjelmistokehitys, digitaalinen markkinointi

Contents

Figures

Tables

1 Introduction

The application programming interface (API) is an interface that exchanges information between separate systems (Chaffey & Chadwick, 2022, p. 495). Companies should not only think about API technical requirements but also think about the people whom they are going to cooperate with in the implementation process. It has been researched that the number of APIs correlates with the Global Startup Index (Huhtamäki et al., 2017, p. 5309). Cultures and markets where API development functions well are where marketers and engineers work together (Huhtamäki et al., 2017, p. 5312). That is why the digital marketing perspective becomes relevant when implementing APIs for software developers.

Each of the companies' departments has a role in developing APIs. Some might even describe the term API as "all people are included" (APIdays, 2019). The value chain defines the value-adding functions companies' departments do to produce their products (Parment et al., 2016). Creative situations are happening when marketers bring IT developers closer to the marketplace to create customer value (Parment et al., 2016). It is also important to get opinions from potential customers to identify design flaws (Nielsen, 2012).

Companies sometimes overlook the possibilities to improve their offerings while only concentrating on their products (APIdays, 2019). Industries might realize that collaborating is critical to surviving (Kotler Hermawan; Setiawan Iwan et al., 2016, p. 20; Svensson et al., 2019, p. 21). One part of committing to APIs is that companies need ecosystems and partnerships where they promote and launch initiatives together (Hannes et al., 2018, p. 133). The key to success with APIs is to collaborate with other companies because partnerships are important and can give a higher degree of success (APIdays, 2019; Parment et al., 2016, p. 54).

This thesis has been done with a case study approach in a big global network operating company that specializes in telecommunication. The initial idea for this thesis came from a real-life problem trying to solve the expansion to a new territory that provides open APIs. The challenge of the thesis is to find solutions to identify what different user groups are expecting and how can these expectations be met. Another challenge is to find ways to market the solution to potential customers.

APIs are costly and modifying them once they have been established is complex and time-consuming, according to Biehl (2015, p. 22). Customers may opt to depart following modifications to the API they have been using and transfer API providers, making API modification even more difficult. To avoid this, the company must ensure that the API is properly done the first time it is released. Investing in developing APIs the way consumers want them minimizes the likelihood of this scenario occurring. Ensuring that consumers are understood and realize what influences customer choices contributes to the foundation of how APIs may be digitally advertised.

The goal of this thesis research is to acquire information on a group of technology firms that might profit from the open ecosystem of the case company. Case company will operate as a platform linking business partners with firms that will construct applications for business partners using case company's APIs. The open APIs that the case firm will provide are intended to improve the radio access network (RAN). The platform will offer a variety of options for developing applications in a safe sandbox environment. The applications that are produced are known as xApps.

User-centered design (UCD) emphasizes the user group’s opinion about the design (Persson et al., 2015, p. 509). This thesis aims to obtain the target group's perspective on the overall business concept and current documentation while keeping UCD in mind. The target audience includes experts from a variety of information technology domains, including cloud, verification, radio technology, IT administration, continuous delivery,

system testing, embedded, networking, machine learning, database, telecommunication, performance testing, and functional testing.

1.1 The purpose of the study

This thesis will focus on enhancing the user experience of an open application programming interface from the standpoint of digital marketing. The thesis is going to focus on improving the user experience (UX) and getting knowledge about the potential user group. These user groups are defined through the company's identified potential market target groups.

This thesis is going to give user experience knowledge about programmable network solutions which have been researched in the field from different angles. Open application platforms have been studied from the standpoint of usability and user experience in the context of transportation, biomedicine, banking, healthcare, entrepreneurship, mobile network, cross-platform, monetization and conceptual modeling (Alay et al., 2017; Bork et al., 2019; Kilpeläinen et al., 2021; Miranda et al., 2016; Nambisan et al., 2018; Nikravesh et al., 2015; Osagie et al., 2017; Schindelin et al., 2015; Wäljas et al., 2010; Zachariadis & Ozcan, 2017). Programmable networks have been studied for their special characteristics, the history behind the technology, recent research initiatives, security reasons, programmable network node, processor architecture, cloud, and software-defined networking possibilities (Anerousis et al., 2021; Campbell et al., 1999; Crowley et al., 2000; Feamster et al., 2013, 2014; Macedo et al., 2015; Nunes et al., 2014; T. Wang et al., 2020; Wen et al., 2022; Yemini & Silva, 1996). Radio network-specific open programmability has been researched regarding xApps over the recent years regarding the benefits, quality of experience (QoE), machine learning (ML) opportunities, trends, market opportunities, future challenges, platform, communication metrics, traffic steering framework, and guidelines (Aryal et al., 2023; Azariah et al., 2022; Bonati et al., 2022; Dryjański et al., 2021; Iturria-Rivera et al., 2022;

Kak et al., 2022; Koufos et al., 2021; Kougioumtzidis et al., 2022; Lacava et al., 2022; Polese et al., 2022; Pönnelin et al., 2023; Singh et al., 2020; Wypiór et al., 2022).

Previous research has shown that it is important to get opinions from potential customers to identify design flaws (Laugwitz et al., 2008; Mäkipää et al., 2022; Nielsen, 2012). However, the focus of this thesis will be on a specific target group and improving the existing solutions to fit their needs. Hence, this thesis is going to provide valuable knowledge in the technology field of ways to improve an open application programmable network platform for software designers, developers, and testers based on their views.

Open application platform has been researched from various aspects. There is less research concerning the digital marketing of open application platforms to customers. Open application programming interfaces are mentioned in platform competition strategies, managing, advertising, marketing, and semantic marketing context as well as being accused of using unfair methods to compete against rivals (Bodle, 2011; Chesbrough, 2004; Foux, 2010; Karhu & Ritala, 2021; Lies, 2019; Sharma, 2019; Simon, 2021; J. Wang, 2009). The motivational factors to encourage users to collaborate in open innovation platforms have been studied too (M. Antikainen et al., 2010; M. J. Antikainen & Vaataja, 2010).

Prior research has shown that user experience, marketing, and technology all have a significant role in product creation (Norman, 1999). While past research has been concentrated briefly on open API marketing and the possibilities to advertise the platform there has not been much research done on specifically digital marketing capabilities of an open API platform. Therefore, this thesis will deliver meaningful information about open APIs in the business field from a digital marketing viewpoint.

These dilemmas lead to forming the following theoretical and empirical objectives:

1) *What are the possible alternatives to improve the user experience for different user groups in an open API platform?*
2) *How could the open API platform be marketed based on these user expectations?*

1.2 The scope of the study

The constraint of this thesis is that it is a case study that will only focus on one technological business that is extending its radio network to programmers who can design, build, and test apps. The technology that the thesis is examining is focusing on programmable wireless networks that create possibilities to create applications known as xApps. The focus group being interviewed for this research is the second of the thesis's restrictions. The focus group consists of a limited number of technical professionals with experience in various sorts of software design, development, and testing.

1.3 The structure of the study

The thesis begins with an introduction and proceeds from there to present an overview of user acceptance, general information on open API, and the current concepts of digital marketing. The thesis also describes the used method and the data collection process. The thesis compares different user groups' answers in interviews and surveys. After this, the thesis outlines the conclusions and analyzes the research questions considering all the previously gathered data. The key concepts of the study are described in Annex 1. Figure 1 illustrates the structure of the study.

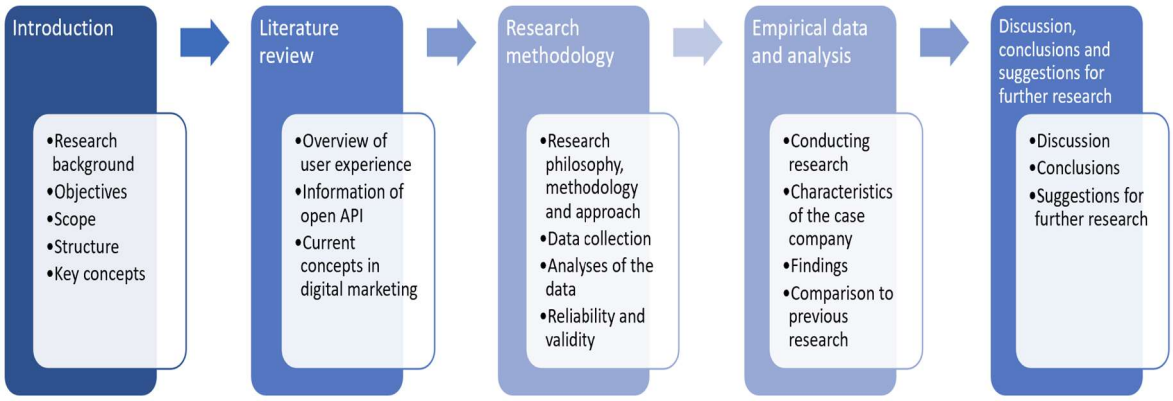


Figure 1. Structure of the thesis

2 Literature review

Technology, marketing, and user experience are the "three legs of human-centered development," according to Norman (1999). The stool stands sturdy when all these components are aligned but collapses when one perspective is overlooked. Figure 2 depicts the same concept as a funnel including these three critical aspects required for the product to function.

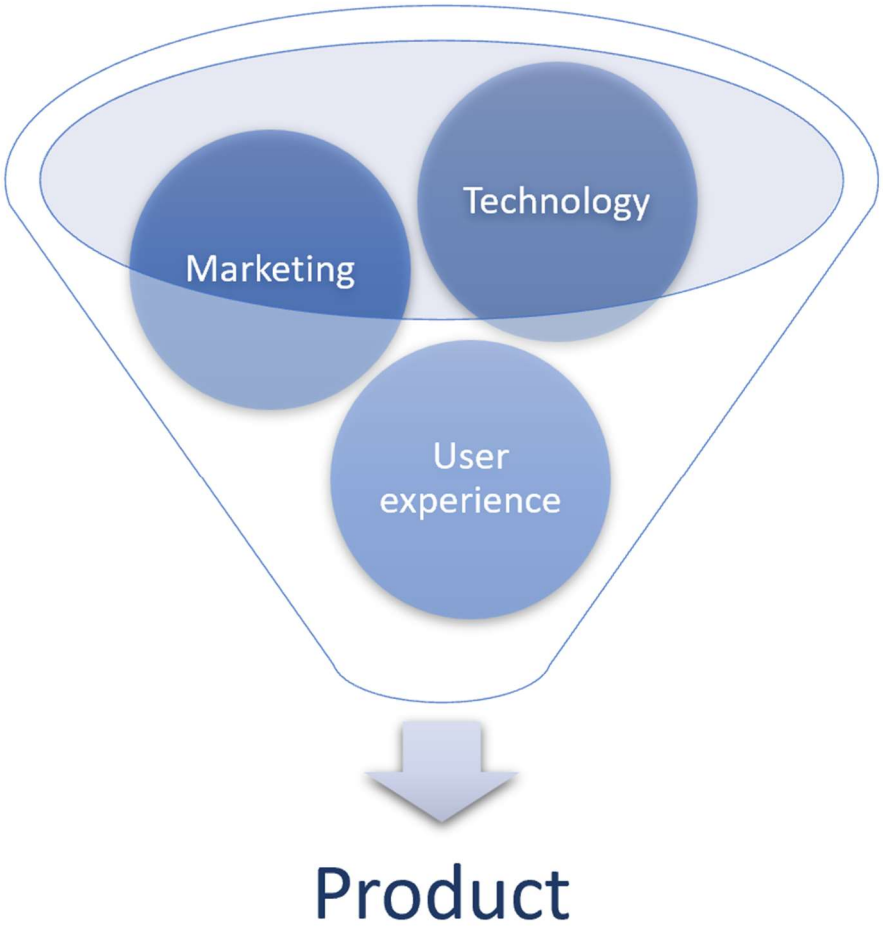


Figure 2. Illustration of needed components for product development (Adapted from Norman, 1999)

A platform is a business model that connects customers and business partners through a platform to create value (Parker et al., 2016, p. 5). The business model that most companies use is a pipeline model (Parker et al., 2016, p. 6). The pipeline model creates

value through steps that connect the customer and the business partner on each end (Parker et al., 2016). Transforming network operators' businesses from a traditional pipeline business to a platform business requires multiple efforts.

In the end platform use depends on users' motivation and what goal the user wants to accomplish (Latham & Pinder, 2005, p. 486). The ideal situation is the feedback loop where the platform provides such interactions that "create a constant stream of self-reinforcing activities" (Parker et al., 2016, p. 296). The platform is succeeding if the user finds the value unit flow relevant and interesting enough to continuously generate more value units and wants to engage in interactions (Parker et al., 2016, p. 296).

Creating value through a platform may lead to changing the company's entire business model (Moilanen et al., 2018). Parker et al. (2016, p. 14) mention "strategy, operations, marketing, production, research and development, and human resources" as business functions that are going to be changing because platforms are becoming more common. Whereas Moilanen et al. (2018, p. 16) want to divide the groups that are responsible for API development into customer experience, marketing, sales and purchases, partnerships and business operations, information security and data protection, cloud services, integration, and business systems. While each of these business functions is important based on Moilanen et al. (2018, p. 16) and Parker et al. (2016, p. 14) way of defining the business groups this thesis is going to be focusing mostly on business functions that relate to research and development by improving customer experience and marketing.

In the following chapters, the thesis is going to proceed to explain each of these key concepts in more detail. This will give an overview of user experience, open API, current concepts of digital marketing, user journey, benchmarking, and service blueprint.

2.1 An overview of user experience

The evolving landscape of telecommunications, including the emergence of O-RAN architecture, demands a deeper understanding of the quality of experience (QoE) from the user's perspective (Kougioumtzidis et al., 2022). As a response to this gap in research, this thesis will provide a comprehensive exploration of user experience within open APIs, with a particular focus on O-RANs potential as an xApp platform. This thesis aims to contribute insights and meaningful information that can shape the design and development of open API platforms, enhancing the user experience in this dynamic and evolving field.

Norman (1986, pp. 41–42) separates the process of user activity into seven stages. These user activity stages are goal setting, intention formation, action sequence specification, action execution, perceiving the system state, interpreting the state, and assessing the system state concerning the goals and intentions. In principle, the first stage is about establishing the goal which leads to a set of activities to accomplish the goal. Afterward, the user reflects on how well these actions were met.

According to Davis and Fred (1986, p. 10), user behavior can be described as a stimulus that is generated by system capabilities. This creates user motivation as a response organism which may lead to actual system use as a response. Figure 3 represents the cause-effect interactions between variables. The system's qualities and capabilities play a vital impact in the establishment of user motivation (p. 11). The system designer's role is to provide a stimulus that encourages users to utilize the system (Saari, 2021, p. 15). This incentive generates a reaction, indicating that the user begins to utilize the system (Saari, 2021, p. 15).

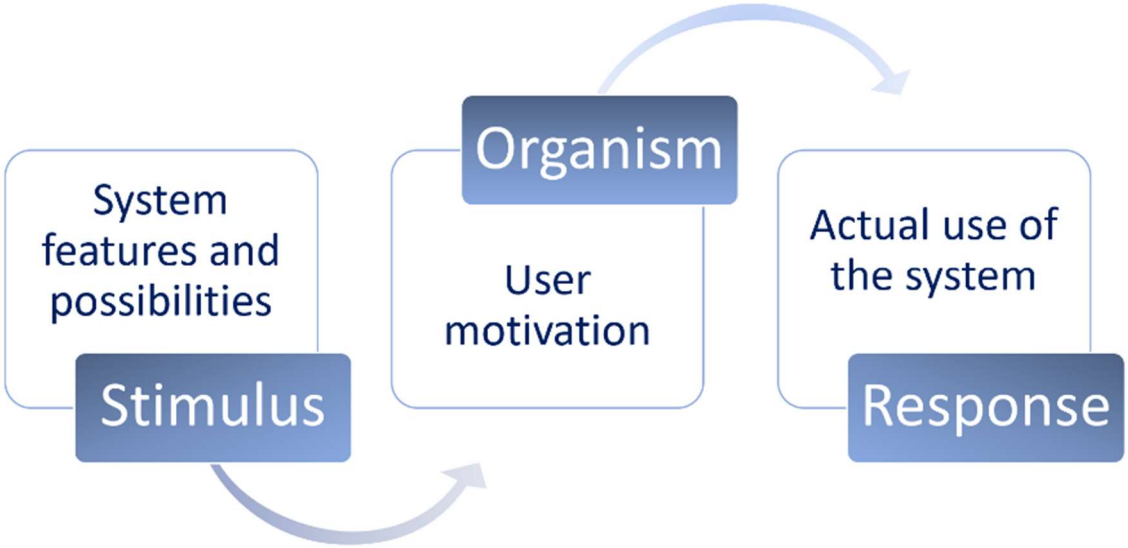


Figure 3. Illustration of user behavior (Adapted from Davis, 1985)

The criteria for user expectations have been researched in the field of technology throughout the years with different types of technology acceptance models (TAM) (Davis, 1985). Technology acceptance models study the usability factors from users' perspectives and try to find solutions that often lead to system usage (Davis, 1985, p. 7). One of these user acceptance models is the unified theory of acceptance and use of technology (UTAUT) model presented in Figure 4 which describes the user behavior by factors and moderators (Venkatesh et al., 2016, p. 329).

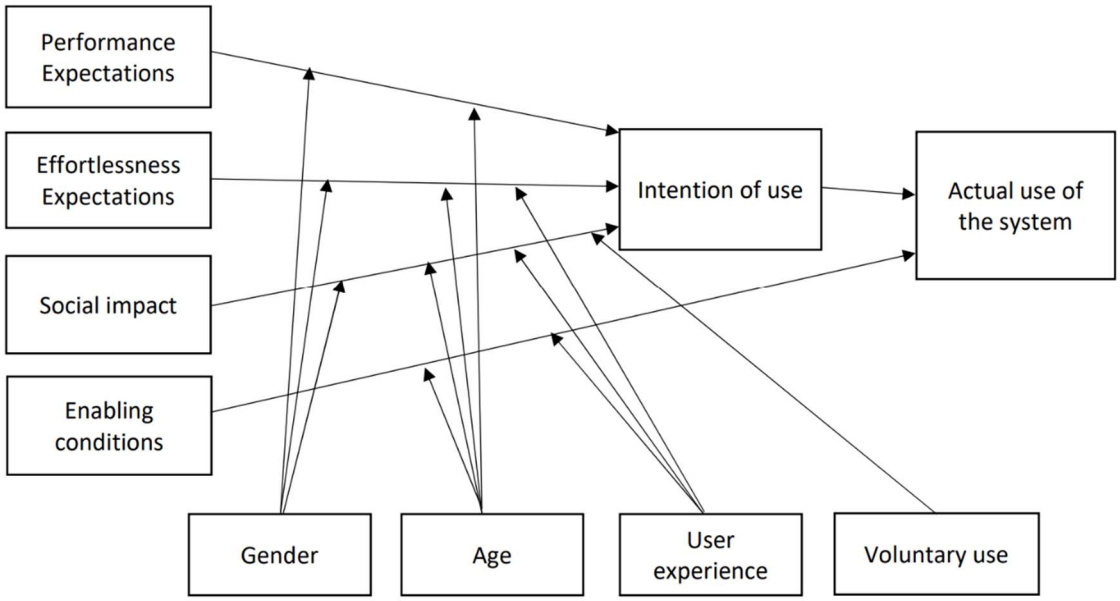


Figure 4. UTAUT model (Adapted from Venkatesh et al., 2016)

Venkatesh et al. (2016, pp. 329–335) discovered that one of the key moderators that affect users' intentions to use the system and leads to actual usage is user experience. User experience influences the user's expectations about the system's accessibility, as well as the societal implications and enabling circumstances. However, in the UTAUT2 model, the previously described links between user experience, effortlessness expectations, and social effect have been deleted (Figure 5). The renewed UTAUT2 model emphasizes user experience as a combination of enabling factors, hedonic motivation, and habit which can lead to intentions of system usage.

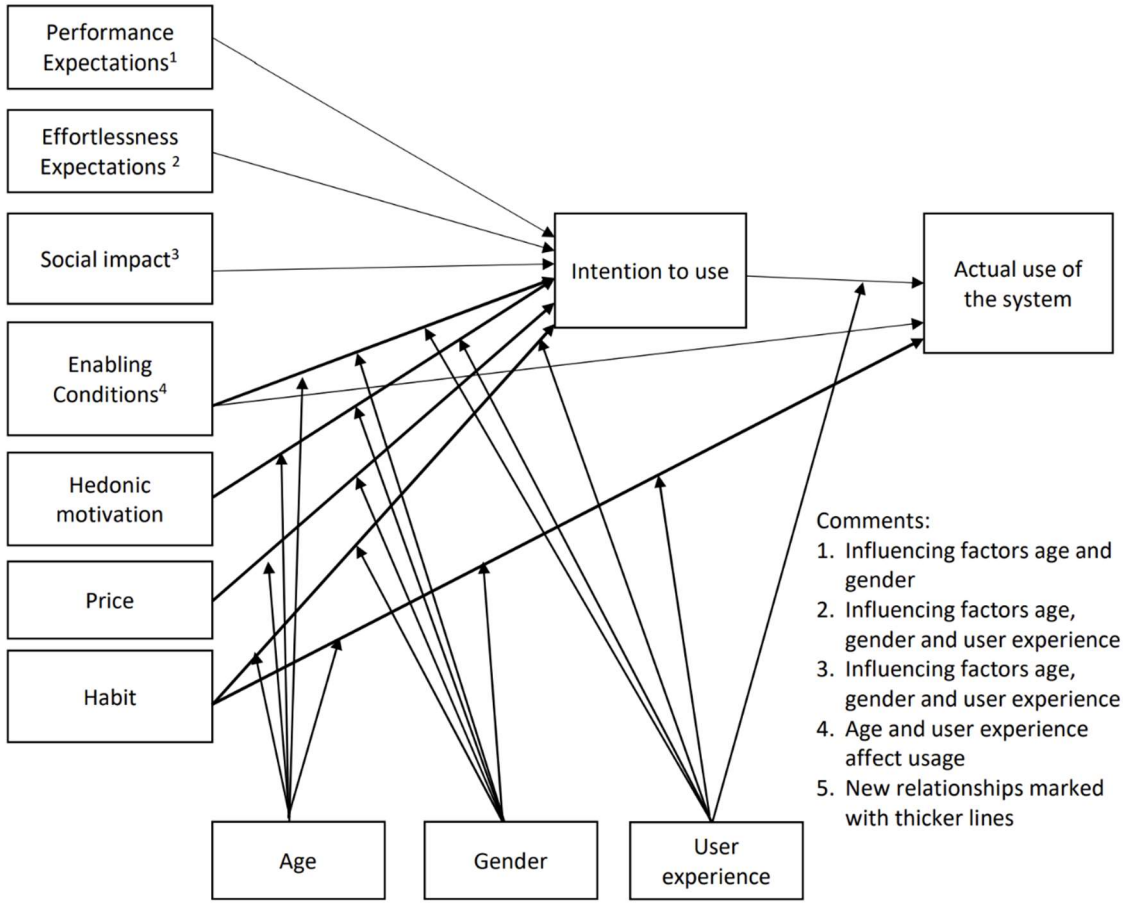


Figure 5. UTAUT2 model (Adapted from Venkatesh et al., 2012)

What makes a design stand out is *accessibility*, *usability*, and *acceptability* (Benyon, 2014, p. 77). Accessibility is the "usability of a product, service, environment or faculty by people with the widest range of capabilities" which leads to being more productive (Benyon, 2014, p. 21; International Organization for Standardization, 2018). These accessibility criteria include physical, psychological, social, cognitive, economic, cultural, or social factors (Benyon, 2014, pp. 77–78). Accessibility is an ethical and legal requirement (Benyon, 2014, p. 77). In the European Union accessibility requirements are controlled by law (Web Accessibility Initiative, 2008, 2018).

Nielsen (1994, p. 24, 2012) emphasizes usability as an acceptability criterion, categorizing it as learnability, efficiency, memorability, errors, and satisfaction. Usability evaluates how enjoyable and straightforward the user experience features are to utilize.

The utility is connected to usability since it considers if the user needs are satisfied by the features that the interface delivers. Both usability and utility result in usefulness.

Acceptability, according to Benyon (2014, p. 84), entails convincing people to utilize the offered technology. Acceptability, unlike usefulness, cannot be examined in a laboratory. Acceptability may be defined in terms of several settings such as politics, convenience, cultural and social habits, usefulness, and economics. The previously described technology acceptance models attempt to understand user motivations and give a method for estimating technological success (Davis, 1985).

A requirement is defined as "something that the product must perform or a quality that the product must possess" (Robertson & Robertson, 2012, p. 9). User experience focuses on the entire experience "to satisfy the specific demands of the consumer" (Nielsen & Norman, n.d.). UX is the interface between humans and technology that necessitates various activities and generates feelings, perceptions, and ideas (Benyon, 2019, p. 212).

Nielsen (1994, p. 24, 2012) recommends allocating at least 10 percent of the overall expenditure to usability. If a user finds the interface tough to use, they will depart immediately in search of a better one. This also applies to what the platform has to offer. If a user's demands are not met by the platform, they will quit and look for another option.

Designing is the process that begins immediately after the user requirements are determined (Robertson & Robertson, 2012, p. 253). It is useful to involve users in the early design process and analyze their present demands (Benyon, 2014, pp. 79–80; Mäkipää et al., 2022, pp. 573–574). Users might supply knowledge about previously unknown challenges, improvements to design, and new possibilities (Benyon, 2014, pp. 79–80; Mäkipää et al., 2022, pp. 573–574). With user testing, observing, and giving the possibility to users tell their opinion the interface can be improved (Nielsen, 2012).

Previous research, such as the study by Osagie et al. (2017, p. 4) has shed light on the critical factors that influence the user experience in digital platforms. Osagie et al. (2017, p. 4) found that many platforms often lack simplicity and user-friendliness, leading users to prefer platforms that resemble popular social platforms such as Facebook. Their findings strongly advocate for the importance of a low learning curve, user-friendly interfaces, clutter-free web pages, easy searchability, access to social communication channels, simple data visualization, and the ease of sharing and engaging with other users through comments.

While these insights are pivotal in understanding user preferences and requirements in various domains such as transportation, biomedicine, banking, healthcare, entrepreneurship, and more, there has been a notable gap in the exploration of user experience within open APIs, particularly in the context of O-RAN architecture and open application platforms (Alay et al., 2017; Bork et al., 2019; Kilpeläinen et al., 2021; Miranda et al., 2016; Nambisan et al., 2018; Nikravesh et al., 2015; Osagie et al., 2017; Schindelin et al., 2015; Wäljas et al., 2010; Zachariadis & Ozcan, 2017). Few studies have delved into the user experience of open APIs in programmable networks, let alone xApps. Notably, the studies related to open API platforms specializing in mobile networks have focused on quality of service and usability aspects (Alay et al., 2017, p. 77).

2.2 General information on open API

While there are studies that have advanced our understanding of radio network-specific open programmability, there remains a noticeable gap in research concerning open API platforms within programmable network platforms. The pivotal role of open APIs in shaping the user experience within this dynamic and evolving field calls for in-depth exploration, which this thesis aims to address.

Application programming interfaces known as APIs “offer a way to integrate several software systems in a simple, clean, clear, and approachable way” (Biehl, 2015, p. 15). APIs can enable consumers and partners to interact and connect throughout the

ecosystem, lowering the costs of company services and systems (Biehl, 2015, p. 16; Moilanen et al., 2018, p. 20). End-users interact with APIs indirectly over applications (Biehl, 2015, p. 18).

APIs allow businesses to diversify their product offerings by embracing the platform economy (Moilanen et al., 2018, p. 20). Figure 6 depicts the key features of open API. Open APIs enable companies to make their data open standard and freely available for usage by anybody (Biehl, 2015, p. 18). The primary consumers of such a solution are developers, who may utilize the offered APIs to construct applications and solutions (Biehl, 2015, p. 16). This also implies that developers are the primary emphasis when building such solutions, making their opinions important (Biehl, 2015, p. 16).

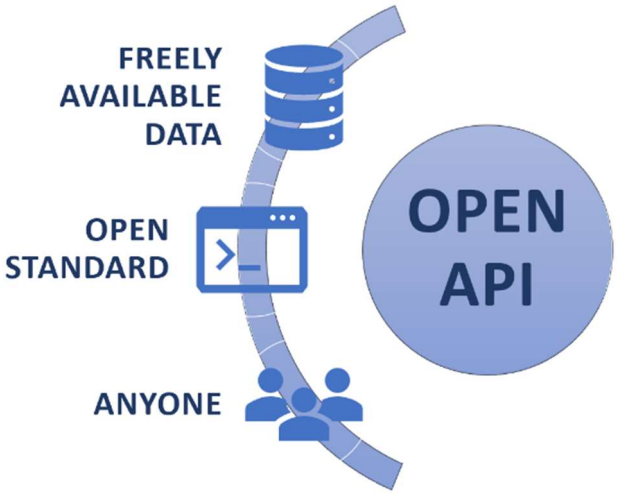


Figure 6. The main characteristics of open API (Adapted from Moilanen et al., 2018)

Is it reasonable for a company to rely on its customers and partners to make its platform open to others, or are the platform and brand too valuable (Moilanen et al., 2018, p. 21)? Open APIs have the potential to open new business opportunities. Companies that have taken a risk by exposing APIs to clients have seen significant revenue growth in recent years (Basole, 2016, p. 21). Over the previous decade, the number of provided open APIs has increased substantially (DuVander, 2012; Moilanen et al., 2018, p. 37; Santos, 2017, 2019). This trend is seen in Figure 7 from 2005 to 2023.

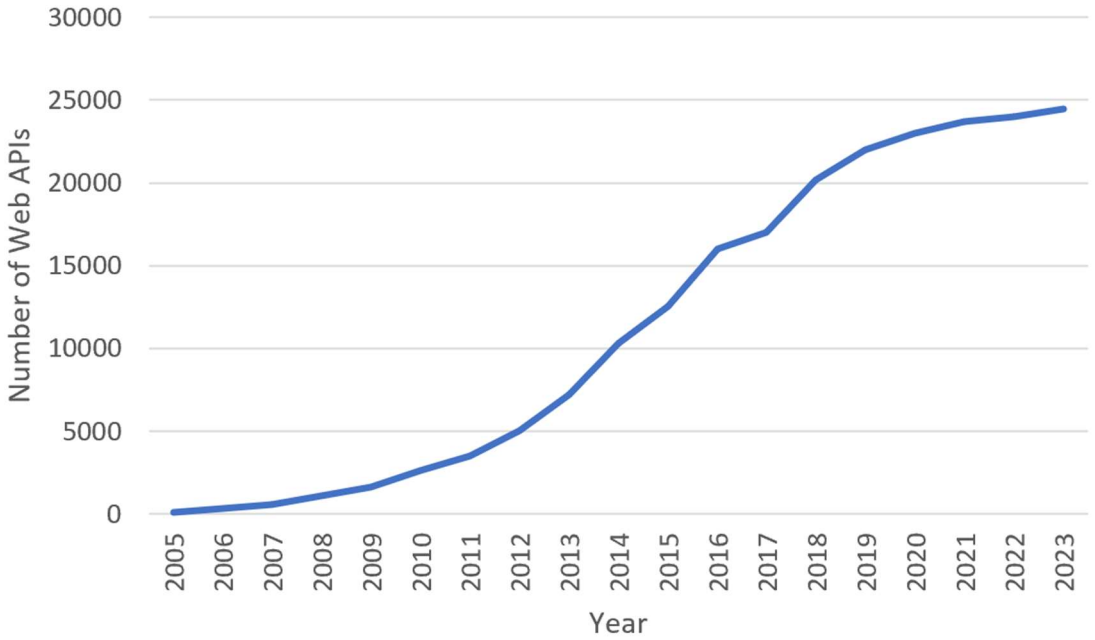


Figure 7. Illustration of API growth (Adapted from DuVander, 2012; Liu et al., 2021; Santos, 2017, 2019)

Webpage ProgrammableWeb has been monitoring this evolution and offering open APIs for developers. At the time of this research, in early January 2023, the ProgrammableWeb dictionary had over 24 471 open APIs. As Liu et al. (2021, p. 13) observed, API development has slowed in recent years, and the open API directory ProgrammableWeb announced its retirement in October 2022 (ProgrammableWeb Staff, 2022). Rapid API, another open API provider that is still active, currently provides over 40 000 APIs (RapidAPI, 2023).

Biehl (2015, p. 17) emphasizes that companies "cannot enforce their business partners to utilize their services". Therefore, the API service must be valuable, helpful, and easy to use to persuade business partners to invest in the product. The solution supplied must also be secure and simple for business partners to monitor. If a company successfully harnesses its API supply it creates opportunities for additional revenues, expanded product reach, and innovations from its research, and development business partners (Moilanen et al., 2018, p. 20). Figure 8 illustrates the benefits of open APIs.

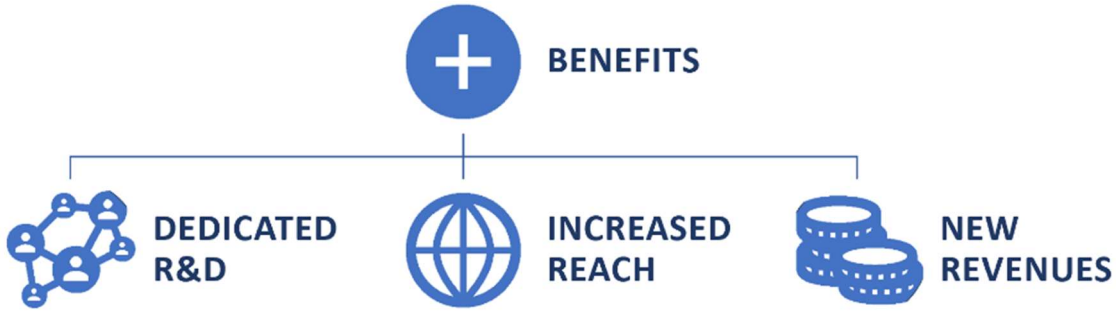


Figure 8. Advantages of open APIs (Adapted from Moilanen et al., 2018)

Most businesses intend to invest in the API economy. Participation in the API economy was a top target for 89% of telecommunication companies in 2020 (RapidAPI, 2020). Participation in the API economy was ranked as a priority by 72.2% of telecoms in 2022, with 11.1% contemplating it in the future (RapidAPI, 2022). It is no wonder why most companies try to "provide access to their company assets" to third-party developers while they can create "free" innovations for the company (Biehl, 2015, p. 18; Moilanen et al., 2018, p. 33).

What should companies then take into consideration when building APIs? Biehl (2015, p. 19) has created a list of things to consider when developing an API:

1. Learning what APIs primary customers want to use.
2. Fitting the API to the company's current API portfolio.
3. Choosing the architectural style.
4. Designing a blueprint of the API.
5. Selecting the API platform.
6. Choosing a generative API methodology.

In other words, companies should invest to listen to their customers and understand their needs. Based on customer wishes the company can then create and make the technical requirements to implement such APIs. After creating an understanding of the requirements, the company must make sure that the APIs are in line with its API portfolio offering.

Previous research has illuminated the dynamics of community building and knowledge sharing among developers in the context of mobile broadband network platforms. Alay et al. (2017, p. 77) suggest that fostering a sense of community can facilitate collaborative efforts and knowledge exchange among developers. Notably, their study revealed that developers responded positively when given the opportunity to integrate their own tools into the platform, underlining the importance of flexibility and customization.

In recent years, the research landscape has witnessed an upsurge in the investigation of radio network-specific open programmability, with a focus on xApps. Numerous studies have delved into various facets of this evolving field, examining the benefits, quality of experience (QoE), machine learning (ML) opportunities, emerging trends, market potential, future challenges, platform architecture, communication metrics, traffic steering frameworks, and guiding principles (Aryal et al., 2023; Azariah et al., 2022; Bonati et al., 2022; Dryjański et al., 2021; Iturria-Rivera et al., 2022; Kak et al., 2022; Koufos et al., 2021; Kougioumtzidis et al., 2022; Lacava et al., 2022; Polese et al., 2022; Pönnelin et al., 2023; Singh et al., 2020; Wypiór et al., 2022).

Among these contributions, Aryal et al. (2023) highlighted a fundamental challenge in this domain – “the delicate balance between system performance and complexity.” They underscored the dilemma faced in designing a high-performance environment that may lead to a complex system architecture, versus a less complex design that may sacrifice system performance. These findings underscore the intricate nature of the field and the importance of addressing these challenges.

2.3 Current concepts of digital marketing

This thesis sets out to address a notable gap in the literature by aiming to comprehend the dynamics of promoting xApp platforms and identifying the essential elements necessary for adapting xApp platforms to meet the evolving demands of the digital

marketing landscape. Through a comprehensive exploration of digital marketing strategies for open application platforms, this research seeks to contribute valuable insights into how, where, and why these platforms should be promoted, with a keen focus on the elements that drive their adaptation and success.

Maslow (1958, p. 18) presented the five basic needs of humans that are “physiological, safety, love, esteem, and self-actualization”. Each need is ranked individually. The most significant need to meet is a psychological need, whereas the least important need to fulfill is self-actualization. The five essential requirements can be represented hierarchically in the form of a pyramid (Figure 9). As the need at the bottom of the pyramid is met, the next need can be addressed until the peak of the pyramid is achieved.

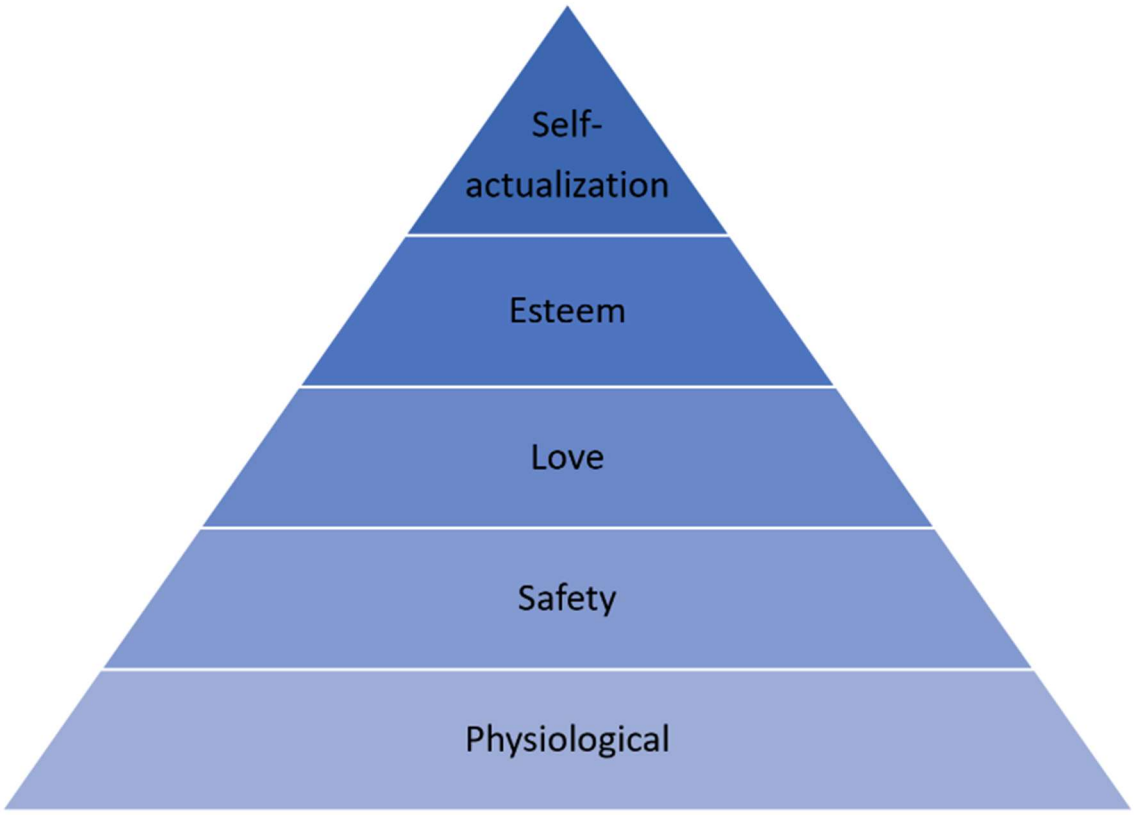


Figure 9. Maslow's hierarchy (Maslow, 1943)

Needs may have changed since Maslow's presented his theory of human basic needs. When there are needs then there is a marketing opportunity while “marketing is

managing profitable customer relationships" by "satisfying customer needs" (Parment et al., 2016, pp. 9–10).

According to Kotler Hermawan and Setiawan Iwan et al. (2016) brand strategy is handled through segmentation and targeting. Segmentation is a term related to marketing that involves "dividing the market into homogenous groups based on their geographic, demographic, psychographic, and behavioral profiles". Targeting is related to segmentation by selecting a segment group to target their market based on segment "attractiveness and fit for the brand". This allows marketers to manage resources more productively and target different segments with different contributions.

Parment et al. (2016, pp. 24–25) divide customers into butterflies, true friends, strangers, and barnacles based on their profitability and loyalty (Figure 10). Strangers are the least loyal and lowest profitable group out of the four, which makes it a group that should not be invested in any form. Barnacles are long-term customers but less profitable. This makes them very problematic. Butterflies offer profit but they tend to change companies from time to time. This makes butterfly customers hard to catch and stay loyal as they should be invested in shorter periods. True friends are what companies should invest in the most as they are both loyal and offer high profitability.

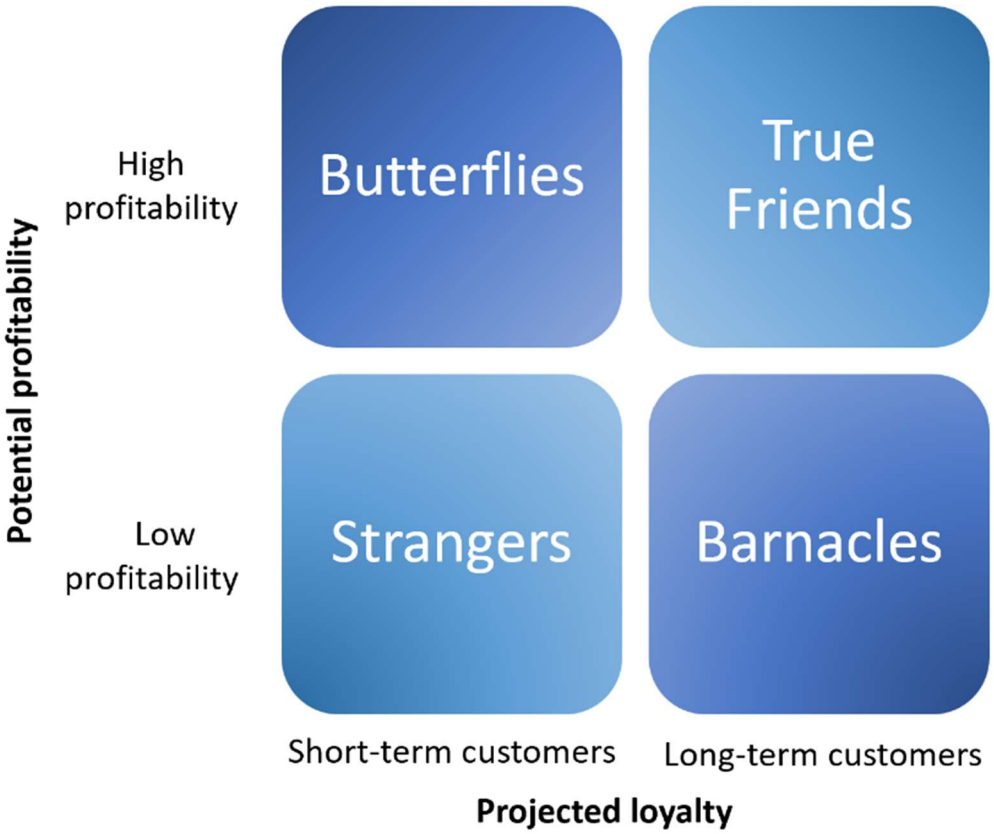


Figure 10. Customer relationship groups (Adapted from Parment et al., 2016)

Kotler Hermawan and Setiawan Iwan et al. (2016, pp. 167–168) mentions the WOW factor as something that leaves people speechless as what customers are constantly looking for. While the WOW feeling is personal, companies should offer something enjoyable, a superior experience, and engagement. This can be achieved by concentrating on user experience. However, companies should focus on their desired group of customers and try to build their relationships with them instead of focusing on providing to every customer (Parment et al., 2016, p. 25).

While there are different customers their adaptation to a specific technology might differ (Chaffey & Chadwick, 2022, p. 106). Some people adopt technologies early as they are innovators who want to get the latest technology (Rogers et al., 2014, pp. 11–12). After innovators, the next adaptors of the technology are the early majority, late majority, and laggards who take the most time adapting to the new technology (Rogers et al., 2014, pp. 11–12). Figure 11 illustrates the diffusion-adoption groups and their contribution.

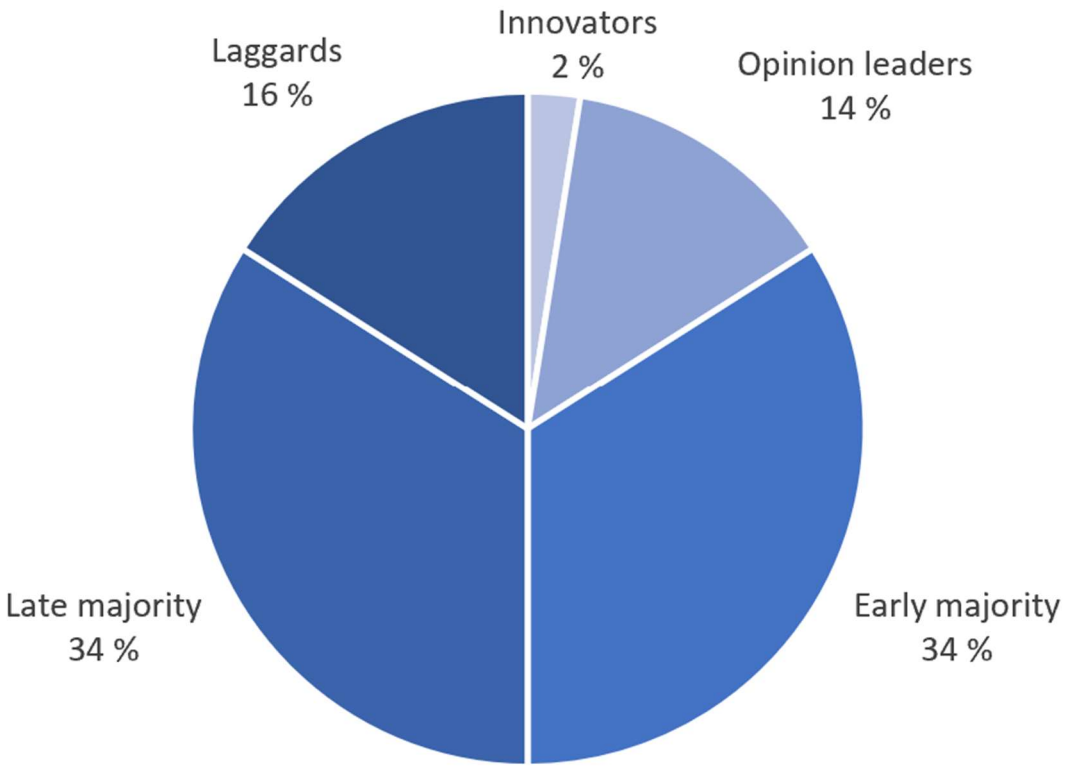


Figure 11. Diffusion-adoption groups (Adapted from Rogers et al., 2014)

Digital marketing is marketing that uses digital technology (Lahtinen et al., 2022, p. 17). Chaffey and Smith (2023, p. 3) extend this notion by stating that digital marketing is marketing. Yet, "digital marketing should be blended with conventional marketing" because the two are complementary (Chaffey & Smith, 2023, p. 3).

Digital marketing has transformed traditional marketing. Digitalization has transformed the purchasing process by giving the means to obtain information and compare options (Lahtinen et al., 2022, p. 56). Digital advertising, content marketing, and social media have increased customer reach (Lahtinen et al., 2022, p. 56). Customers are also more aware of their wants because of digitalization (Lahtinen et al., 2022, p. 56).

Lahtinen et al. (2022, pp. 20–24) classified digital marketing progression into three stages. The first stage began in the middle of 1990, with the launch of the first websites and search engines (Lahtinen et al., 2022, p. 20). Only the most resilient organizations

survived this period, recognizing that not all corporate tasks could be shifted to the web (Lahtinen et al., 2022, p. 21).

According to Lahtinen et al. (2022, pp. 22–23) the second stage of digital marketing began in the middle of 2000 with the launch of social media platforms. Initially, social media networks did not advertise anything because the marketing potential of these channels was not yet recognized. At this time social media platforms were only utilized for communication.

The third stage of digital marketing has come from the adaptation of the internet and user content according to Lahtinen et al. (2022, pp. 23–24). The third stage began around 2015 when the internet transitioned to the mobile environment. Marketers recognized the advertising potential of social media platforms and began to concentrate on content marketing. The most recent advancements in this process have been commercial partnerships and influencer marketing. Marketing is nowadays transitioning from fragmented to personal, experience, and recommendation.

According to Labrecque et al., 2013, pp. 258–259) customers have gained power in demand, information, network, and crowd level to help their decision-making. Demand power relates to the trend of multiple options to choose from instead of relying on local resources. Information power means that customers have increased reach for data. Network power involves the feedback and reviews that customers leave based on their experience with the company. Crowd power affiliates with the communities and platforms that share information. All these powers influence customers’ buying decisions.

Businesses must devise new techniques to gain access to their customers' networks (Lahtinen et al., 2022, p. 57). Nowadays segments are communities (Kotler Hermawan; Setiawan Iwan et al., 2016, p. 47). These communities form naturally while customers specify the kind of communication in which they wish to participate (Kotler Hermawan;

Setiawan Iwan et al., 2016, p. 48). This has changed the transition from push marketing to permission marketing where the company asks permission from customers to send marketing messages (Kotler Hermawan; Setiawan Iwan et al., 2016, p. 48).

Digitalization has intensified competitiveness and globalized it (Lahtinen et al., 2022, p. 49). Technology is rarely the answer, but a tool that allows a company to achieve its objectives more effectively and with higher quality (Lahtinen et al., 2022, p. 45). Marketing models may aid in the identification of prospective technologies by breaking down the adaption process into phases and quantifying the change (Lahtinen et al., 2022, p. 48).

Ye and Kishida (2003) delved into the underlying motivations that propel individuals to engage with open-source software (OSS) communities, revealing that learning stands as a pivotal driving force. Their research highlighted that, even in the absence of monetary incentives, the pursuit of knowledge and skill development offers immediate satisfaction and the potential for personal growth within the platform community.

While a substantial body of literature exists on digital marketing, surprisingly little research has been dedicated to the digital marketing strategies specifically tailored for open application platforms when targeting customers. Open application programming interfaces (APIs) have found their place within discussions in platform competition strategies, managing, advertising, marketing, and semantic marketing context as well as being accused of using unfair methods to compete against rivals (Bodle, 2011; Chesbrough, 2004; Foux, 2010; Karhu & Ritala, 2021; Lies, 2019; Sharma, 2019; Simon, 2021; J. Wang, 2009). Motivational factors that inspire users to actively engage and collaborate within open innovation platforms have also garnered scholarly attention (M. Antikainen et al., 2010; M. J. Antikainen & Vaataja, 2010). These studies shed light on the intricacies of what drives participation in open innovation, a concept with clear relevance to the realm of open application platforms.

2.4 User journey

User journey is about user experiences that a person has when interacting with the system software. The user journey outlines the main activities that the customer wants to accomplish by visiting the website (Chaffey & Chadwick, 2022, p. 500). These customer scenarios typically consist of series that require information or experience to effectively fulfill the desired goal (Chaffey & Chadwick, 2022, p. 500). The user journey collects all the touchpoints and channels that customers are using and considers the improvement possibilities to enhance these interactions (Benyon, 2019, p. 89).

The user journey, as previously said, collects all touchpoints and seeks to grasp the path that leads to the next touchpoint (Benyon, 2019, p. 93). This is known as navigation (Benyon, 2014, p. 328, 2019, p. 93). Brinck et al. (2002, pp. 126-127) identify seven distinct navigation techniques that users employ, ranging from procedural recall to content learnability. As a result, users should be helped, and the web page should have labels, navigation assistance, and a search function (Rosenfeld & Morville, 2002).

One of the biggest problems in user journey mapping is the constant change since users' interests and needs change constantly while only a "few users will follow the ideal journey carefully" (Benyon, 2019, p. 92). Sometimes customers decide to try new opportunities which may modify the user journey (Benyon, 2019, p. 92). This makes forecasting user journeys challenging.

Another crucial consideration that complicates user journey mapping is distinguishing distinct user groups, as some users may function as both consumers and content providers (Benyon, 2019, p. 93). Side switching refers to "the phenomena in which users from one side of the platform join the opposing side" (Parker et al., 2016, p. 299). User roles can change, and the platform should allow side-switching.

Lewko and Parton (2021) created the developer journey map to illustrate the user journey of a developer. The map visualizes developers' perspectives and helps to gain

perspectives of the user experience (Parton, 2021). While Lewko and Parton divided the developer journey into five stages I added four more stages to the map which are registering, onboarding, testing, and feedback. The adapted developer journey map can be seen in Annex 2. The modified developer journey consists of seven stages:

1. Awareness: The developer becomes aware of the open API platform's existence. This awareness can be created through online advertisements, social media, word-of-mouth, events, conferences, or other channels. The developer becomes interested in learning more about the platform and may begin to research it further and determine whether it could be useful for their needs.
2. Exploration: The developer visits the website and begins exploring the open API platform. At this stage, the developer looks through the provided features, documentation, and other resources. The developer tries out the platform demo or sandbox environment. A developer may consider platform features, pricing, billing, reputation, reviews, competitors, and other alternatives before committing to the platform.
3. Registration: At this stage, the developer creates an account on the open API platform and provides any necessary information. Providers may ask for information such as the developer's name, email address, and company or organization name. Depending on the platform, the registration process may also involve providing payment information, agreeing to terms and conditions, and completing a verification step such as email confirmation.
4. Onboarding: Once the developer has registered for the open API platform, the onboarding process typically involves familiarizing themselves with the platform's features, capabilities, and documentation. The open API platform provider may offer onboarding resources such as tutorials, webinars, or one-on-one support to help the developer get started.
5. Integration: The developer begins integrating the open API platform into their software. This is the stage where developers encounter their first challenges and difficulties. The developer may encounter challenges related to the API's functionality, performance, or compatibility with their software. The open API

platform provider may offer technical support, documentation, or other resources to help overcome these challenges.

6. Testing: The developer tests their integrated software to ensure that it works correctly and that it meets their requirements. The developer may need to make changes or modifications to their integration based on the test results. Testing may involve an error, exception, scenario and use case handling, unit, integration, performance, and user acceptance testing, and testing in a controlled setting.
7. Deployment: The developer deploys their software to production for their end-users by using the open API platform which provides the functionality. The deployment stage includes configuring and coding in the production environment to make the final product. The developer may continue to monitor and optimize their integration over time.
8. Feedback: The developer provides feedback to the open API platform provider. At this stage developers usually report their issues and suggest improvements. Providing feedback to the open API platform provider can be beneficial for both the developer and the provider. The developer can help identify areas for improvement, while the provider can use feedback to enhance their platform and better meet user needs. Some open API platform providers may even offer rewards or incentives for feedback. The developer may also be willing to participate in user communities or forums to share their experiences and learn from others.
9. Scaling: As the developer's usage of the open API platform grows, they may need to consider factors such as pricing, performance, and reliability. If the developer is satisfied with the open API platform's performance and features, they may renew their subscription or continue to use the platform in future projects. If not, they may look for alternative solutions that meet their needs better.

2.5 Benchmarking

The benchmarking process is used in this study to compare xApp platform to similar practices other companies are using. Benchmarking is defined as "the practice of

comparing the company's goods and procedures to those of rivals or top enterprises in other industries" to improve resource allocation and quality (Parment et al., 2016, p. 447). Benchmarking competitors is an excellent technique to begin learning about the company's marketing features (Kotler et al., 2009, pp. 142–143). What distinguishes a great and good marketing strategy is that instead of just benchmarking the company should aim to outperform the competition (Kotler et al., 2009, pp. 142–143).

Developer experience (DevEx) is a significant competitive factor in interface adoption (Moilanen et al., 2018, p. 25). This thesis will analyze the developer sites of Amazon Web Services (AWS), Google, RedHat, and Microsoft to gain a better understanding of what other firms provide their software developers as best practices. The most important findings regarding each developer portal supported front-end and back-end languages, social channels, engagement opportunities with other developers, useful material and qualifications can be seen in Annex 3. There are some similarities and clear differences between each developer portal providing.

The most differences concern the provided useful material. Google provides the least social channels out of the four but arranges big events internationally and has large communities. Google is lacking certifications while offering badges. Amazon has been using APIs internally which eases external integration (Biehl, 2015, p. 17). AWS has the most social channels out of the four and a huge variety of useful materials. AWS also provides multiple ways for developers to find like-minded people virtually. Microsoft stands out by providing the possibility to code with any programming language and by offering comprehensive support for developers. RedHat has useful material like e-books, use cases, a sandbox, and learning paths.

Every platform is built around communities. They offer support for each developer and tester and give a place to discuss matters that the platform is offering. Every platform also offers events where like-minded people can meet and feel a sense of belonging. These events are usually held virtually but some companies like Google, Twilio, RedHat,

and AWS even offer possibilities to meet other platform users face-to-face. Microsoft and Twilio take events even further by offering community-led after-parties. The main reason is to get people to feel connected and involved in the platform offering.

Social media platforms aid in spreading the news to as many like-minded people as possible. YouTube, Facebook, Twitch, Soundcloud, Stack Overflow, Reddit, GitHub, Twitter, Peer Insights, Instagram, LinkedIn, and Discord are among the platforms' well-known social channels. YouTube and Twitter are used by every platform firm, from AWS to Twilio. Platforms utilize YouTube to distribute announcements, knowledge, and lessons, as well as tech presentations, podcasts, and instructions about apps and platform use. Twitter is used for discussions to disseminate development-related material, case studies, research, guidance, instructions, tutorials, career insights, and news. Google, RedHat, Microsoft, and Twilio each maintain a platform blog and host tech talks.

2.6 Service blueprint

Service blueprint is used in this study to gain an understanding of how the service is defined in customers eyes and diagnosing problems that may affect operational efficiency. A service blueprint is a widely used customer-driven approach to get an understanding of the tasks that are implemented from a customer’s point of view (Bitner et al., 2008, p. 67). Service blueprint creates a customer experience journey that visualizes the service process, customer actions, interactions, support processes, and stages of customer engagement in a structured manner (Bitner et al., 2008, p. 67). This makes the customer’s journey easier to follow and helps to visualize what affects the customer's perception of quality (Bitner et al., 2008, p. 73).

Customer actions are the central of service blueprinting. Customer actions conclude “all the steps that customers take a part in the service delivery process” (Bitner et al., 2008, p. 72). Every time the customer interacts directly there is a moment of truth situation

(Bitner et al., 2008, p. 72). The line of interaction divides customer actions and front-stage actions (Bitner et al., 2008, p. 72).

Front-stage interactions are interactions that are visible to the customer while backstage interactions are that are not visible to the customer (Bitner et al., 2008, p. 72). Both front stage and backstage interactions are important fulfilling the ultimate experience for the customer. The line between visible and non-visible interactions is called the line of visibility (Bitner et al., 2008, p. 72).

Support processes are activities that must take place by individuals and units inside the company to make it possible for the service to be delivered (Bitner et al., 2008, p. 72). The line that divides backstage actions from the support processes is the internal line of interaction (Bitner et al., 2008, p. 72). Physical evidence is what customers encounter when using the service (Bitner et al., 2008, p. 73).

The stages of the service blueprint can be divided into previously described stages of the developer journey. The service blueprint can be found in Annex 4. The service blueprint consists of eight stages that are now highlighted by customer action points:

1. Noticing advertising: This is the phase where the customer has their first interaction with the company. Physical evidence can for example be social media or a search engine. Front-stage interactions that are visible to the customer are marketing efforts. The backstage interactions that are not visible to the customer include creating marketing content and advertising. The support process for this phase is the marketing plan.
2. Visiting the website: In this phase, a customer enters the website for the first time. The customer has been intrigued by the advertising or heard of the product and wants to see the product for themselves. Physical evidence can be, for example, a website or an application. Front-stage interactions include greeting the visitor and asking for cookie preferences. Backstage interactions include

setting the cookie preferences and therefore tracking the customer. The support process for this phase is visitor analytics.

3. Registering to the platform: In this phase, the customer registers to be able to explore the platform a bit more. Physical evidence is the registration system. Front-stage actions include filling out the needed information to get the rights to register. Backstage actions include collecting all the needed information and storing it in a customer database for further use granted by the customer. The support process for this phase is therefore the customer database.
4. Signing in: This is the phase where the customer lands on the first page of the platform and explores the page. Physical evidence is the landing page. Front-stage interactions include customers scrolling through the provided material that they find interesting and useful. Backstage interactions include efforts to introduce pages that might interest the customer and get them engaged using the platform. Support processes for this phase are customer analytics that help to make this type of recommendation.
5. Analyzing the features: In this phase, the customer analyzes the features and provided material to choose to try the product. The physical evidence of this phase is the product description, which needs to be as appealing as possible. Front-stage interactions include checking the available data to make the choice. Backstage interactions include creating an appealing introduction that assures the customer that investing time and money into this platform is a smart choice. The support process of this phase is traffic scanners that help to investigate the used amount of time on the page and the success rate of making the customer try out the product after they have analyzed the features.
6. Sandbox: In this phase, the customer interacts with the API itself for the first time. Physical evidence is the sandbox that provides a place to test if the platform provides enough capabilities to do what the customer wants to achieve. Front-stage interactions of this phase are reading the available documentation and watching tutorials that help to try out the sandbox. Support processes include

documentation, tutorials, and tools that need to be described so that the customer understands them and can create something based on it.

7. Creating the first application: This is the phase where the customer decides to invest in doing something more concrete and tries to create their first application on the platform. The physical evidence of this phase is the APIs that gather the needed information for the customer. Front-stage interactions include checking the features and opportunities that the platform has to offer. This is the crucial part where the customer needs to feel they get enough help, and that the solution is what they looked for. The support process for this phase is customer satisfaction metrics that collect useful information for further development of the platform since this is a crucial phase.
8. Giving feedback: This is the phase where customers can raise any concerns or feedback about using the platform. Physical evidence of this phase is on the feedback page. Not all customers are willing to give feedback and usually, the feedback that is received can be mostly negative. Customers, however, value companies that try to fix issues and get heard. Front-stage interactions measure the received service and the quality of it. Backstage interactions include improving the customer experience based on this feedback. Support processes of this phase are reviews and feedback that help the company to form an opinion on where the platform is heading and what to improve.
9. Continuing to use the platform: This is the phase where the customer decides to use the platform in the future and possibly reviews their subscription if they have any. Physical evidence can be, for example, a website or an application, or the used platform. Front-stage interactions include receiving the subscription. Backstage interactions include continuing to create appealing opportunities for the customer to keep them happy and engaged to the platform in the future. The support process for this phase is the customer retention rate which defines how many customers have continued to use the services.

3 Research methodology

This chapter is going to be explaining the research philosophies and methodologies. Based on these research philosophies and methodologies, the thesis goes through the used approach, data collection, and research settings. Research settings are going to introduce the conducted research and special characteristics of the case company. Lastly, the thesis is going to question the reliability and validity of the results.

3.1 Research philosophy

A paradigm is “a set of basic beliefs” that connects with the way the individual views the world, what role the person has in the world, and what different relationships the person can create in “that world and its parts” (Guba & Lincoln, 1994, p. 107). Each human has their own perceptions, beliefs, and own will which make each person hard to predict (Järvinen, 2001, p. 196). However, three questioning paradigms that can be classified from the response (Guba & Lincoln, 1994, p. 108). The philosophy paradigm depends on choices in three questions: *ontological, epistemological*, and *methodological* (Denzin, 2008).

Ontological belief is considering the world to be detached and separate from humans or subjective to only the changes humans make in it (Burrell & Morgan, 1979; Orlikowski & Baroudi, 1991). Ontology is in other words “the study of being” and “what is there to know about it?” (Crotty, 1998; Guba & Lincoln, 1994, p. 108). To ontology, only real-world problems matter that concern real existence and actions (Guba & Lincoln, 1994, p. 108).

Epistemological belief is considering the one who is questioning the matters and what accurate information about the phenomenon can be known (Guba & Lincoln, 1994, p. 108; Järvinen, 2001, p. 197). Epistemology wants to know “how things really are and how things really work” if the real world is assumed (Guba & Lincoln, 1994, p. 108).

Epistemological belief can be seen through positivism, anti-positivism as well as critical ones (Burrell & Morgan, 1979, p. 5; Orlikowski & Baroudi, 1991).

Methodological belief is considering how the one who is questioning the matters can find out what can be known by "obtaining first-hand knowledge of the subject under investigation" (Guba & Lincoln, 1994, p. 108; Järvinen, 2001, p. 197). In methodology, if the real world is assumed the one who is questioning "mandates control of possible confounding factors, whether the methods are qualitative or quantitative" (Guba & Lincoln, 1994, p. 108). The methodology is in other words "a preference for certain methods", "a theory of scientific knowledge", "a range of solutions" and "a systematic sequence of procedural steps" (Gobo, 2008).

Table 1. Basic beliefs of alternative questioning paradigms (Adapted from Guba & Lincoln, 1994)

ITEM	POSITIVISM	POSTPOSITIVISM	CRITICAL THEORY ET AL.	CONSTRUCTIVISM
ONTOLOGY	Naive realism • "Real" reality but apprehendable	Critical realism • "Real" reality but only imperfectly and probabilistically apprehendable	Historical realism • Virtual reality shaped by social, political, cultural, economic, ethnic, and gender values • Crystallized over time	Relativism • Local and specific constructed realities
EPISTEMOLOGY	Dualist/Objectivist • Findings true	Modified dualist/Objectivist • Critical tradition/Community • Findings probably true	Transactional/Subjectivist • Value mediated findings	Transactional/Subjectivist • Created with findings
METHODOLOGY	Experimental/Manipulative • Verification hypotheses • Chiefly quantitative methods	Modified experimental/Manipulative critical multiplism • Falsification of hypotheses • May include qualitative methods	Dialogic/Dialectical	Hermeneutical/Dialectical

Table 1 provides the four paradigms *positivism*, *postpositivism*, *critical theory,* and *constructivism* view to ontological, epistemological, and methodological questions (Guba & Lincoln, 1994, pp. 108–109). *Positivism* tries to find regularities in the social world while "what we see is true" because it can be seen (Burrell & Morgan, 1979, p. 5; Metsämuuronen, 2000, p. 11). *Postpositivism* or so-called *anti-positivism* is the criticism of positivism (Guba & Lincoln, 1994, p. 109). Postpositivism understands the social world directly from individuals' perspective and that there can happen certain events that are

harder to explain but can still be observed (Järvinen, 2001, p. 197; Metsämuuronen, 2000, pp. 11–12). The *Critical theory* explains the social world as historically continued and is a "set of several paradigms including Marxism, feminism, materialism, and participatory inquiry" (Chua, 1986; Guba & Lincoln, 1994, p. 109). *Constructivism* is "an alternative paradigm whose breakaway assumption is the move from ontological realism to ontological relativism" with assuming reality to be relative while others assume it is real (Guba & Lincoln, 1994, p. 109; Metsämuuronen, 2000, p. 12). This thesis is going to focus on using qualitative and quantitative research methods. The qualitative method is related to *critical theory* and *constructivism* while the quantitative method is related to *positivism* and *postpositivism* (Metsämuuronen, 2000, p. 14).

3.2 Research settings

This thesis is trying to describe the problem scope objectively, research the focus group and holistically discuss the conclusions. The focus group is going to answer an interview, a survey, or both based on their choice. The focus group consists of people that may work in the same company as the interviewee, but all the interviewees and the interviewer do not know beforehand. This applies also to the survey. The interview questions and survey were made using prior survey and interview question forms and theories. The focus group answered these questions, and the results were analyzed with the literature reviewed earlier.

Quantitative research uses "quantitative data in the form of survey methods, laboratory experiments, formal methods and numerical methods such as mathematical modeling to understand and explain natural phenomena" (Myers, 1997, p. 241). Qualitative research uses "qualitative data in the form of interviews, documents, and participant observation data to understand and explain social and cultural phenomena" (Myers, 1997, p. 241). This thesis is going to use both quantitative and qualitative methods.

Both qualitative and quantitative methods can be added to the same research. This is something called mixed methods (Silverman, 2015, p. 46). Table 2 shows four ways in

which qualitative and quantitative approaches can be combined in the data collection and analysis phases. The most common way to use the mixed model is to gather and analyze both qualitative and quantitative data separately (Brannen et al., 1992; Hirsjärvi & Hurme, 2008, p. 29). This is shown in Table 2 with models 3 and 4.

Table 2. Four ways to combine qualitative and quantitative approaches in data collection and analysis phases (Adapted from Hirsjärvi & Hurme, 2008)

	MODEL					
STEP	1		2		3	4
METHOD	Qualitative		Qualitative	Quantitative	Qualitative	Quantitative
ANALYSIS	Qualitative	Quantitative	Qualitative	Quantitative	Qualitative	Quantitative

In Table 2 models 1 and 2 show how qualitative and quantitative approaches can be combined (Hirsjärvi & Hurme, 2008, p. 29). In model 1 the data has been gathered qualitatively and is being analyzed qualitatively and quantitatively whereas in model 2 the data is summarized in conclusions while the data has been gathered both qualitatively and quantitatively and analyzed qualitatively and quantitatively (Hirsjärvi & Hurme, 2008, p. 29). This thesis is going to be using both qualitative and quantitative methods in a way model 2 has shown.

Table 3 shows the use cases for both quantitative and qualitative methodologies of four different methods: observation, textual analysis, interviews, and audio and video recordings. The quantitative methodology does not find observation as the most suitable or the most reliable form to conduct research with a bigger sample group (Silverman, 2015, pp. 43–44). Observation can however act as a tool for framing the questionnaire (Silverman, 2015, pp. 43–44). In qualitative research, observation can be seen as a tool to gain an understanding of the culture (Silverman, 2015, p. 44). This thesis is going to use observation solemnly as guidance to form the questionnaire and interview questions.

Table 3. Use cases for quantitative and qualitative methods (Adapted from Silverman, 2015)

RESEARCH METHOD	QUANTITATIVE	QUALITATIVE
OBSERVATION	Preliminary work, e.g., prior to framing questionnaire	Fundamental to understanding another culture
TEXTUAL ANALYSIS	Content analysis, i.e., counting in terms of researchers' categories	Understanding participants' categories
INTERVIEWS	Survey research: mainly fixed-choice questions to random samples	'Open-ended' questions to small samples
AUDIO AND VIDEO RECORDING	Used infrequently to check the accuracy of interview records	Understanding the organization of talk, gaze and body movements

Silverman (2015, pp. 43–44) states that content analysis is a method where a group of people is divided into categories to count how many instances fall into each category. Content analysis is used in quantitative methodology whereas qualitative methodology uses categories to gain a better understanding of the participants. Categorizing the sample groups is going to be a crucial factor while analyzing the results in this thesis while it affects the user experience and digital marketing strategy provided for each target group.

3.2.1 The case company and the open application platform

The case company is a global leader in B2B technology innovation and offers technology solutions leveraging its expertise across mobile, fixed, and cloud networks. The case company has long-term research efforts and wants to create value for its global customers, partners, and stakeholders. Overall, the case company's commitment to innovation and collaboration enables it to stay at the forefront of technological advancements and help its customers and partners achieve their goals.

The company in question has earned the trust of global service providers, enterprises, and partners for its track record in delivering networks that are secure, dependable, and environmentally sustainable. Concurrently, it collaborates with these entities to shape the landscape of future digital services and applications. Consequently, the company

assumes a pivotal role in propelling digital transformation across diverse industries and sectors, spanning telecommunications, manufacturing, transportation, and healthcare. Presently, the company is embarking on exploratory research into a promising business opportunity within cloud networks, with aspirations to cultivate a specialized platform leveraging its wealth of expertise.

The case company has made a brand renewal over the spring of 2023 and has been vocal about the want to share expertise to drive development by opening networks for innovation and collaboration. The mission in the future is to gain a collaborative advantage by bringing together customers and partners on the same platform to create digital applications. This approach has been modified from the O-RAN alliance’s goal.

The O-RAN Alliance, as of 2023, represents a global consortium comprising more than 300 stakeholders, including mobile network operators (MNOs), vendors, research institutions, and academic bodies, all actively engaged in the radio access network (RAN) sector. O-RAN's principal mission revolves around expediting the development of intelligent, open, virtualized, and seamlessly interoperable mobile networks. By pursuing this objective, O-RAN seeks to bolster the effectiveness of RAN deployments and operations for mobile operators, while simultaneously nurturing a competitive and dynamic ecosystem of RAN suppliers. This ecosystem is instrumental in driving innovation and enhancing the overall user experience.

The case company has been a part of the O-RAN Alliance since 2018 and was one of the first leading RAN vendors to join the community. Aira Technologies and VMware in collaboration with Intel revealed their plans in February 2023 to launch a platform that would allow users to create applications (xApps) to improve RAN (PRWeb, 2023). The case company has seen this as a potential revenue stream and has been developing its platform idea regarding xApps and created two applications concerning resource allocation and issue detection.

RAN intelligent controller (RIC) is a component of open RAN that gives third parties the possibility to deploy applications (Kompany et al., 2021, p. 7; PRWeb, 2023). RIC can be either non-real time (non-RT RIC) or near-real time (near-RT RIC) (PRWeb, 2023). Whereas non-RT RIC manages use cases that take over a minute, near-RT RIC handles use cases that take less than a minute to deliver information from the sender to the recipient (Kompany et al., 2021, p. 6). Both RICs use AI/ML-based radio resource management to generate applications (Kompany et al., 2021, p. 6). The applications that can be generated with non-RT RIC are called rApps and the applications that can be generated with near-RT RIC are called xApps (Kompany et al., 2021, pp. 6–7).

The case company is especially interested in xApps and wants to understand the possibilities it could give them in the future. The business idea is visualized in figure 19. UE makes a connection to the mobile network base transceiver station (BTS) with technology that is accessible to use and based on what the UE supports. The radio technology can be for example 4G or 5G radio technology as shown in the picture. The base station gathers data on the quality of the connection to different UEs and stores this information in a cloud database. Cloud data can be used with the use of open APIs. Customers can therefore innovate and create their own desired applications (xApps) with the help of APIs.

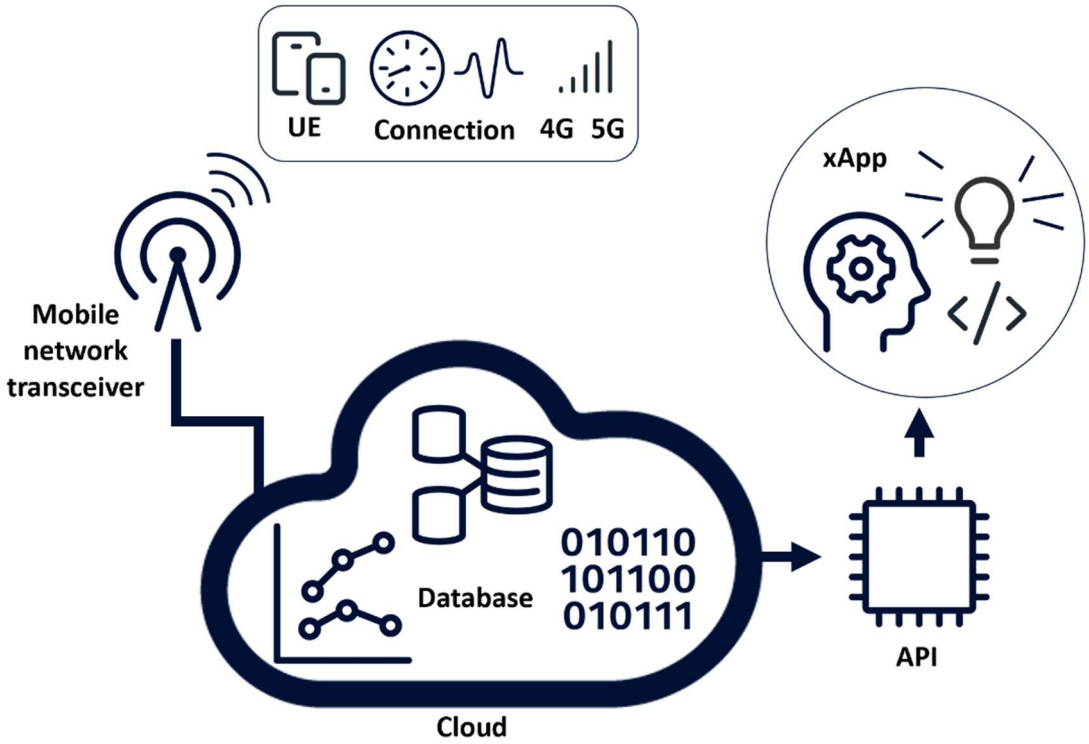


Figure 12. A simplified picture of the xApp business idea

According to Smith (2022, pp. 17–18) the most common application logic consists of data access layer, business logic layer and user interface layer. Each layer has their own responsibility, and each layer can interact with the layer that is beside them. For example, user interface cannot directly interact with data access layer but can get the data through business logic layer.

Figure 20 provides a visual representation of the various components and elements integral to the architecture of xApps. Within this architectural framework, we observe distinct layers, each contributing distinct functionalities to the succeeding layer. The layers depicted include data access, the xApp framework, the business layer, and the user interface. Notably, the xApp framework serves as an intermediary layer positioned between the data access layer and the business layer. This strategic placement enables data accessibility through the xApp framework while affording individual developers the flexibility to incorporate their unique business logic. At the user interface layer, interaction with end-users occurs, facilitating the presentation and visualization of xApps to the user.

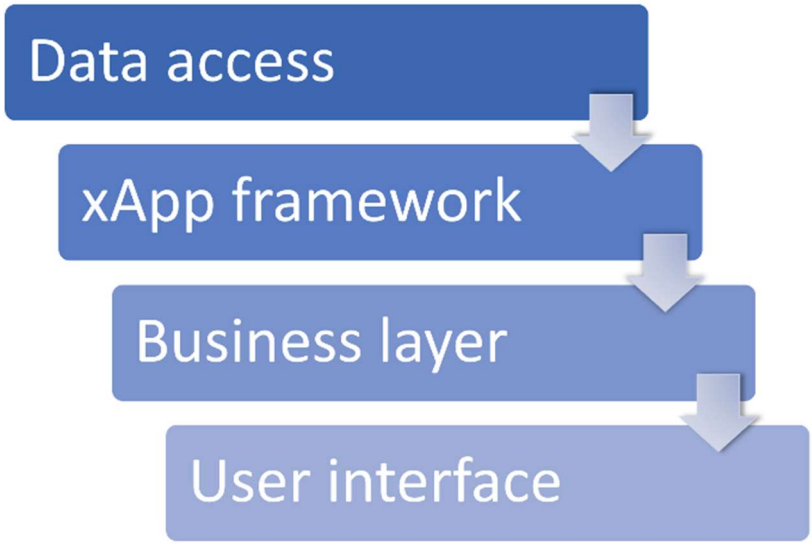


Figure 13. Illustration of application layers in the xApp architecture (Adapted from Smith, 2022)

Figure 21 represents the use case diagram of xApp business process with different actions and interactions between various elements. The arrows connecting the sections indicate the flow and progression of the process. The actors are SW developers, testers and architects, platform owners and business partners. SW developers, testers and architects can search for customers, join, and add communities, develop, design and test xApps, ask for help, register and login to the platform.

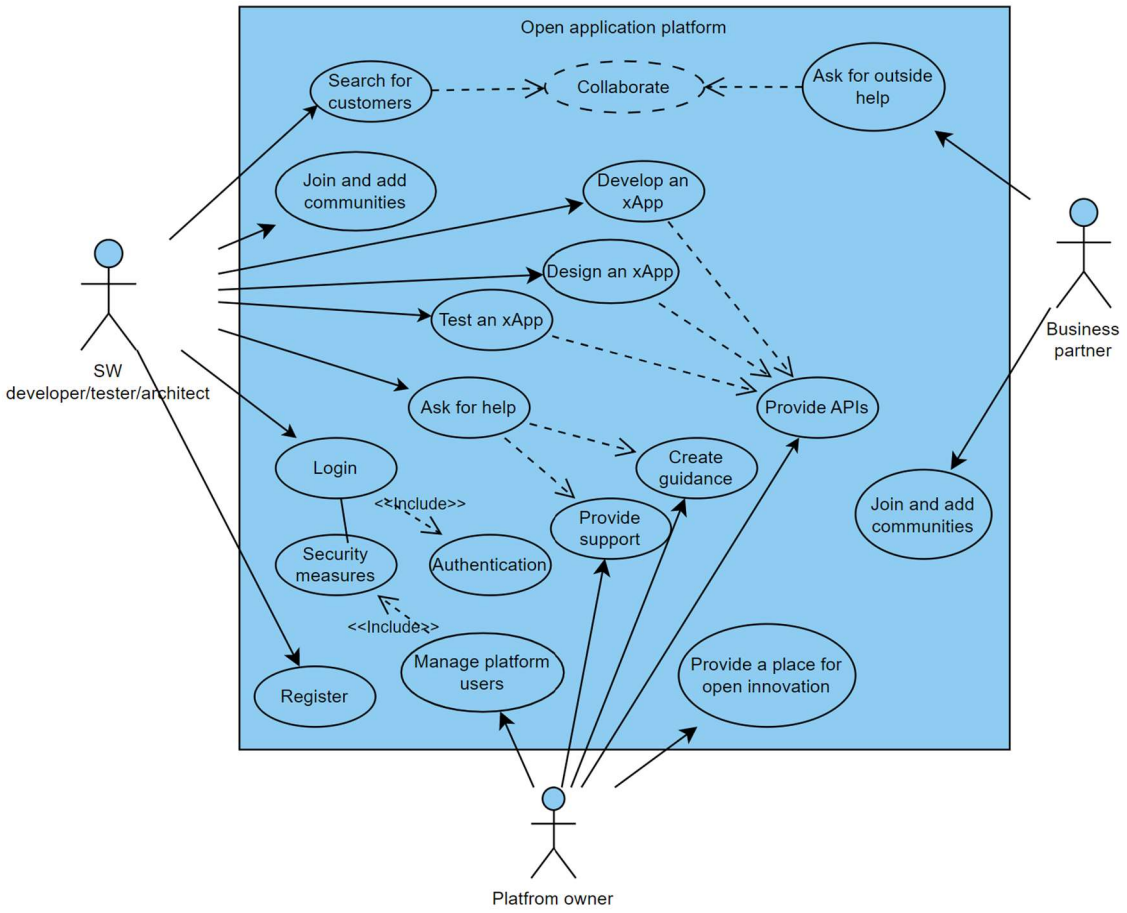


Figure 14. Use case diagram of open application platform

Platform owners provide a place for open innovation and make it possible to develop, design, and test xApps by providing APIs. Platform owners manage platform users. This includes security measures like authentication regarding logging to the platform. Platform owners can also provide support and guidance for SW developers, testers, and architects when they ask for help with platform use.

Business partners can join and add communities and have several users by the same company. Business partners seek and ask for help outside their network in their xApp projects. When supply and demand of business partners demands and SW developers, testers and architects offering meets this can create a collaboration where the SW developer, tester, and architect works for the business partner to create an xApp solution. It must be noted that individual SW developers, testers and architects can also act as

business partners if they are also running a business. Multiple business partners can also join forces on a mutual project.

Kompany et al. (2021) have been researching MNOs plans to deploy open RAN and use near-RT RIC in the future. While xApps are based on O-RAN, xApps are not limited to O-RAN. In the future, the scope of xApps will expand. 35 percent of MNOs are planning to use near-RT RIC in their open RAN deployment while 20 percent of MNOs plan to use near-RT RIC in specific use cases in their open RAN deployment. 31 percent of the MNOs are considering using RIC and deploying open RAN. Most MNOs are going to be deploying their near-RT RIC between the years 2021—2026.

Research shows that there is a possibility to persuade this market since 46 percent of MNOs are active in open RAN and 26 percent are considering open RAN while 14 percent are evaluating open RAN (Kompany et al., 2021, p. 4). xApps serve low-level access to the radio performance and capabilities to improve RAN functionalities. xApps help the RAN community to innovate together, but the main driver of MNOs' desire to develop a platform around them is the total cost of ownership (Kompany et al., 2021, p. 5). Most MNOs are expecting 10—30 percent efficient use of their assets (Kompany et al., 2021, p. 12).

According to LightCounting (2020) there have been several questions linked to xApp control over the years. Questions raised have concerned security, robustness, technical standards like radio control maintenance, radio unit concerns, the maximum number of applications, and simultaneous application running. There has also been criticism towards deploying use cases that can be deployed already with non-RT RIC.

Table 4. Illustration of xApp market growth (Adapted with permission from Crawshaw, 2022b)

	YEAR							
xApp	2020	2021	2022	2023	2024	2025	2026	2027
REVENUE ($M)	0.3	1	2	10	37	81	142	215
YEAR-OVER-YEAR GROWTH	0%	233%	100%	375%	286%	122%	75%	51%
ANNUAL GROWTH RATE		2%	1%	4%	3%	1%	1%	1%

Table 5 shows the estimated revenue, year-over-year growth, and annual growth rate between the years 2020 to 2027. The annual growth rate is calculated with the equation

(1) $\boldsymbol{AGR} = \frac{\boldsymbol{ending\ value}}{\boldsymbol{beginning\ value}} - \mathbf{1}$ (Hayes, 2022).

Two other calculations average annual growth rate (AAGR) and compound annual growth rate (CARG) are calculated based on the data available from Table 6

(2) $AAGR = \frac{GR_A + GR_B + \cdots + GR_n}{N}$, where GR_A = Growth rate in period A, GR_B = Growth rate in period B, GR_n = Growth rate in period n and N = Number of payments. Based on the data AAGR is

$$AAGR = \frac{0\% + 0\% + 7\% + 26\% + 43\% + 60\% + 72\%}{7} = 29{,}67\ \% \approx 30\ \%$$

(3) $CARG = \left(\left(\frac{EV}{BV}\right)^{\frac{1}{n}} - 1\right) \times 100$, where EV = Ending value, BV = Beginning value and n = number of years (Fernando, 2023; Hayes, 2022). Based on the data CARG is

$$CARG = \left(\left(\frac{215}{0{,}3}\right)^{\frac{1}{7}} - 1\right) \times 100\ \% = 155{,}80\ \% \approx 156\ \%$$

The estimated AAGR of xApps is 30 percent while the estimated CARG is 156 percent.

Table 5. Illustration of global xApp market growth (Adapted with permission from Crawshaw, 2022a)

	YEAR							
REGION	2020	2021	2022	2023	2024	2025	2026	2027
ASIA AND OCEANIA	151 4%	159 6%	171 7%	214 26%	268 25%	324 21%	374 16%	420 12%
LATIN AMERICA AND THE CARIBBEAN	17 5%	18 6%	20 10%	26 34%	35 35%	42 19%	48 14%	53 11%
EUROPE, THE MIDDLE EAST AND AFRICA	69 1%	72 5%	77 8%	103 33%	134 30%	156 17%	174 12%	189 8%
NORTH AMERICA	50 4%	50 0%	53 7%	68 27%	87 28%	102 17%	115 12%	125 9%
TOTAL	286 3%	299 4%	321 7%	411 28%	524 27%	624 19%	712 14%	787 11%

3.2.2 Case company's most recent research

The case company has recently researched its idea of an open ecosystem with third-party feedback from potential customer companies. The case companies' idea was introduced for the research group benchmarking it with rival companies. In this research case, the company's idea was rated on a scale from 1 to 5 (Table 7).

Table 6. Research groups previous rating of the xApp idea

FEATURE	RATING FROM SCALE 1 TO 5
EXPERIENCE WITH THE SANDBOX	3
TEST ENVIRONMENT OPERATIONALITY	4
ABILITY TO EXECUTE ALL THE REQUIRED TESTS/TASKS	2
SANDBOX AND USEFULNESS OF THE APIS/PLATFORM	1
SUPPORT/GUIDANCE FROM CASE COMPANY TEAM	2
DOCUMENTATION QUALITY	1

The case company's most recent findings act as a guideline for further development. Nevertheless, it should be emphasized that earlier research was conducted with a small number of customers focusing solely on the trial sandbox. The idea was introduced to the 3rd party to understand the production environment and test features before going to production.

Customers' responses revealed a sense of disappointment with the sandbox's quality, as they described their experience as

> *"Sandbox seems to be still quite unfinished. However, it is a major improvement compared to the previous version. The usage of standard tooling is a massive improvement."*

Consumers ranked the utility of the offered platform and APIs as low, yet they still believed the concept was good when expressing their thoughts

> "*Good idea and possibly a good product. … unstable, our sandbox was deleted without mentioning and not well documented. Just provide every user its sandbox version and back up the state. Short downtimes are no problem, but deleting the content is."*

Consumers completed the needed activities but were unsatisfied because

> *"… it took too much effort and time on our side."*

Despite past difficulties with accessing the ecosystem's online interface, the customer ranked the test environment as the most successful of all parameters evaluated

> *"… The command line tools worked, which is the most important part."*

The documentation quality was rated low and appears to fall short of user expectations, which is a result of the selected documentation style and the document's content

> *"The documentation was the main issue. It might have been a good choice not to create the technical documentation with Office Suite … The yaml generation was almost impossible from the PDF, because of formatting issues … The documentation was quite verbose without covering some relevant topics."*

The quality of the documentation might have affected the need to ask for help from the case company since the customer stated that

> *"It would have been good if less support would have been needed."*

Based on these comments and ratings, it is apparent that the most significant faults concern the platform's user experience and documentation.

3.3 Data collection

The data for this thesis is going to be collected from a survey and interviews. The thesis is going to mix the two methods by generating a quantitative survey and conducting qualitative interviews based on the survey findings. Data for the thesis is gathered from the case company's internal users and external potential users.

3.3.1 Data collection of the survey

The survey is going to focus on the details of the use of specific technologies and provide information regarding the concept of an open ecosystem, the present user experience, and promotional initiatives. The survey which can be found in Annex 5 consists of a total of 96 questions. There are 8 basic questions, 27 questions related to documentation, and 2 questions related to advertising channels.

Depending on previous experience, users answer 15 questions targeted to users who do not know or have not used xApp technology before or questions that are related to the use of xApp technology. The two choices that are related to xApp use are either testing or developing. Those who choose testing will have 19 questions targeted to xApp testing and those who choose developing will have 20 questions targeted xApp development. Users have a choice to move past the questions related to documentation. This means users will at minimum answer 25 questions and at maximum 57 questions.

The survey contains a shorter version of the user experience questionnaire (UEQ-S) which Schrepp et al. (2017, pp. 104–106) created to analyze the user experience. The UEQ shorter version contains 8 questions and is used in scenarios where answering needs to be fast and the full 26-question version might take too long time to answer. Since the survey contains multiple questions, the researcher opted for the shorter version of the UEQ since it still can predict user experience quite accurately.

The survey was completed by 25 people in total: 6 internals, 10 externals, and 9 students. Figure 12 depicts the many positions that people may hold if they are currently employed. Most of the participants were working at the time of the survey (25 out of 27). There are many different types of IT practitioners, but the most prevalent one is software developers.

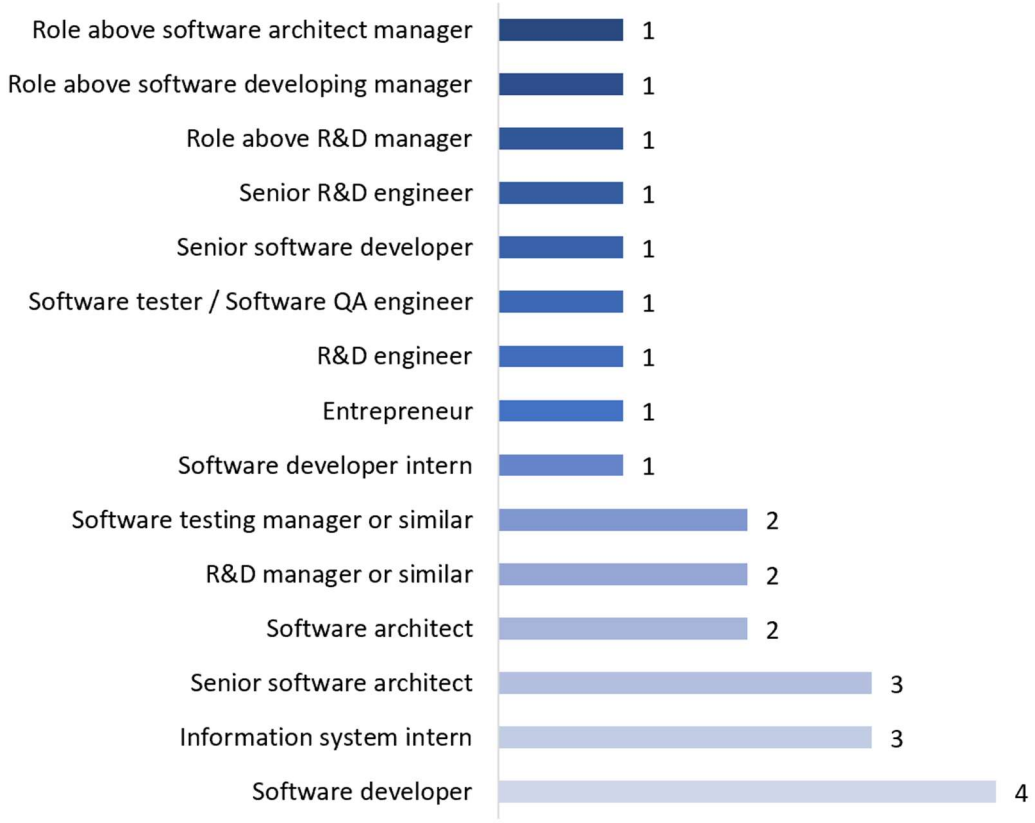


Figure 15. Distribution of survey participants work roles

Most of the participants are experienced in the IT field. Participants in the survey have an average of more than 11 years of experience in the IT business (Figure 13). Students are concluded in the same distribution in figure 13 as every participant.

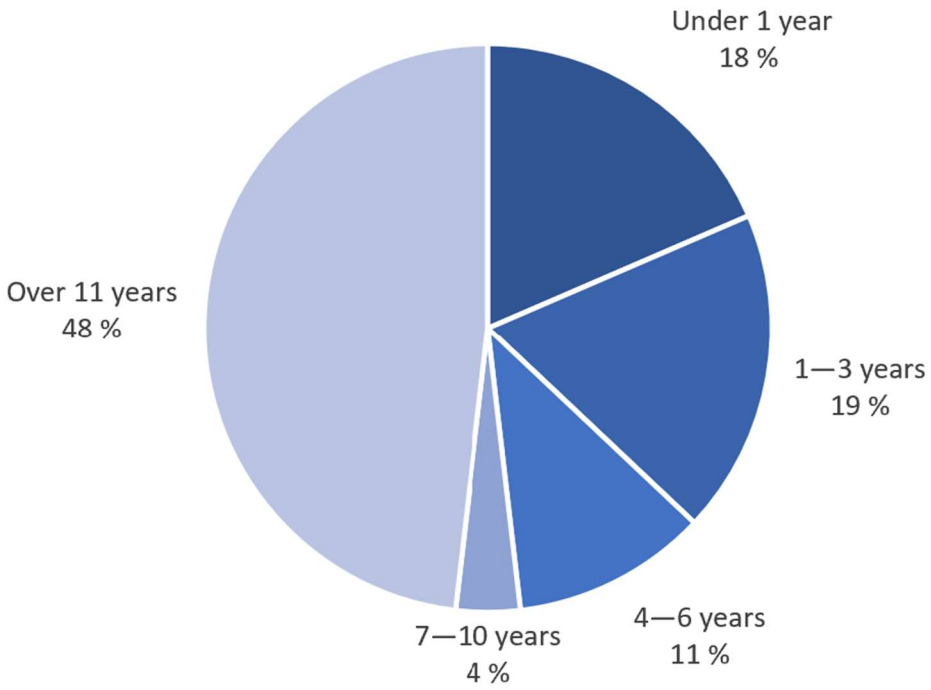


Figure 16. Distribution of survey participants years in the industry

Figure 14 shows the difference between different student statuses. 78 percent of the students are studying for a bachelor's degree and 22 percent are studying for a master's degree.

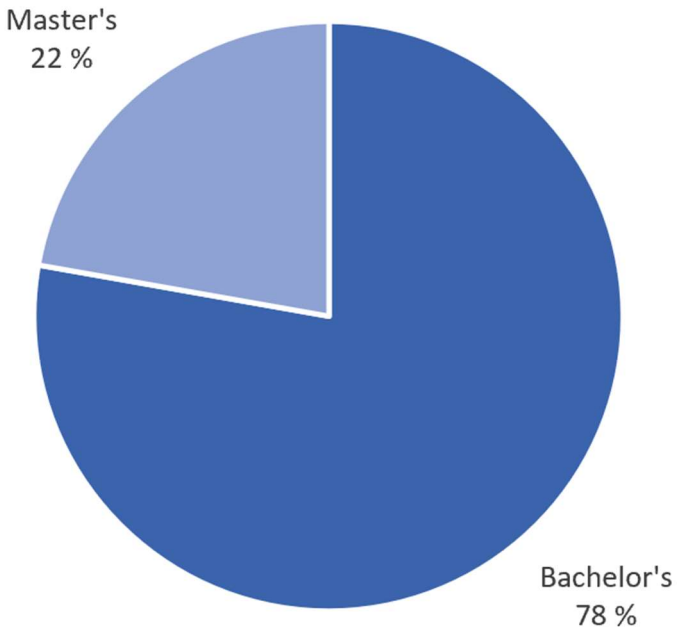


Figure 17. Illustration of survey participants student status

Figure 15 illustrates the various industries represented among survey participants. The most common sectors to work in are telecommunication (40 %) and information and communication (27 %). Other industries that participants have mentioned are manufacturing (10 %), infrastructure (5 %), transportation (5 %), information technology (3 %), administration (3 %), real estate (3 %), law and support services (2 %) and wholesale and retail (2 %).

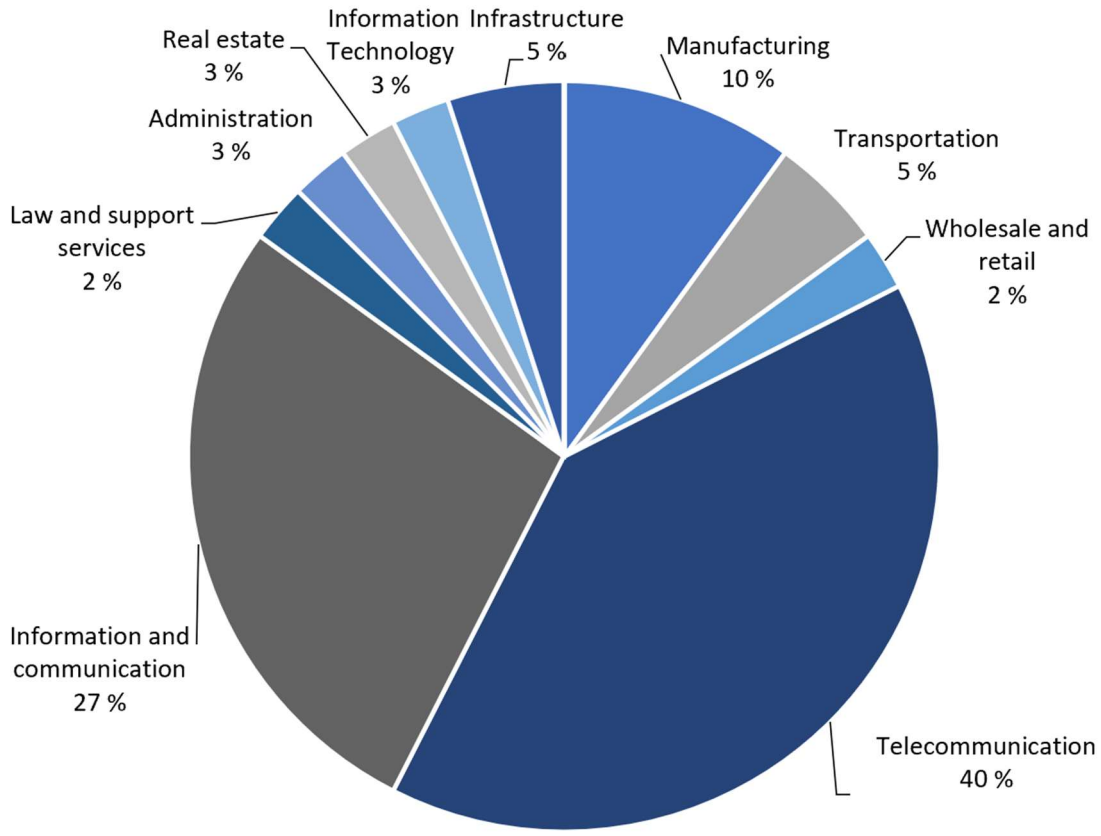


Figure 18. Distribution of survey participants work industries

Survey participants also have a variety of knowledge of different domains and specific technologies (Figure 16 and 17). The five most common domains where participants work are cloud, functional testing, telecommunication and continuous integration/delivery, and IT administration/operations. None of the students have knowledge of telecommunications or radio technology that is related to the xApp technology that this survey is researching. Nonetheless, students have knowledge of

different databases, testing, and embedded systems and provide useful insight into the research with their vision since they are new to radio technologies.

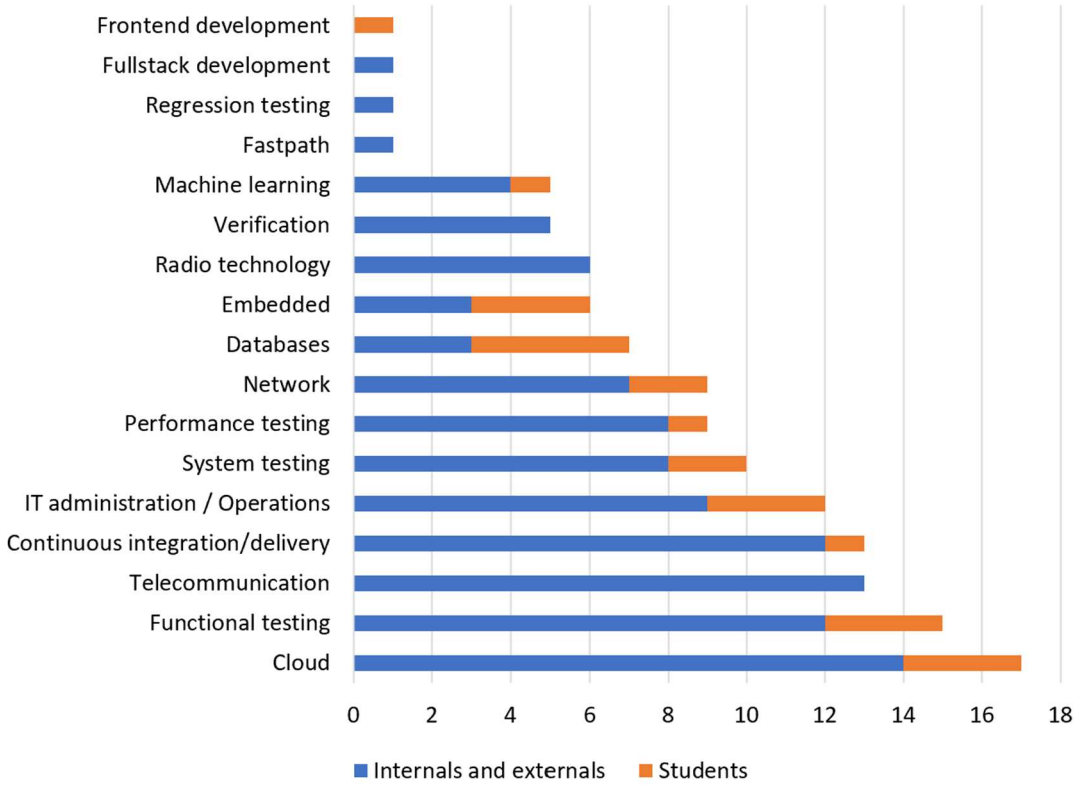


Figure 19. Distribution of domains where survey participants work

There are some differences in knowledge between students and people who work in IT but who are not studying (Figure 17). Students are more familiar with agile processes than people who are not studying. On the other hand, students do not have knowledge about Kubernetes or OpenStack technologies.

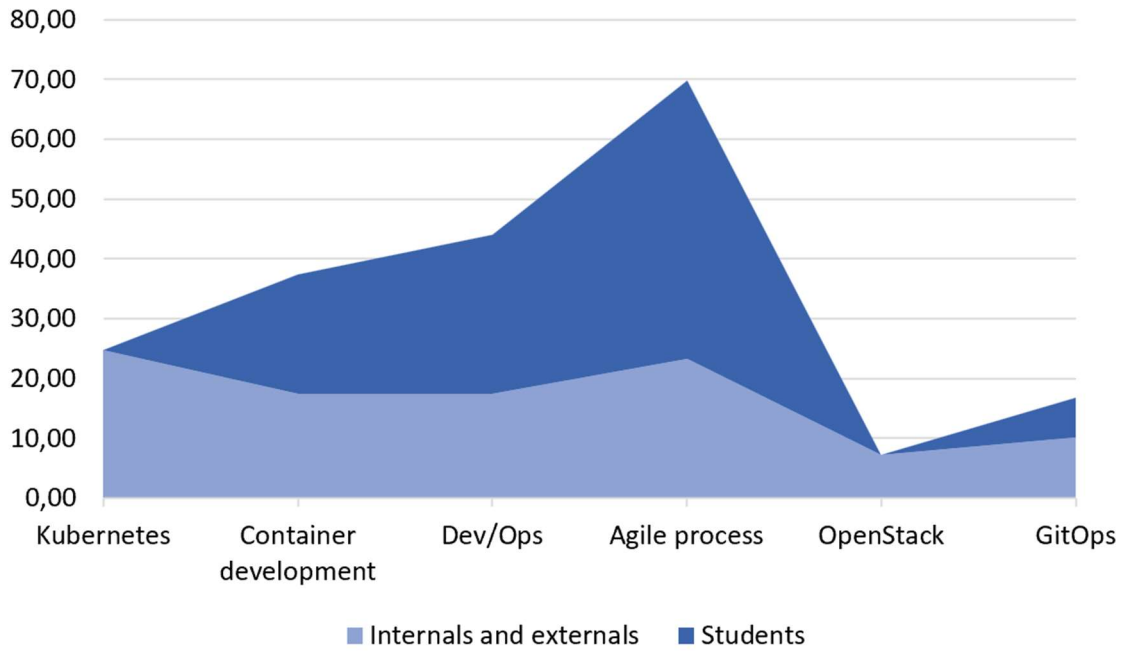


Figure 20. Distribution of knowledge of specific technologies among survey participants

3.3.2 Data collection of the interviews

The interviews are conducted in a manner where the users respond to five short questions which can be found in Annex 6 aim to determine whether the survey findings are consistent with the interview results. The interview questions are likely to provide valuable insights into the challenges and opportunities associated with open API platforms. The questions can help gain valuable insights into the target audience's needs and preferences, which can enhance product development, marketing, and communication strategies.

- Question 1 seeks to understand where users typically look for information related to new technologies and opportunities. This information helps to identify the most effective advertising channels for marketing open API platforms to the target audience.
- Question 2 seeks to identify the critical factor that users consider when deciding to use an open application platform. This information can help understand the

key drivers of the adoption and use of open API platforms among the target audience.

- Question 3 seeks to understand where and how users measure end-customer satisfaction. This information can help to identify the most effective ways to measure and improve customer satisfaction with open API platforms.
- Question 4 seeks to obtain an overall score from users regarding the open-ecosystem idea and the factors they consider in their assessment. This information can help to identify the strengths and weaknesses of the open-ecosystem idea and develop strategies to address any challenges.
- Question 5 seeks to understand the other competing solutions that users have used before and their advantages. This information can help to identify potential areas for improvement in the open API platform and develop strategies to address any gaps or weaknesses.
- Question 6 seeks to obtain feedback on the characteristics of a good partner that assists with developer/testing efforts and the characteristics of a bad partner. This information can help identify what the target audience values in a partner and tailor case company partnerships accordingly.

The interviewees are selected from internal employees of the case company and from the external partner companies' employees to get a throughout view of the thesis topic. The internal employees have experience working directly with xApp testing or development or are in projects that relate to xApp development. The external partner companies get an introduction to the xApp developer portal in the survey and are interviewed based on the impression of what they have received. Table 4 showcases the timetable and other relevant information about the interviews.

Table 7. Timetable and details of the interviews

PERSON	DATE	DURATION	LOCATION OF INTERVIEWEE	METHOD	RELATIONSHIP TO THE CASE COMPANY
A	16.03.2023	26 min 1 s	Dallas, US	Online video call	Internal employee
B	20.03.2023	23 min 4 s	Bangalore, IN	Online voice call	Internal employee
C	20.03.2023	21 min 15 s	Bangalore, IN	Online voice call	Internal employee
D	24.03.2023	38 min 50 s	Bucharest, RO	Online video call	Partner organizations employee
E	28.03.2023	22 min 44 s	Munich, DE	Online video call	Previous internal employee
F	28.03.2023	24 min 42 s	Espoo, FI	Online video call	Partner organizations employee
G	31.03.2023	26 min 41 s	Espoo, FI	Online voice call	Partner organizations employee
H	05.04.2023	30 min 32 s	Espoo. FI	Online video call	Internal employee
I	11.04.2023	36 min 8 s	Espoo, FI	Online video call	No previous relationship (student)
J	14.04.2023	32 min 50 s	Espoo, FI	Online voice call	Internal employee
K	20.04.2023	1 h 30 min 54 s	Espoo, FI	Online video call	Partner organizations employee
L	28.04.2023	29 min 23 s	Bangalore, IN	Online video call	Partner organizations employee
M	28.04.2023	29 min 13 s	Dallas, US	Online voice call	Partner organizations employee

Each interviewee is asked the same set of questions while recording and transcribing their responses. The researcher asks clarifying questions based on what the interviewee answers if needed. This enhances the clarity of the responses and ensures what the interviewee has replied. A total of 11 interviewees attended the research from different locations (Figure 18). The number of interviewees started to repeat the same findings which said that the required level of saturation was reached.

Figure 21. Location of interviewees (Map data from OpenStreetMap)

3.4 Reliability and validity

"Knowledge gives us the ability to predict with a reasonable degree of certainty or validity that a certain observed state or pattern of events, data, or information will lead to a new, describable and definable state" (Mason, 1988). Research methods differ according their internal and external validity (Järvinen, 2001, p. 56). It is suggested to test the empirical validity, logical consistency, relative predictive power, and falsifiability of case studies (Järvinen, 2001, p. 62; Mason, 1988).

Yin and Campbell (2003, pp. 33–35) suggest using four tests for case studies to judge the quality and to establish a research. These four tests are construct validity, internal validity, external validity, and reliability. Construct validity ensures the study will be using "correct operational measures to explain the concepts". This tactic occurs in the data collection phase when evidence is searched, and informants review the draft of the study. It is

difficult to interpret the construct validity because the evaluations of the collected data are always subjective.

Yin and Campbell (2003, pp. 33–36) describe internal validity as research tactic that occurs in data analysis phase where explanations, logic models and patterns between factors are being found and made. Internal validity tries to help build a causal relationship between conditions. The risk in internal validity phase is doing wrong interpretations and creating false relationships.

Yin and Campbell (2003, pp. 33–37) define external validity as a research tactic that occurs in research designing phase where theory and replication is studied. External validity tries to build a generalization of the study's findings. The problem with external validity is the possibility to replicate and suggest a generalization based on a single case study.

Yin and Campbell (2003, pp. 33–36) characterize reliability as a research tactic that occurs in data collection phase where study protocol and database are being created. Reliability demonstrates the operations that were made to get the results so that the study can be repeated. The risk in reliability is that even though the steps of the study would be repeated the result would not be the same.

An effort to pay special attention to all the previously described details of the validity and reliability is being made through the research process. As an example, the research has been reviewed by senior researchers who have agreed on the outcomes. Furthermore, the key informants have evaluated the manuscript and agreed on the study's findings.

4 Research findings

This chapter to be focusing on the findings. The data collection process and analyses of the collected data have been described earlier in the methodology part. The findings are based on the conducted survey and interviews.

This thesis follows the case study approach proposed by (Hollweck, 2015). First the survey was conducted and analyzed, the interviews were conducted and transcribed, and secondary data was gathered and summarized. The data from the survey was transferred to SAP tool to help in the data analysis. The data gathered from the survey and interviews was cross-checked together with case company representatives.

All the surveys material and interviews were held in English. Some of the people would have preferred another language since there were sometimes difficulties with some interviewees getting the full meaning of what they wanted to say. As Myers (2019, p. 45) suggests “a social researcher must already speak the same language as the people being studied”. Therefore, some Finnish interviewees were given the opportunity to explain something that was too difficult for them to explain in English in Finnish.

After data collection and analyzing this chapter, is trying to make sense of the underlying concepts and findings. Throughout the process only the key findings from the survey and interviews were selected for this thesis. To make the evaluation easier the most relevant findings are then compared with prior studies.

4.1 Findings form the survey

UEQ is used to predict the user experience of the case company’s current documentation. Short UEQ data analyzing tool where the gathered results are placed indicates that pragmatic quality, hedonic quality, and overall condition of the current documentation are all bad (Table 8 and figure 22). Pragmatic qualities are “qualities that relate to the tasks or goals the user aims to reach when using the product” (Schrepp et al., 2017, p.

104). Hedonic qualities are “quality aspects, in other words, they do not relate to tasks and goals, but describe aspects related to pleasure or fun while using the product” (Schrepp et al., 2017, p. 104).

Table 8. Measures of documentation quality

SCALE	MEAN	STANDARD DEVIATION	N	CONFIDENCE	CONFIDENCE INTERVAL		COMPARISON TO BENCHMARK	INTERPRETATION	CRONBACH'S ALPHA-COEFFICIENT
PRAGMATIC QUALITY	0,200	1,311	10	0,813	-0,613	1,013	Bad	In the range of the 25% worst results	0,90
HEDONIC QUALITY	-0,175	1,155	10	0,716	-0,891	0,541	Bad	In the range of the 25% worst results	0,88
OVERALL	0,013	1,076	10	0,667	-0,654	0,679	Bad	In the range of the 25% worst results	.

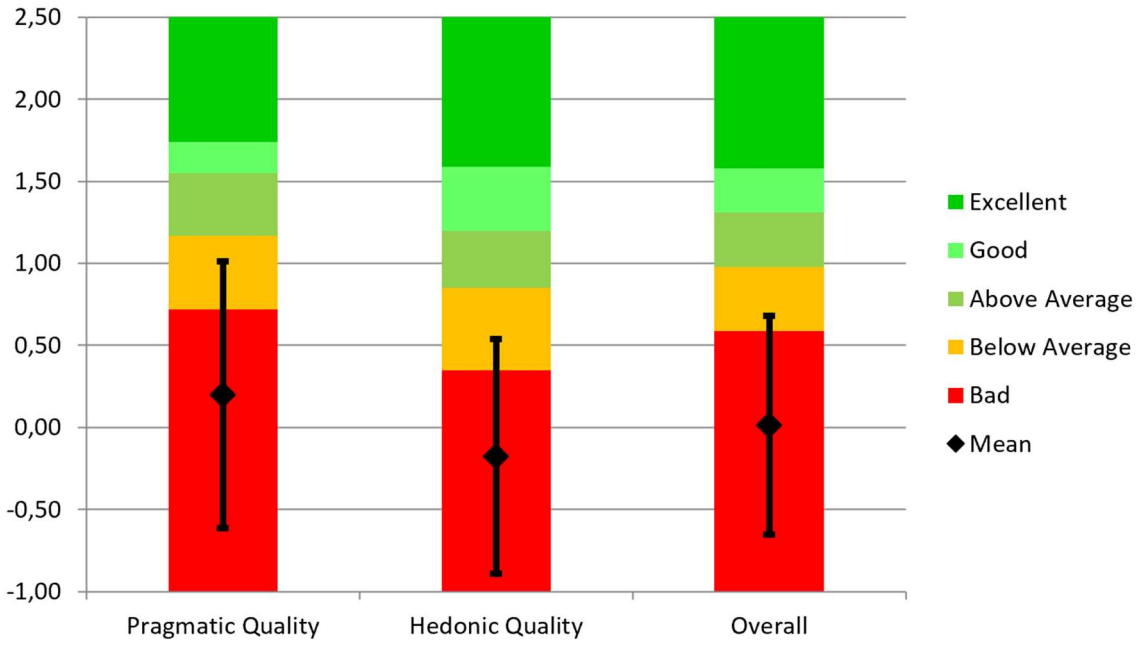


Figure 22. The pragmatic, hedonic and overall quality of the documentation

UEQ “defines areas of improvement and allows a rough measurement on higher level meta-dimensions” (Schrepp et al., 2017, p. 107). The Cronbach-Alpha Coefficient suggests that the hedonic quality and pragmatic quality are considered as sufficiently consistent while the alpha value in both quality factors is bigger than 0.7 (Figure 23). Figure 23 shows the mean values for pragmatic quality with blue color and hedonic quality with yellow color. Figure 23 shows that there is an opportunity for improvement in hedonic quality since three of the four hedonic quality factors are negative.

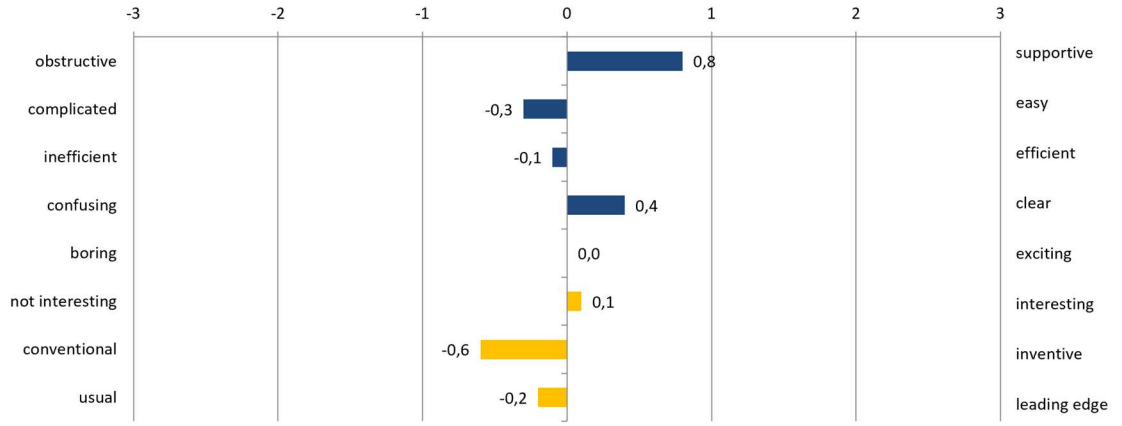


Figure 23. Illustration of mean values for each pragmatic and hedonic quality criterion of documentation

The survey suggests that there is potential for improvement as users view the documentation as extremely conventional and very usual. The documentation is viewed as nothing out of the ordinary and it does not create interest. This may be related to people's typical notions of documentation. Documentation is viewed as something that must be utilized only when necessary and contains extremely precise information that may tire some people.

The current documentation is supportive and clear to use, which is good. It is, however, concerning that users view the documentation as very complicated and inefficient to use. These are the key concerns that the case company should focus on improving the documentation since the usability of the documentation might be an important factor than the usual and conventional feel of the documentation.

Figure 24 represents the different documentation metrics in a similar manner as in the UEQ model. It can be observed that documentation is not attractive, self-descriptive, satisfying, and understandable enough to test or develop an xApp since they are all negative attributes of the mean analysis. Documentation is also lacking API use cases since the users have voted that high on the survey.

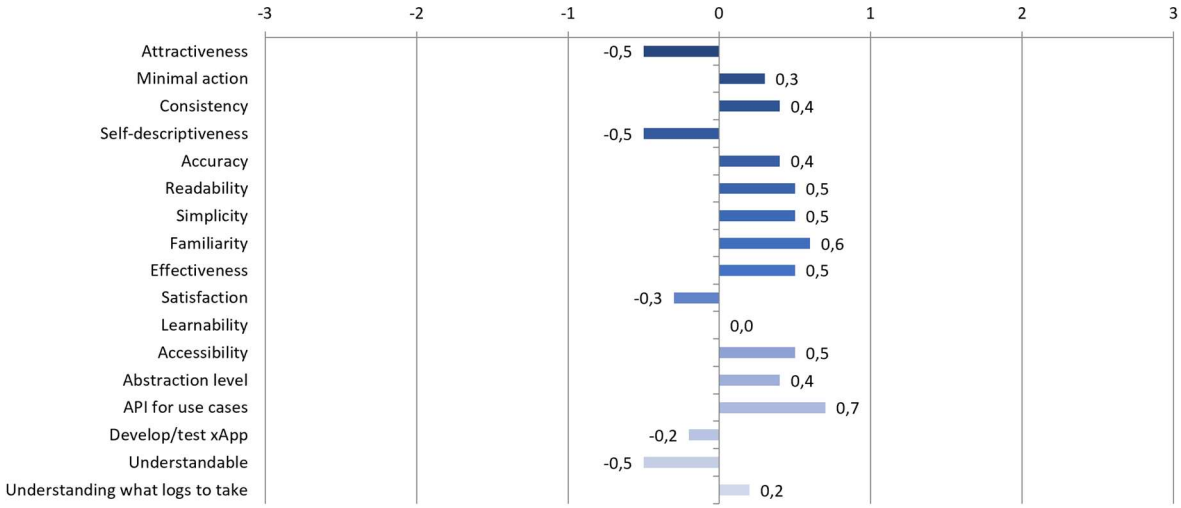


Figure 24. Illustration of mean values for case company's criterion for documentation

Learnability is neutral, which is alarming since the documentation should provide help to get started. If users do not find the documentation learnable then it might result in difficulties understanding what to do regarding creating or testing an xApp. On the other hand, this might be a result of the difficulty understanding the xApp framework since users think that the documentation is not understandable or the complexity of the radio technologies. The documentation does not describe the application use case details to such a level that the functionality is understandable.

Many positive metrics can be concluded from the status of the documentation. Familiarity, simplicity, readability, effectiveness, and accessibility are all highly important criteria for successful documentation since they affect the use of documentation. Criterions such as minimal action, consistency, accuracy, readability, simplicity, familiarity, effectiveness, accessibility, abstraction level, and understanding what logs to take to problem investigation have positive values in the mean analysis.

The survey visualizes the idea and introduces the xApp technology. Figure 25 illustrates five elements that users who have no experience with xApps have answered. The first impression of the idea is positive and meets users' expectations. However, the motivation to create or test an xApp, the overall design, and the helpfulness of the idea

seems to be quite neutral or in the case of the design negative. This is not a good sign since the participants tend to think it is an innovative idea but not for them.

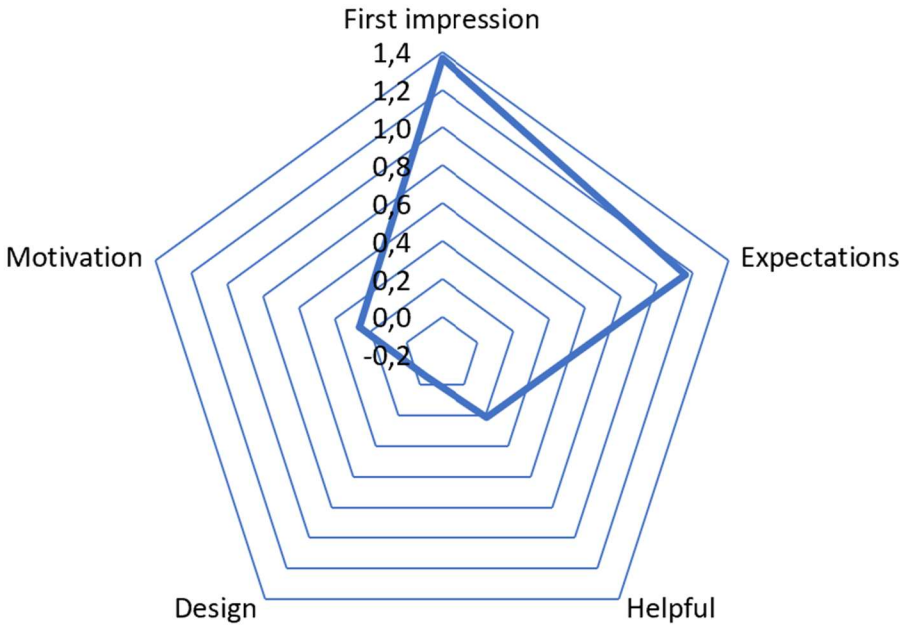


Figure 25. Radar illustration of external participants opinion about the xApp idea

Figure 26 illustrates the mean values of different metrics according to actual xApp creators and testers. Users think that third-party developers should be more on the expert level of the needed skills to be able to implement an xApp based on the documentation. Actual xApp creators and testers think that documentation might be too difficult for third-party developers to understand. Minimal action and self-descriptiveness have low scores. This means it takes more time and effort to learn to use, deploy and test an xApp than usual. Users also feel that there is not enough help in the form of information, documentation, and tooltips to test or implement an xApp on the platform.

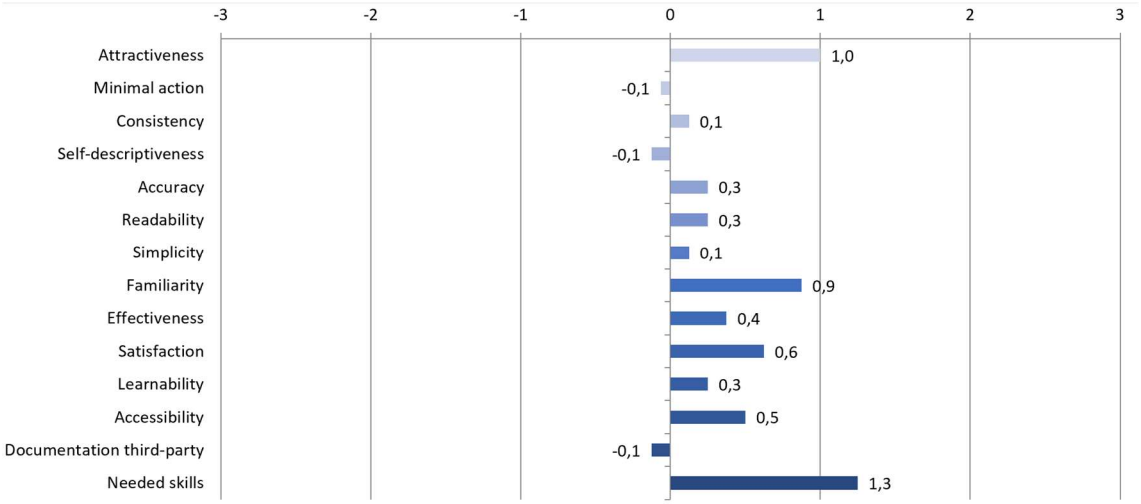


Figure 26. Illustration of mean values for each criterion of xApp according to case company results

Social channels are considered to provide an important support network and help when participants are searching for information regarding the new platform. According to the survey, the most important social channel is YouTube (Figure 27). The second social channel that students especially consider important is GitHub. Other highly valued social channels are Stack Overflow and LinkedIn.

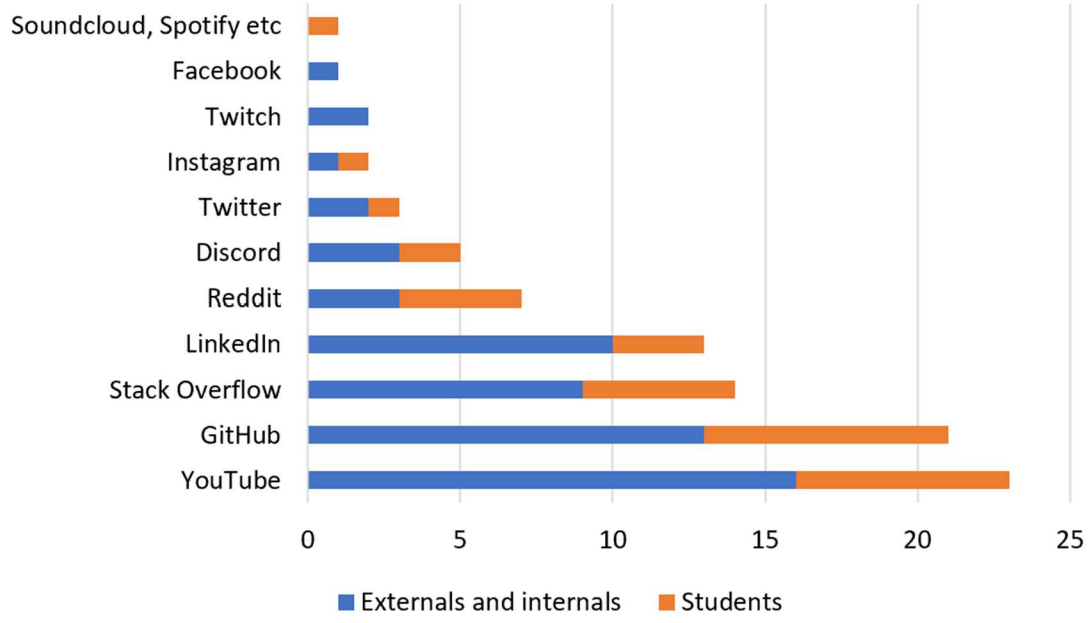


Figure 27. Most preferred social channels among survey participants

Figure 28 represents content that benchmarked platform providers provide that users value the most. Code samples, tutorials, courses, cheat sheets, sandbox, or test environments are considered the five most valued content by users. Use cases and articles are also considered interesting among the survey participants.

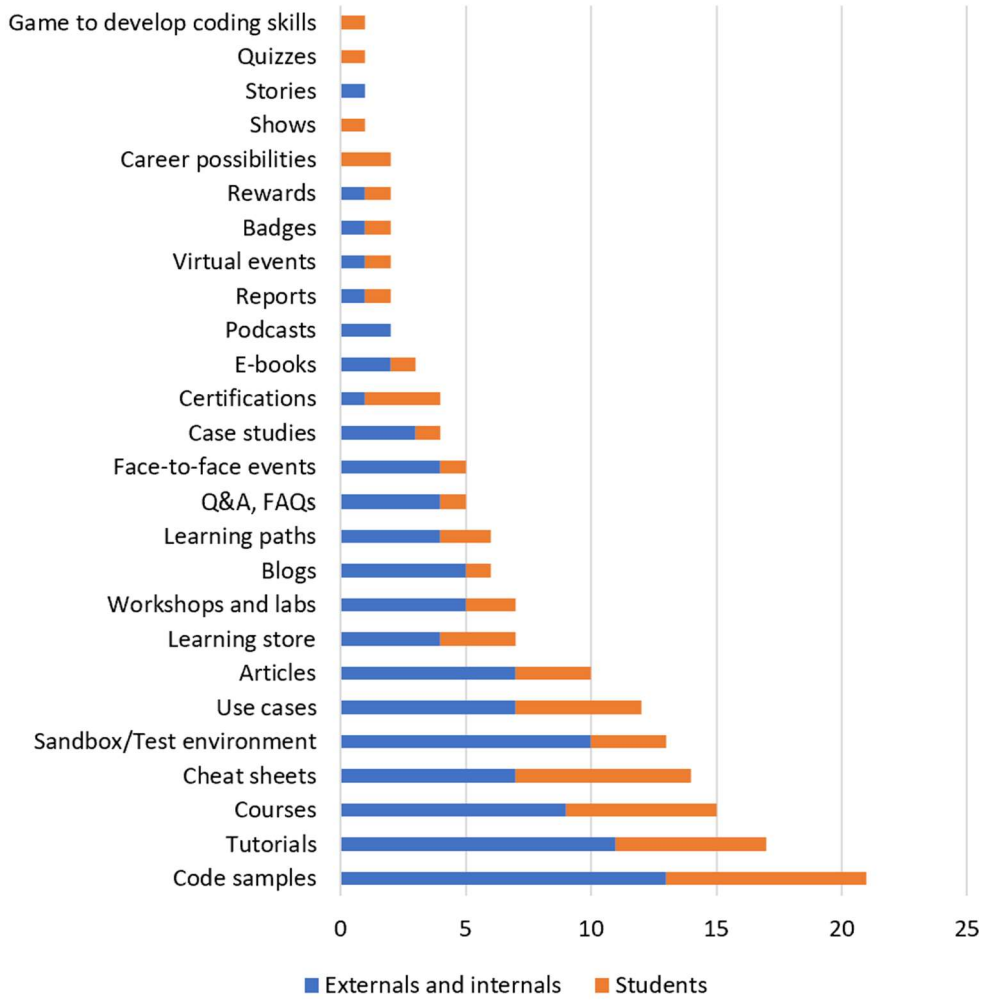


Figure 28. Most interesting content among survey participants

Figures 29 and 30 display the preferred software and testing tools. Over half of developers want to run the application instantly and use a debugger. Unit tests are performed often, and IDE is one of the most wanted tools for xApp development. Syslog, ifconfig/ip, tcpdump/Wireshark, ChatGPT, and IDE are the most used testing tools among survey participants.

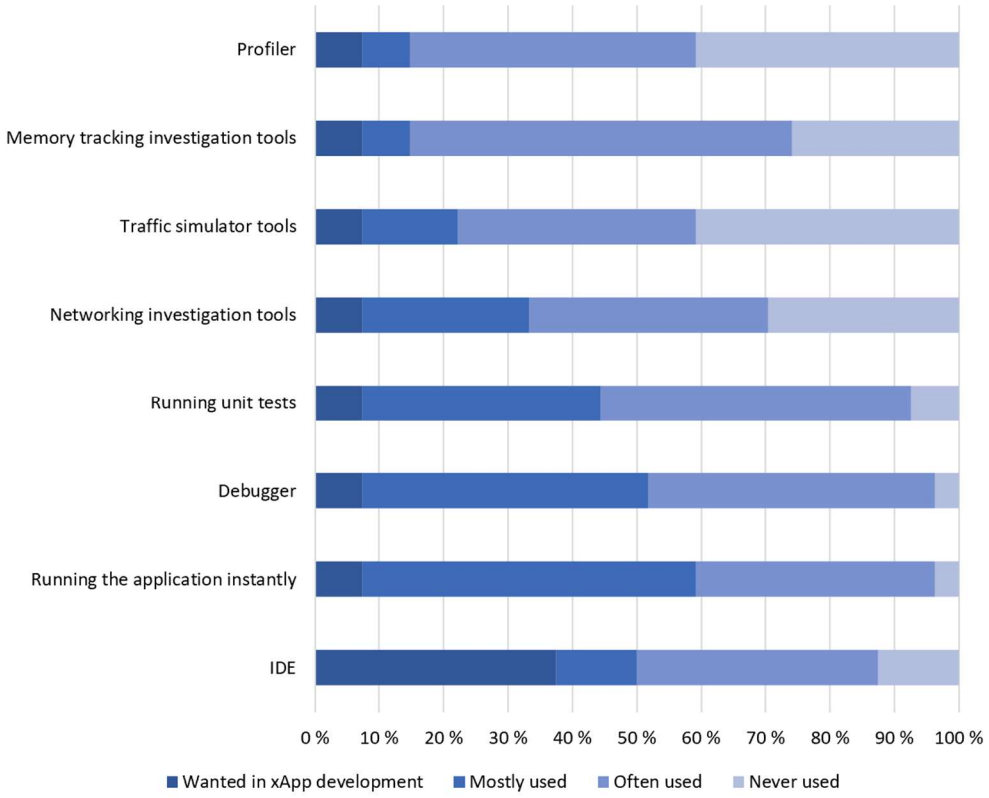


Figure 29. Preferred software tools among survey participants

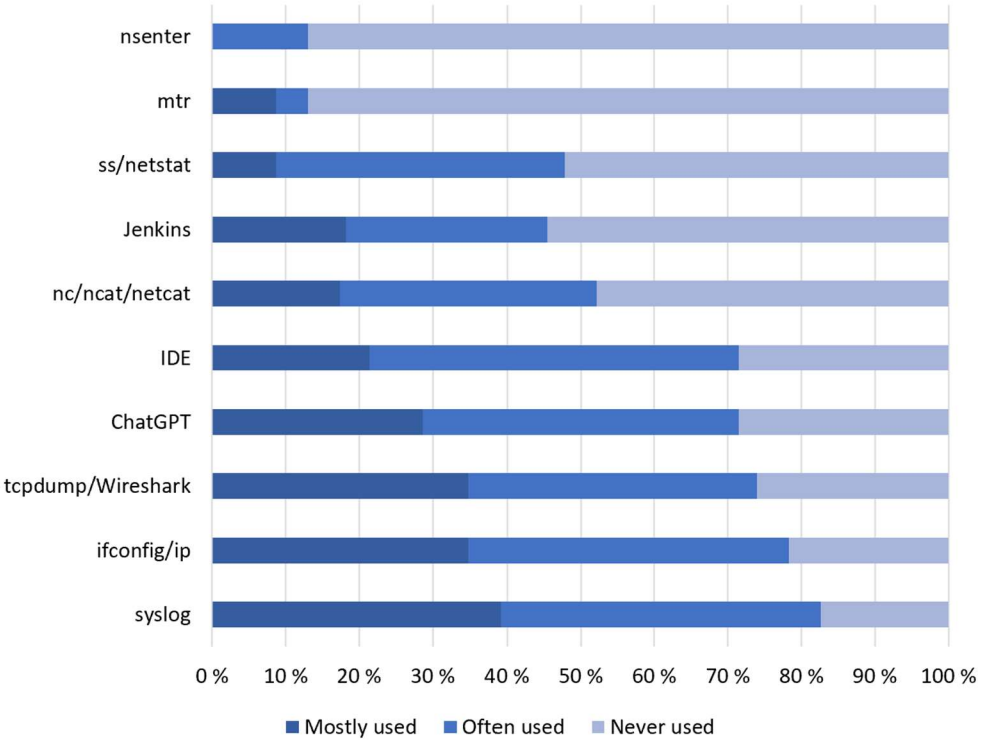


Figure 30. Preferred testing tools among survey participants

Tables 9 and 10 illustrate the correlations and the relationship between testing and developer experience for first impression, expectations, helpfulness, design, and motivation of externals and students. Externals have only one high correlation which is between developer experience and expectations. This high negative linear correlation exists because people who have a higher amount of developer experience do not think that the xApp platform meets their expectations while people who have a lower amount of developer experience do think it meets their expectations.

Table 9. Correlation of testing and developer experience of externals and students

			PEARSON CORRELATION COEFFICIENTS				
xApp idea			FIRST IMPRESSION	EXPECTATIONS	HELPFULNESS	DESIGN	MOTIVATION
TESTING EXPERIENCE	EXTERNALS (N = 10)	r	-0.15	-0.18	0.44	0.42	0.32
		p-value	.7224	.6690	.2718	.3026	.4364
	STUDENTS (N = 9)	r	0.09	-0.40	0.42	0.05	-0.29
		p-value	.8172	.2924	.2664	.8973	.4561
DEVELOPER EXPERIENCE	EXTERNALS (N = 10)	r	-0.48	-0.70	-0.10	-0.30	-0.32
		p-value	.2286	.0556	.8190	.4799	.4386
	STUDENTS (N = 9)	r	0.20	-0.10	0.68	0.23	0.03
		p-value	.6110	.7904	.0459	.5479	.9465

Table 10. Relationship between testing and developer experience correlations for externals and students

		PEARSON CORRELATION COEFFICIENTS RELATIONSHIP				
xApp idea		FIRST IMPRESSION	EXPECTATIONS	HELPFULNESS	DESIGN	MOTIVATION
TESTING EXPERIENCE	EXTERNALS	VERY LOW NEGATIVE LINEAR	VERY LOW NEGATIVE LINEAR	MODERATE POSITIVE LINEAR	MODERATE POSITIVE LINEAR	LOW POSITIVE LINEAR
	STUDENTS	VERY LOW POSITIVE LINEAR	MODERATE NEGATIVE LINEAR	MODERATE POSITIVE LINEAR	NON-EXISTING POSITIVE	LOW NEGATIVE LINEAR
DEVELOPER EXPERIENCE	EXTERNALS	MODERATE NEGATIVE LINEAR	HIGH NEGATIVE LINEAR	VERY LOW NEGATIVE LINEAR	LOW NEGATIVE LINEAR	LOW NEGATIVE LINEAR
	STUDENTS	LOW POSITIVE LINEAR	VERY LOW NEGATIVE LINEAR	HIGH POSITIVE LINEAR	LOW POSITIVE LINEAR	NON-EXISTING POSITIVE

This might make marketing for highly professional developers more difficult if they already have high expectations and feel suspicious about the initial idea. The business might become risky since the platform owner needs to put a lot of effort into reassuring

the experts to get more users. On the other hand, experts might not have thought about this type of idea before and feel overwhelmed by it. However, it is not good that they feel that the idea does not meet their criteria of expectations.

There exist also three moderate correlations for externals. There are moderate positive linear correlations between testing experience and ideas helpfulness and design. Externals who have more testing experience think that the idea is going to help them, and that the solution is designed for them. Externals also get a bit more motivation to test the idea while they have more experience.

What stands out from student correlations is the high positive linear correlation that is between developer experience and helpfulness. The more students have developer experience the more they think this solution is going to be helpful for them. On the other hand, students who do not have great developer skills think that this solution is not going to help them. This might be a problem while trying to get more students invested to learn the xApp technology if they do not feel familiar enough with coding and do not feel confident enough to understand how to implement xApps.

Tables 11 and 12 illustrate the features and the correlations between them. Design and helpfulness have a very high correlation. If the user thinks the xApp development environment will provide them lots of help the more certain it would be they would agree with that it was designed for them. First impression and design also have a highly positive correlation. If a user has a better first impression the more likely they feel like the solution is designed for them. Perceived helpfulness and design moderately affect motivation to try implementing or testing xApps.

Table 11. Correlation between first impression, expectations, helpfulness, design and motivation

		PEARSON CORRELATION COEFFICIENTS				
		FIRST IMPRESSION	EXPECTATIONS	HELPFULNESS	DESIGN	MOTIVATION
FIRST IMPRESSION	r	1.0	0.32	0.43	0.65	-0.12
	p-value		.3683	.2151	.0437	.7465
EXPECTATIONS	r	0.32	1.0	0.30	0.36	0.19
	p-value	.3683		.3992	.3054	.6036
HELPFULNESS	r	0.43	0.30	1.0	0.90	0.55
	p-value	.2151	.3992		.0004	.1001
DESIGN	r	0.65	0.36	0.90	1.0	0.50
	p-value	.0437	.3054	.0004		.1394
MOTIVATION	r	-0.12	0.19	0.55	0.50	1.0
	p-value	.7465	.6036	.1001	.1394	

Table 12. Relationship between correlations of first impression, expectations, helpfulness, design and motivation

	PEARSON CORRELATION COEFFICIENTS RELATIONSHIP				
	FIRST IMPRESSION	EXPECTATIONS	HELPFULNESS	DESIGN	MOTIVATION
FIRST IMPRESSION	-	LOW POSITIVE LINEAR	MODERATE POSITIVE LINEAR	HIGH POSITIVE LINEAR	VERY LOW NEGATIVE LINEAR
EXPECTATIONS	LOW POSITIVE LINEAR	-	LOW POSITIVE LINEAR	LOW POSITIVE LINEAR	VERY LOW POSITIVE LINEAR
HELPFULNESS	MODERATE POSITIVE LINEAR	LOW POSITIVE LINEAR	-	VERY HIGH POSITIVE LINEAR	MODERATE POSITIVE LINEAR
DESIGN	HIGH POSITIVE LINEAR	LOW POSITIVE LINEAR	VERY HIGH POSITIVE LINEAR	-	MODERATE POSITIVE LINEAR
MOTIVATION	VERY LOW NEGATIVE LINEAR	VERY LOW POSITIVE LINEAR	MODERATE POSITIVE LINEAR	MODERATE POSITIVE LINEAR	-

Tables 13 and 14 illustrate the correlations and the relationship between testing and developer experience for first impression, expectations, helpfulness, design, and motivation of actual xApp developers and testers. There cannot be many conclusions based on the developer experience and factors since the correlations are low or very low. On the other hand, there are several high and very high correlations between tester experience and the factors.

According to the results xApp technology is difficult to master as experienced xApp testers think that to test an xApp the needed skills need to be high, and the tester needs to be experienced enough to do it. In other words, experts think that you need to be an expert to test xApps. Experienced xApp developers also think moderately that to develop an xApp the developer needs to have expert level skills.

Table 13. Correlation of testing and developer experience of xApp users

PEARSON CORRELATION COEFFICIENTS															
FACTORS		ATTRACTIVENESS	MINIMAL ACTION	CONSISTENCY	SELF-DESCRIPTIVENESS	ACCURACY	READABILITY	SIMPLICITY	FAMILIARITY	EFFECTIVENESS	SATISFACTION	LEARNABILITY	ACCESSIBILITY	DOCUMENTATION THIRD-PARTY	NEEDED SKILLS
TESTER EXPERIENCE (N = 3)	r	-1.00	0.50	-0.76	-0.94	-0.50	0.50	0.50	0.87	-0.94	-0.87	-0.50	-1.00	-0.50	0.87
	p-value	<.0001	.6667	.4544	.2123	.6667	.6667	.3333	.3333	.2123	.3333	.6667	<.0001	.6667	.3333
DEVELOPER EXPERIENCE (N = 5)	r	-0.15	0.00 & 0.16	0.17	0.00	-0.30	-0.38	-0.20	0.00	0.00	-0.16	0.00	-0.15	-0.43	0.46
	p-value	.8105	1.0 & .7911	.7864	1.0	.6186	.5240	.7419	1.0	1.0	.7911	1.0	.8105	.4664	.4397

Table 14. Relationship between testing and developer experience correlations for xApp users

PEARSON CORRELATION COEFFICIENTS RELATIONSHIP															
FACTORS	ATTRACTIVENESS	MINIMAL ACTION	CONSISTENCY	SELF-DESCRIPTIVENESS	ACCURACY	READABILITY	SIMPLICITY	FAMILIARITY	EFFECTIVENESS	SATISFACTION	LEARNABILITY	ACCESSIBILITY	DOCUMENTATION THIRD-PARTY	NEEDED SKILLS
TESTER EXPERIENCE (N = 3)	VERY HIGH NEGATIVE LINEAR	MODERATE POSITIVE LINEAR	HIGH NEGATIVE LINEAR	VERY HIGH NEGATIVE LINEAR	MODERATE NEGATIVE LINEAR	MODERATE POSITIVE LINEAR	MODERATE POSITIVE LINEAR	VERY HIGH POSITIVE LINEAR	VERY HIGH NEGATIVE LINEAR	VERY HIGH NEGATIVE LINEAR	MODERATE NEGATIVE LINEAR	VERY HIGH NEGATIVE LINEAR	MODERATE NEGATIVE LINEAR	VERY HIGH POSITIVE LINEAR
DEVELOPER EXPERIENCE (N = 5)	VERY LOW NEGATIVE LINEAR	VERY LOW POSITIVE LINEAR	VERY LOW POSITIVE LINEAR	NON-EXISTING	LOW NEGATIVE LINEAR	LOW NEGATIVE LINEAR	LOW NEGATIVE LINEAR	NON-EXISTING	NON-EXISTING	VERY LOW NEGATIVE LINEAR	NON-EXISTING	VERY LOW NEGATIVE LINEAR	MODERATE NEGATIVE LINEAR	MODERATE POSITIVE LINEAR

Experienced testers are also dissatisfied with their xApp testing experience and they disagree with the fact that there would be enough information, documentation, and tool tips available to test an xApp on the platform. Advanced testers think that xApps are not helping to achieve their goals with accuracy and relevance and they are not satisfied after working with xApps. From a satisfaction point of view experienced testers think that the xApp is unclear compared to other platforms or programs.

Both experienced testers and developers think moderately that a third-party tester or developer could not be able to test or develop an xApp based on the provided documentation. This is something found in the case company's previous research. Experienced testers however think that the provided xApp platform tools are familiar

with other ones they have been using for testing which gives hope to beginners who are learning xApp testing.

It must be noted that the number of people that have xApp testing experience is three and the amount of people with xApp developer experience is five. This can distort the results and affect why so many high correlations exist between the factors and testing experience. Another reason for many negatively affecting correlations between the tester experience and factors is that testers are more dissatisfied with xApp testing than developers are with developing xApps.

4.2 Findings from the interviews

Users search for information related to new technologies and opportunities across different platforms. The most common channels where interviewees come across information regarding new technologies are social media (17 %) and word of mouth (15 %) (Figure 31). Other good channels which are useful and are mentioned in interviews are research articles (10 %), newspapers (10 %), blogs (10 %), and magazines (10 %). Other mentioned channels include YouTube (8 %), communities (7 %), networking (5 %), television (3 %), podcasts (2 %), and events (2 %).

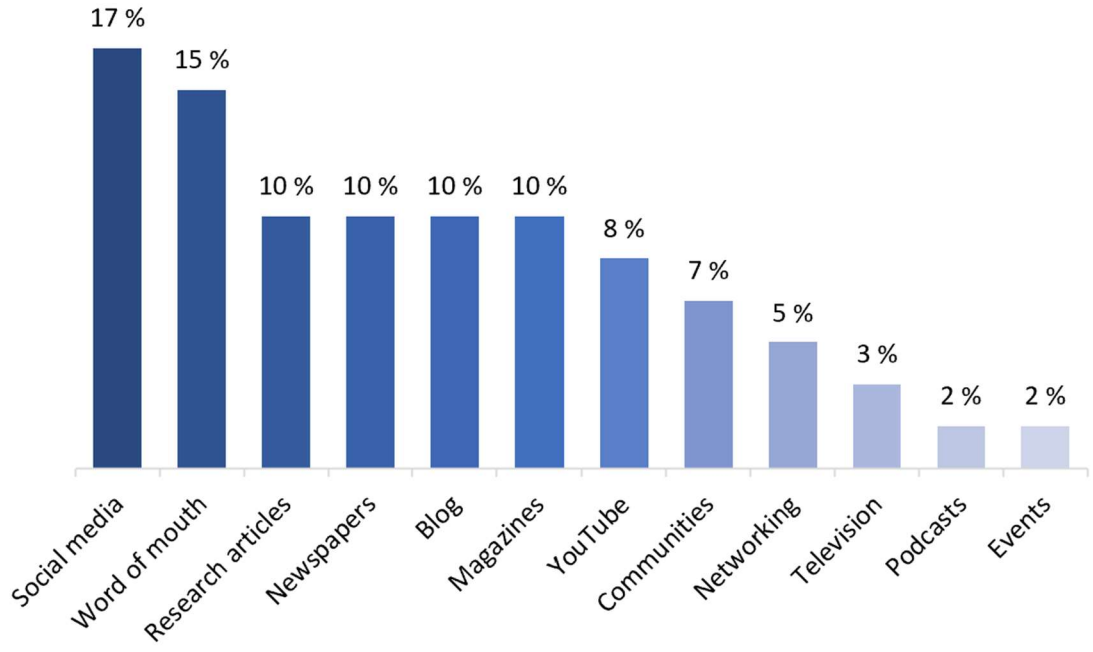


Figure 31. Information channels among interview participants

According to the interviews 1 of 13 would have preferred physical magazines or newspapers and the rest would prefer online channels. Even word of mouth interaction that has happened usually live has transformed during the pandemic to an online setting between colleagues in a remote or hybrid work mode. Most interviewees want to learn and read about new technologies using user equipment (UE) such as mobile/smartphone, tablet, or computer. Thus, focusing on online digital marketing is essential.

What stood out from the responses is that users tend to stay with the technologies they feel comfortable and familiar with. Interviewees feel that there is hesitation about whether the technology is worthwhile to learn. Interviewee A refers to the adaptation process as

> *"... whenever you go to a new thing, there is a learning curve ... there is this mountain that you have to overcome to learn something new. You are always more comfortable to stay in your comfort zone … I would say it is almost like a*

religious aspect. That is why you are a little bit hesitant to change because you get so comfortable using one set of tools and how your process and how you do things."

To improve this type of hesitation to adapt to learn a new technology helps the technologies later use and reuse in the future. The adaptation depends also on the person's interests and how they evaluate the technology usage based on what other users say and if they are contributing. Interviewee E analyses first do the application platform meets their requirements. After deciding whether the platform does meet interviewee Es' requirements interviewee E starts to analyze what benefits they could gain, ease of use, and simplicity in decision making.

First, the platform needs to be easy to approach. Interviewee F pays special attention to the first impression of the first page of the website and the overall look and feel of the website. If the website feels clumsy and the overall look of the website does not appeal to interviewee F, they will not proceed any further.

Interviewees want to learn as much as it is possible from the platform before committing to it. Interviewee A mentions how-to articles and videos as something that makes it easier to want to absorb and start to use new technology. Interviewee J also investigates the availability of example tools, development and testing resources, and documentation. Interviewee D also wishes for a supportive and empathetic platform provider that has guided training.

Interviewees need to read the documentation and get to know API libraries before committing to signing up for anything. Interviewee G would like to know how much the developer is supported and what is provided before making any decisions. Interviewee G explains what affects their platform use after signing up

"When I start using it of course I learn how things work and if there are any pain points or something that is missing, and it is all about at that point … either the

community around the platform or the support that they provide. Can I report an issue and can I find somehow easy access to people behind the platform?"

While it is easier to get people interested in open-source technologies, people tend to change open-source technologies more often than paid ones. Interviewee H thinks the advantages of open-source platforms are their ability to cater to diverse user needs, avoid hardware limitations, and run in cloud environments. It is something to think about when a company wants to invest in making its resources open to use. Interviewee D describes the qualities open-source technology must have to stay relevant

"… in three months, it is not necessarily obsolete, but people already move to a different tool that can do more or is easier to use. … you simply must be up to date … paid ones are harder to switch from … open source, gives you this ability ditch … because there is no cost involved … you do not have any licenses and you do not have any agreements in place that keep you to use that tool … it is flexible, which is why we like open-source environment."

The work environment also affects the adaptation of an open API. Interviewee B mentions that the decision-making of an open application platform depends on multiple factors. Company policy, licensing which affects certain patented technologies, and team decisions as critical factors while figuring out if developers are willing to contribute to the new technology. Some teams might test out the technology before other teams adapt to it. Cost is also a critical factor when deciding which open source is generating a lot of revenue. Interviewee D describes the most critical factor for them

"Cost is basically the main factor. If you are working for a services provider or if you are doing something internally just to avoid the licensing part people usually prefer an open-source solution. … It is affordable, and you can find easier support without having to spend money. … speaking from a company point of view I would

normally go with the product that is being licensed, but for personal use or internal use open source is simply more flexible."

Being open source is a decision that must be kept. Interviews reveal that changing the cost accounting strategy in the middle of everything is not perceived as favorable. Interviewee D tells the situation where cost accounting changed their image of the platform

"I have not used tools that switched from open source to paid. … that would also be like foul play because you started free, and people started to contribute to your product and then you saw a way to make some money. … I think there are many tools that have went through this."

Interviewee D emphasizes the technology trending and word of mouth from various sources as the factor that influences their technology acceptance the most. Interviewee G also emphasizes the popularity of the platform. Interviewee J relies on feedback or experiences from other users. Other teams and colleagues introducing the way of working with the tool's influences technology adaptation. Interviewee D explains the effect community adoption has on their behavior

"… word gets around quickly, especially with the open-source tools. It is no secret to anyone. The bigger the interest people start automatically talking about it and adopting tools and using them and this quickly comes a trending solution that quickly becomes a standard. It is like a domino effect. … it is valid because so many others are using it … the wider or the more people talk about and use certain tools the easier for others to adopt them. … hearing that other teams that are doing the same type of work that you are doing using the tool can influence you and you using the toolkit influence others."

Interviewee H highlights interoperability, flexibility, scalability, security, and a strong community base as key critical factors when they decide to use an open application platform. A strong community provides advantages when trying to reach for help and communicating with other developers. The presence of a community provides different benefits as Interviewee H justifies

> *"… we should be able to discuss the technical doubts that we have. It is like a joint development or joint contribution. A good community base to develop the platform as well as the application."*

Interviewee G has not yet been using many API platforms due to the ease of use of regular language libraries. What would make them interested in getting started with APIs would be use cases and providing the programming language they are used to. Interviewee G explains this situation

> *"I would like to see easy access to documentation and gain understanding of what this is about and what are the use cases. … regular language libraries especially Python usually provide sensible defaults but also if you want to configure something there is the option to override those default values. A beginner does not know what every option does or does not even care. An expert has this very specific use case and then it would be nice if they could achieve that."*

The partner and their privacy policies might affect adopting an open API. Interviewee C mentions user-friendliness, ease of use, and application security as the main points that affect decision-making. Interviewee C says that partner companies have personal information that they expect to be safeguarded. That is why data protection is a critical aspect to think about when deciding to develop applications on a specific platform.

The integration of diverse tools on the same platform assists various users in adapting to new technologies. This interaction is only feasible thanks to plugins and extensions.

Interviewee J emphasized the need of mobility to various platforms. Interviewee A discusses creating something that can embrace the way consumers want to work rather than forcing them to learn a new set of tools. Users still prefer to learn new instructions or keystrokes to broaden their expertise. During the interviews, it became evident that users do not want to be sidetracked during the development process

> "*When I am developing code, I want to have something that does not kind of get in my way and that requires the minimal amount of interaction back and forth. Switching between input devices and that kind of thing.*"

Although information and access to different channels may be a good idea some people get distracted reading a lot of information. It is important to include only the information developers need to implement their tasks. Interviewee A talks about their experience

> *"I am kind of being a typically inquisitive person. … I'm easily distracted by shiny things. … by having the information kind of very targeted, it kind of helps me stay focused and keeps me on track getting what I need to get done."*

Interviewee C reveals that there is a lack of a common main channel for testers to ask for help and gain access to useful tips. Interviewee C mentions that information concerning testing is found from multiple channels across Google search results, YouTube videos, Stack overflow solutions, Facebook groups, and as tactic knowledge around colleagues. Interviewee C explains the current situation surrounding testing

> *"The information is right now quite scattered and difficult to find. If I have any problems, first, I will seek Google. If I am not getting any solution, then I will go through the related videos or related stuff. After that, I will contact colleagues who have more experience. I will check with them and try to resolve the issue. … We also have Facebook channels, a Python group, and a Kubernetes group, where I might get a solution.*"

Even though information is scattered interviewee C has managed to work in the following way throughout the years. However, interviewee C shares their experience with a previous employer who had in use a testing channel dedicated to posting problems and solutions. Interviewee C thinks that this issue could be avoided

> *"I think that there should be a common channel where you could post and share views on issues. I think our life would become easier by sharing problems."*

While developers work with code and production, they have been switched to assess how the assets have been created. The phenomenon that surrounds this topic is called DevOps - where the developer does more than just code. This shift has affected developers and their way of working by moving their focus from developing to operational aspects as well. Interviewee A thinks this shift has increased the amount of work but improved the management and operation of the applications.

While making the code understandable developers need to take stakeholders' perspectives into account. Measuring key performance varies depending on the customer and their feedback. Early feedback is collected to minimize the costs and time of creating something complicated that stakeholders do not need. Interviewee A refers to this as an agile and iterative process rather than a waterfall method. It is about collecting feedback as the application is evolving rather than spending time and effort to create the application first and then getting the feedback once the product is finalized.

On the other hand, interviewee B mentions that they typically start introducing the product to the stakeholders when it is 80 percent ready and internally approved. This approach contributes to the waterfall method. Interviewee B justifies this approach by costs that are related to the testing phase when the customer gets to try out the solution. Once the product is 90 percent ready customers' opinions are collected during the trial

and the product is finalized. However, the final alterations are not enough as further development is always needed when no product is 100 percent ready.

Interviewee C also collects customer input at the end of each project. According to interviewee C, KPIs are obtained from the partner once they have had the opportunity to test the application firsthand. The partners are then asked to score their satisfaction on a scale of 1 to 5, with 1 being very dissatisfied, 2 being dissatisfied, 3 being neutral, 4 being satisfied, and 5 being extremely satisfied. The collected input will influence future testing and development activities.

There are however different approaches for collecting partner feedback. Measuring end customer satisfaction is not extensively covered in interviewee H's work. However, they mentioned the importance of metrics and performance monitoring, such as CPU usage and memory consumption, to evaluate the platform and application performance. For example, interviewee J investigates applications' performance in different scenarios and uses this information as part of customer feedback.

Direct interaction with end users can provide valuable insights. Interviewee E describes the direct feedback approach they use where the feedback button allows them to collect all user feedback instantly. Interviewee E analyzes the webpage usage with Google Analytics. Interviewee E explains the things they tend to analyze and their needs

> *"How long people are staying on your page? Are they looking at specific areas more in detail … following links? We are not using at the moment, but we could use an online tool to automatically test application performance on our customer devices … and draw our conclusions in automated fashion."*

Interviewee D talks about their company's approach where the technical personnel do not get the customer feedback directly. There is a lack of real knowledge, but Interviewee D explains that if partners are satisfied with the services the company provides, they will

bring more services to our platform. Interviewee G also believes that the number of users on the platform and the number of API calls made are good indicators of customer satisfaction. This blind spot in information flow can however create issues for the feedback receiving the correct people who influence the development. Interviewee D reveals their company's situation

> *"Customer feedback is not directly sent to us, so if a customer is indeed satisfied and wishes to express their satisfaction and feedback, it usually goes to a manager or to an account manager or account engineer. It is not something that we are handling. I would assume that the main key performance indicator for us is usage and consumption."*

The idea of the xApp platform divides opinions and emphasized that there is always room for improvement. Interviewee A states that they typically work with open-source technologies, and they see a lot of potential for new and interesting use cases, machine learning opportunities, and ways to minimize latency in the cloud-type environment. Interviewee B emphasizes the developer requirements like common baseline development language and the developing and software functionalities of the development environment. The benefits also include the possibility to minimize on-premise locally available resources and leverage them instead into the cloud.

Interviewee C addresses the xApp platform's requirements from the standpoint of a tester. It is considered beneficial if a platform gives troubleshooting information and the ability to provide recommendations for issue remedies. Interviewee C for example supplements their testing using Stack Overflow, cheat sheets, YouTube videos, and tutorials. These support platforms provide useful knowledge to testers and allow them to interact with other testers.

Interviewee H mentions at first that they might be biased since they have been working with xApp development for the last three years. Interviewee H also has several years of

expertise in the field. Keeping this in mind interviewee H rates the idea highly and thinks the platform has the potential to boost innovation, facilitate faster development, and provide a platform for collaborative development.

Interviewee J has been working with xApps since the beginning and underlines the quick development prospects. According to interviewee J, the framework enables developers to simply launch the application and incorporate their business logic into it. As the case company's main xApp developer, interviewee J explains the logic behind the xApp platform and addresses the difficulty that developers may encounter

> *"There are two parts. One is creating the initial application and the other part is adding your own business logic. This first part, using the framework that we have is very simple … this first part can be done pretty quickly. … then adding your own business logic … Every application has its own functionality. That part then depends on the level of experience that the developer has. … if it has something to do with this telecom or something … then I think that would require some kind of good knowledge about telecom environments. For example, if you want to develop an xApp to optimize some network elements or do some kind of energy saving … I think that would require some kind of telecom knowledge. At least understanding of the network elements that your application is trying to interact with and to do some kind of optimization. It is that part I think is a bit more complex I would say."*

Interviewee B has been hearing some early customer feedback about the current solution. It seems like the case company's previous research has shown that there are problems related to the documentation and the developer environment. This has been discovered also in the survey. Interviewee B explains the current customer issues

> *"Customers have raised some issues on the quality of the documentation, the functionality mismatch, and their environment differences."*

These problems can relate to the lack of understanding and knowledge about the field of specific technology. As interviewee J previously stated, the business logic that the developers want the program to apply may cause some challenges, particularly if the developer does not understand telecommunications. Interviewee B mentions that partners who might have developer knowledge but lack telecommunication knowledge typically expect that the product is finalized in days while the process of development is more time-consuming. Integrating different backgrounds can become a problem when trying to make partners satisfied with the result

> *"A partner who has only development knowledge but who does not have telecommunication knowledge will be very difficult to work with. … They expect everything is going to be working on day one of our development."*

Interviews revealed that the provided API requires expert knowledge about mobile networks to be useful which may limit its appeal to a broader range of developers. Interviewees E and G think that the xApps are too low a level of knowledge and the user is expected to have expert-level knowledge about the telecommunication industry when they have close to none. Interviewee E thinks that the idea is too complicated for most developers. Interviewees E, F, and G comment on the idea

> *"A fair amount of telecommunication jargon … The person is assumed to have too much knowledge and familiarity with mobile network applications and the RAN world to truly leverage the systems. … There is a need to understand what the use is in the end because the more you put into an application, the more complex it gets, and the more things can go wrong. I feel this is already a lower layer API that is provided here that requires quite a bit of expert knowledge about mobile networks to be useful."*

Interviewee H has worked within the RAN field for several years and mentions their background in RAN as beneficial for developing applications tailored to specific use cases. Interviewee H, however, thinks that the level of required RAN (Radio Access Network) knowledge depends on the specific use case. While some applications may require extensive RAN knowledge, others may not need it.

Interviewee D has not heard of a similar type of telecommunication radio technology tool but finds the idea intriguing. The idea is deployable in interviewee Ds opinion, but they do not know what to emphasize in this assessment since they have been working in different role rather than development or testing. Interviewee D talks about their opinion on how technical people usually view documentation

> *"I think people like to test and work with things before reading the document. I think this is how people in this field are built. … people hate reading through the documentation … you would rather bang your head against the wall 20 times than go read the document for 5 minutes."*

The interviewee describes their impression of the current documentation

> *"It is okay. I have seen far worse. … if I were to start tomorrow this [documentation] would bring me to more than a decent starting point but you would require some sort of assistance and support … if I were to have an interaction with someone to explain what I am failing to achieve it would give me knowledge to ask for help. You probably have additional documents for different use cases or different features."*

Interviewee G feels that the documentation lacks speaking to the reader. Visual noise is also one of the problems that interviewee G states. Interviewee G reveals their biggest issues with the current documentation

> *"… it is just stating things very bluntly. … There is this sort of feeling it is a big mess. There is too much visual noise, not that much text and … too many various visual elements. … Things that you can do with markdown pages is basically what I consider to be good for documentation. I did not find any use cases. In the sense that for example like I want to create … a machine learning application. There was not any example of … what you do to create a machine-learning application. There was also … Python library as a reference but there was no getting started guide on how to start using the library. … Even if you asked me, how would I get started with the library I would not know what to say."*

Interviewee G addresses the community aspect and emphasizes the developer journey. Interviewee G thinks that Stripe has successfully managed to do this with their solution

> *"Stripe has quite good developer journey. The documentation part puts lots of emphasis on covering many use cases instead of trying to cover all the features. I think that is a bit too dry to just cover feature by feature. Instead, they usually cover by use case by use case. I remember one example was when you want to set up a payment method on your website and they basically walk you through the steps of how to get that payment widget on your website."*

Interviewee G shares a previous example of a good developer journey with Mapbox

> *"I think Mapbox.com has a very nice developer journey. It is all about getting the sort of first Hello world as fast as possible. It is straightforward. They do not ask too many questions up front. … if I recall correctly, they do not even ask payment details. They provide given the API keys that are already generated, you just have to go copy paste them and that is it."*

The competing solutions mentioned by interviewees include AWS, Microsoft, Twilio, Google developer solutions and previously mentioned Stripe and Mapbox. While

interviewees state that both Google and AWS provide multiple tools, they do not offer solutions for telecommunication application development. Interviewees A and B like the capabilities that AWS provides for the developers. Information is said to be easily consumable, packaged, and targeted. Interviewee A addresses the cleverness of AWS informative packages as

> *"... areas that come up in a typical development process and so you can quickly get to the information you need without being sort of bogged down with having to dig through lots and lots of kind of superfluous information."*

The simplicity of coding and letting the platform do the rest is perceived as attractive. Interviewee E explains their expectations of a platform

> *"… develop some methods, some code, and then give it to Microsoft a WS, or Google and they will just run it for you. They will scale it for you, and you do not have to see any server deployment. That or no networking, no nothing. They would just take care of it for you. That is the kind of level of service I would expect these days."*

The niche way that most platforms utilize their tools makes it hard to teleport the information to some other platform that is not perceived as attractive. That is why open-source platforms are gaining popularity. Interviewee A mentions the characteristics of a good platform

> *"… it would be portable rather than kind of causing me to have to be locked in to say that one specific environment."*

Interviewee E feels that most developers nowadays do not develop their products from scratch. Instead, developers try to learn what others have already developed and copy that if possible. That is why chatbots are gaining popularity. ChatGPT is one of the tools

interviewee E has learned to use and would wish some kind of similar solution to be used on a platform. Interviewee E explains why they would prefer a chatbot instead of a documentation

> *"I would really like to have a bit more interactive platform. I would really like to have a chatbot or something to interact with to tell me how I have this problem, can you point me to where I need to go for the solution or maybe even provide solution already? I think that would be more useful for me than this document because if there is a solution there at that point it might be outdated because there is a new version of the software."*

One special thing rose as a topic when discussing what interviewees are looking for from a good partner whom they collaborate with and sell their expertise. Interviewee A mentions the previous trust issues they had with partners and how it affected their future alignment

> *"… the partners were using us to be sort of Guinea pigs, where they would enter an unknown or untested area … they would get their partners to spend resources and time to see if whatever the idea was success or if it failed. If it fails, they would just drop it, and if it was successful, taking it over themselves … they would take whatever the partner was doing and do it themselves and sort of cut that partner out. It was a very almost abusive kind of relationship … and you lost any trust going forward and you never did not really want to work with that partner again because they were more or less using you."*

When discussing potential partner firms, interviewee C acknowledges that communication and transparency are essential for a successful project. According to interviewee C, being transparent entails being clear about the project process, matching the timetables of developers and testers so that everyone is on the same page, and committing to reasonable deadlines that do not necessitate overworking. Transparency

entails informing testers about the testable application early in the process and providing documentation so that they can start developing their testing strategy from different aspects regarding system functionality, performance, and scalability.

Getting trustworthy partners to the platform is key in keeping the developers attached and keep using the platform. Both developers and partners are valued while developers are expecting their partner to support them in that overall solution and help them to be successful. The partner should be open especially when the partner company is larger. The openness also concludes with openness about the strategy and what the partner wants to accomplish. Partners should also accept criticism.

A bad partner would lack technical expertise and deliver poor-quality assistance. Interviewee J highlights that a business partner should have a strong understanding of the domain and the specific area of development, such as xApp development. They should also possess relevant knowledge and experience to provide effective assistance.

Interviewee A says that the developer wants to be aligned with their partner's road map and offer a joint solution to their shared customer base. Interviewee A explains the relationship between a developer and a partner company

> *"… where a lot of companies keep the customer close and have a good open working relationship, partners must have an even tighter relationship. … I am investing a lot of my own resources to align with my partner. … and the relationship needs to be very close because there is a dependency especially if I am working with the larger partner, I would have a lot more to lose than the larger partner would if the relationship should not be very close and stable. … Opening and being much more transparent I think is very, very important not just for my success, but I think also for my partner's success because we are going to be so tightly linked together in our success."*

Getting trustworthy partners to the platform is not the only thing a company should be concerned about. Developing trust in the platform itself with potential customers is critical. This trust can be built by making guarantees about keeping the platform alive over time. Interviewee E explains the situation they want to avoid

> *"… if that platform goes out of business or is basically no longer maintained … I need to understand the support structure of the platform. … I need some guarantees when I roll out something that it will be there in one years' two years' time. Otherwise, I cannot be sure that I should bet on it. … the effort spent may then be in vain, essentially because I have to redo it on another platform."*

4.3 Building trust among developers and partners is crucial

Interviews revealed that if developers encounter any problems related to trust, they do not want to work with that partner again. Therefore, the platform owner must take care of the relationships and create a safe and stable environment. This leads to legal and problem situation handling in which the platform must take a stand. While the dispute processes take a lot of resources trust building can be taken as an advantage for marketing purposes.

One solution to solve the misuse of partner relationships would be giving both partners and especially developers the possibility to report any misuse to the platform provider. On the other hand, this creates an issue of resource allocation. The platform provider needs to plan resources to become the go-between in the situation and solve the situation. This can however affect the relationship the platform provider has with both developer and partner relationships in the wrong way. Developers and partners might leave the platform and leave negative reviews based on the solution that the platform provider has offered. One might think that the platform provider favors the other participant and might express their feelings online. Whatever the scenario is there is a possibility of losing potential platform participants.

If the misuse that interviewee A explained happens a lot, there is a possibility to appear as a solution for this kind of behavior. This could be beneficial, especially in fostering the developers to the platform as the platform could market itself as a trustworthy place to create applications for the partners. Since other platforms might have failed in the past building developers' trust there could be made a promise for them on behalf of the platform regarding the misuse cases. The platform could keep developers' side on disputes against larger companies who might have more power to alter the relationships with their developers.

Since trust and risk-taking to support a partner are outlined in the interviewees' answers there could be some type of legal agreement. To reduce the trust issues another solution is to provide a contract or agreement with the platform provider between the developers and the partners to build trust. Misuse of this contract could for example end up banning the partner or developer from the platform for a certain amount of time. This would make the willingness to take advantage of the developers or vice versa the partners smaller.

One solution could be a contract or terms of use when signing up to the platform for each developer and partner who enters the platform. There could be a channel dedicated to reporting breaking the terms which would make the willingness to take advantage of partnerships more difficult. Breaking such terms could for example lead to restricting the use of the platform. Either way, the relationships would be more valued and hopefully more cared for in contrast to situations where there would not be any terms or contracts for platform use.

Trust also goes a long way with the platform provider. Interviewees revealed that they feel hesitant to trust platform providers and need some type of guarantee or support to make sure the platform is still in use in a few years. Customers want to make sure that they do not risk losing resources by developing something useful on a platform to lose it and having to create it somewhere else if the platform goes out of business.

4.4 Developers need to be encouraged to learn something new

Interviews revealed that developers need a push to adapt and learn new technologies. Interviewees revealed something that must be overcome, a hurdle that might influence their use of new technology. Developers compare the new technology to the existing ones they have already learned and decide based on how easy it could be to learn and how they feel this new technology might help them in the future.

To make this decision easier the technology must contain certain characteristics. Developers tend to like integrated solutions that are later used and that can be transferred to other platforms. This links to open-source code which can be created on other platforms but can still be used openly on other platforms. A platform that harnesses this type of open-source code possibility and integrates its solutions might have a competitive advantage over those who restrict theirs.

Apart from integration and open-source code, the platform must have information dedicated to fulfilling the developers' needs without giving too much information to digest. Videos, tutorials, and articles are perceived as the most useful when looking for information regarding newer technologies according to the interviews. These might also be the best channels to market to developers as they browse the internet looking for answers.

Some interviewees perceive physical magazines and newspapers as useful sources to find information. This approach is however not considering the younger generations who mostly rely on to find information online. Case company focuses to target various people who have at least one year of expertise in the field. While some target audience has more experience in the field and are thus expected to be from older generations survey and interviews reveal that digital marketing channels are perceived as more relevant than traditional marketing channels when deciding to market the open application platform to customers.

Word of mouth and creating hype around technology might be a good idea according to interviews. Interviewees talked about the significant effect trending technologies and the effect colleagues and other companies have on their decision-making. As word of mouth has a great impact on customers, the reviews and the way platforms are marketed should be well designed.

The platform needs to invest in the future to keep the customers and partners happy. Investing to create the open-source platform is not going to end once the platform is created and the first customers have started using it. Interviewees raised the concern of cheap switching costs that are related to open-source technologies. If there are no agreements, licenses, or costs associated with the use of platform customers do not see any problem switching it to something new. This is something case company needs to address while developing its platform.

While the user experience has a great impact on how the customers utilize the open API platform the company should focus on giving a realistic image of the platform in their advertising. If the switching costs are low the user experience should exceed the user's needs. The company needs to carefully view the user expectations and advertising material so that they align with current customer satisfaction and user reviews.

Moving from a free open-source platform to one that requires payment midway through the development process is viewed unfavorably. Customers have been using the platform for free, and after they have learned how to utilize it, they should begin to pay for it. Interviewees believe they are being used and find the situation extremely frustrating. To limit the likelihood of this negatively impacting situation occurring, cost measures should be developed well in advance when exposing the platform to external clients and partners.

Interviews revealed that software testers have difficulties finding a channel for mutual discussion with other testers. This can be an opportunity for the case company to bring all testers into the same community to discuss their difficulties and spread their knowledge. Testers mentioned that they prefer to ask their colleagues directly for help or search for the answer online. However, this takes time, and testers mentioned that a mutual channel to ask and provide help would be appreciated.

4.5 Developers that have more experience have fewer expectations

The survey revealed that there is a correlation between the more the person has developer knowledge and having fewer expectations. This is an interesting finding regarding the fact that the situation was the opposite for students. The more students have tester knowledge the less they have expectations. Thus, testers tend to trust that an open application platform is going to provide them with more and have higher expectations whereas developers tend to not get their hopes that high.

This finding can be explained by the character of the developer's job since they are directly going to be implementing applications with the help of provided tools. Testers are going to check if the developed solution fills the requirements and get attached to the code once they find an issue. It is the developers' job to get familiar with the API and if there are problems, they will be the first to know. Therefore, it is expected for the developers to think carefully before implementing anything and keep their expectations low.

Interesting is that there is an existing moderative correlation between xApp development and needed skills. Thus, experienced xApp developers think moderately that to be able to develop an xApp the developer needs to have expert level skills. If a beginner level developer has big expectations and gets the impression, they need to be an expert to develop an xApp this might lead to motivational problems and them abandoning the platform. Thus, xApp developing is not beginner friendly.

Interviews revealed the same type of hesitation to adapt to any new technology and commit to a platform. Developers are interested in trying new things once they get familiar with it and feel that it is necessary. Complexity and lack of usability improvements are the main things developers highlight in the interviews that may affect the adoption and the actual use of the platform.

4.6 Summary of the findings

The goal of this thesis is to understand the technological requirements to improve software developers' user experience and attract new customers. This research is gathered from a global telecommunication company and is based on a real situation. The research is case study research, which uses both quantitative and qualitative research methods. Information gathered for this research is collected via interviews and surveys from the target group.

Based on the contribute, the following factors are the considerations that software developers, testers and organizations weigh when deciding to use an open application platform as each of the factors contributes to the overall assessment:

- Familiarity and comfort
- Learning curve
- Traction and reuse
- Integration and tooling
- Ease of use
- Application security
- Developer operations (DevOps)
- Corporate policies and IP rights
- Licensing and contribution
- Cost
- Documentation and support
- Interoperability and scalability

Developers often prefer open-source technologies and frameworks that they are already familiar with and have experience using. They have a set of core technologies they are comfortable with and tend to stick with them unless there is a strong reason to switch. Developers consider the learning curve associated with new technologies. While they are open to exploring new things, they weigh the effort required to learn and adopt a new platform against its potential benefits.

Developers pay special attention to the traction and popularity of the technology within the industry. They look for signs of active community support, such as discussions, contributions, and tutorials. The availability of hands-on tutorials and practical resources helps in their decision-making process. To avoid pitfalls developers, make sure there is active maintenance and user engagement on platforms.

The ease of integration with existing tools and development environments is crucial. Developers prefer platforms that can be extended and integrated with other tools, such as version control systems and continuous integration/delivery pipelines. They look for streamlined processes and minimal switching between different tools, especially those tools that they prefer to work with.

Apart from integration developers prioritize the ease of use of an application platform. They look for platforms with user-friendly interfaces, straightforward configuration steps, and a simple registration or login process. The platform's ease of use reduces development effort and enhances productivity.

Application security is a critical factor for developers. They value platforms that prioritize data protection, safeguarding personal information, and providing secure mechanisms for users. Trust in the platform's security features, platform provider, and other business stakeholders is essential.

Developers are expected exceedingly to handle operational aspects alongside coding. They consider platforms that facilitate the transition to DevOps, making it easier to deploy and manage applications in production environments. That is why developers evaluate the platform's readiness and ensure important functionality is available. Ease of operation and management becomes an important factor to ensure stability and efficient feature development.

Organizations' corporate policies and intellectual property (IP) considerations play a role in the decision to use open application platforms. Developer verifies that the platform's licensing allows for the intended use, considering business restrictions or requirements. Some proprietary technologies may not align with the company's IP strategy, while open-source solutions can provide flexibility and cost benefits.

Cost is a significant consideration, especially for service providers or organizations seeking to avoid licensing fees. When using an open application platform, developers evaluate the licensing associated with it and consider whether they want to contribute to the platform. They assess the licensing terms, the potential for contributing their functionalities or improvements, and the decision process for contribution. Open-source solutions are often preferred for their affordability and the availability of community support, reducing the need for additional expenditure.

Developers emphasize the importance of comprehensive documentation and good support structures as they research user experiences, problems, documentation, troubleshooting support, and feedback related to the platform. They look for platforms with well-documented APIs, clear usage guidelines, and easily accessible support channels. A strong community base, developer forums, and responsive customer support are also very much appreciated.

Ensuring compatibility with the project and avoiding limitations that could hinder development or usage is important. Open application platforms should offer

interoperability to work with different types of applications and programming languages. Flexibility and scalability are important attributes, enabling customization and accommodating growing demands for traffic and data.

5 Discussion, conclusions, and suggestions for further research

This thesis aims at answering the following research questions: “What are the possible alternatives to improve the user experience for different user groups in an open API platform?” and “How could the open API platform be marketed based on these user expectations?”. Quantitative and qualitative methods are used with data from interviews and survey. The thesis has both theoretical contributions and practical implications.

The research has enhanced our understanding of motivational factors, the balance between system performance and complexity, the role of community in user engagement, and the significance of user experience in open application platforms. Additionally, it has highlighted instances where our findings may diverge from existing theories, underscoring the dynamic and evolving nature of the field. These contributions to theory collectively strengthen the relevance and significance of research within its domain.

To improve the user experience for different user groups in an open API platform there are many things to consider. Participants were divided into software testers and developers and in some scenarios to externals, internals, and students in the survey. Participants role was considered also in the interview. As mentioned earlier the nevertheless to which user group participants belong, developers or testers, they value and consider familiarity and comfort, learning curve, traction and reuse, integration and tooling, ease of use, application security, developer operations (DevOps), corporate policies and IP rights, licensing and contribution, cost, documentation and support and interoperability and scalability. To excel in creating a successful open API platform the company must consider the following things that affect the overall assessment of the platform.

These thesis findings suggest that there are some changes in the way a person views the open API platform. Some differences were found regarding the community aspect of platforms since testers found that they do not have many common ways to communicate

with other testers. One of the major findings that creates a difference between testers and developers is the fact that developers who are more experienced have lower expectation. This must be noted once defining improvements that affect user experience on a specific user group.

The open API platform can be marketed eventually by reinforcing the positive outcomes and by tackling the obstacles that lie ahead. This can be achieved by clearing doubts, fears, and overlining the possibilities rather than problems of xApp development and testing, creating an equal rights and active platforms with strong community. The introduced marketing channels that participants find the latest information are social media and word of mouth. What stood out from the interviews is that there are still some people who prefer to read news and articles on paper physically but most of the participants preferred the digital places and communities built upon social media. Therefore, these digital channels would be wise places to market.

YouTube and GitHub were participants' favorite social channels in the survey and where participants found the most interesting content. These social channels could be targeted by the company's digital marketing efforts. Other marketplaces to be considered are Stack Overflow and LinkedIn, which over half of the participants found desirable. Company can use the knowledge of the most common technologies, tools and domains and fulfill these wishes in the platform. These can be later used as marketing efforts and be expressed to the public to attract more users to the platform. Once the open API platform has gained a foothold in the market this victory must be expressed in social channels to create a cycle of hype since participants value hype and the WOW-factor.

First, the thesis contributes to theory by addressing learning as a motivational force by indicating that learning serves as a major driving force for individuals to engage in open-source software (OSS) communities, even when no monetary benefits are involved. In the study of Ye and Kishida (2003) indicate that learning offers immediate satisfaction

and the potential for personal growth within platform communities. This thesis' findings support this indication.

Second, the thesis contributes to theory by addressing balancing system performance and complexity by indicating that there is a challenge finding a balance between system performance and complexity. Aryal et al. (2023) previously highlighted this dilemma, and research results reinforce this idea. The challenge of maintaining a high-performance environment that may lead to complex system architecture versus a less complex design resulting in lower performance is a recurring theme in this study as well.

Third, the thesis contributes to theory by addressing the role of community as motivation by indicating that the role of community motivates user engagement. Ye and Kishida (2003) proposed that by participating in a community, developers and users learn from the system and from each other, thereby enhancing their involvement. Research findings echo this idea, with multiple interviewees mentioning the platform community as a key factor influencing their decisions to use open application platforms. The community is seen as a place to seek help, develop both the platform and its applications, and foster a sense of collaboration.

Fourth, the thesis contributes to theory by user experience and the balance of functionalities by indicating that user experience is important when developing open programmable networks. Aryal et al. (2023) underscored the challenge of balancing functionalities and keeping the platform simple, which our research results corroborate. Interviews and surveys in this study indicate that users often perceive telecommunication technology as overly complex. In this context, our findings align with Aryal et al.'s observations regarding the complexity of technology.

While the thesis offers substantial contributions to the theoretical landscape, it is important to acknowledge instances where findings may provide an alternative perspective to existing literature. Notably, one of the findings of this research is that

more experienced developers have lower expectations. This discovery is a novel addition to the field of study, as no identical results from previous research were discovered during the investigation. As such, it is apparent that future studies will be required to validate this unique finding.

Based on theoretical findings, the practical implications of this research are motivational issues that the company needs to address. The company needs to build a strategy to monitor the process tackling these motivational issues. The practical implications to help address these problems are clearing doubts about the technology, showing what possibilities lie within xApp development, ensuring that every participant is treated equally, being an active platform, and trying to build a strong community.

Companies willing to invest in open application platforms will benefit from these recommendations. Especially those companies that specialize in telecommunication and would like to create their own open API platform around xApps would benefit from this study findings and practical implications. Companies wanting to expand to open API platform would benefit from these results since the motivational issues can be reviewed and altered to fit the business need in the specific technological scene.

The practical implications will help the companies to create a strategy and investigate the need to tackle these problems before the company moves any further with any other activities. The weakest link of motivational issues needs to be found, a long-term strategy defined and the xApp strategies must be updated to keep track of the process. Finding the solutions to motivational issues not only helps with change resistance but makes the platform resistant to hardest criticism. Eventually the effort will lead to a situation where the company has enough competence to enter the market and introduce their platform.

This chapter aims to discuss the theoretical implications of the study, highlighting how the thesis contributes to existing knowledge and theoretical frameworks offering a comprehensive view of the research's significance within its field of study. Ishikawa

fishbone diagram on motivational issues presenting investing into API applications is introduced. At the end of this chapter thesis is going to provide recommendations for further research.

5.1 Comparison between findings and earlier literature

Few studies and findings have been published on the user experience of O-RAN and open application platforms that focus on programmable networks, much alone xApps. Kougioumtzidis et al. (2022) discovered that O-RAN offers potential for establishing "new business models that will cut expenses, improve corporate efficiency, and facilitate more innovation". One of the research findings was the interest in xApps and the creative opportunities it offers, but as this study suggests, developers need to be pushed to learn something new.

To examine what affects user experience on an open programmable network Kak et al. (2022) constructed an O-RAN platform. The platform increased “the user experience with an intuitive graphical interface and accessible APIs”. Kak et al. (2022) did not however explain how improvement in user experience was tested and what affected the improvement.

Other research focuses on the future challenges around O-RAN user experience. Kougioumtzidis et al. (2022) found that the quality of experience (QoE) is important in O-RAN architecture. As Kougioumtzidis et al. (2022) assess QoE in terms of service quality (QoS), this study evaluates QoE in terms of open application platform quality. As a result, the findings of Kougioumtzidis et al. (2022) and this study cannot be directly compared.

One of the research findings is that users want an open application platform that is easy to use, familiar and comfortable. In other words, users need to be encouraged to learn something new. Aryal et al. (2023) discovered that there is “a challenge finding a balance

between system performance and complexity". This is also what research results indicate. Finding the balance between the functionalities and keeping the platform simple enough is a challenge.

The thesis also contributes to theory by addressing the user experience in open programmable networks. Aryal et al. (2023) highlighted the dilemma of having "high-performance environment that may result in a complicated system architecture whereas a less complex design may result in a system with lower performance". Results indicated a similar problem with the complexity of telecommunication technology. Results from interviews and surveys suggest that users do not feel comfortable with the technology and think that is too complex for them to use it. These parallel findings emphasize the importance of addressing this complexity-performance trade-off in open application platform design, underlining the theoretical significance of this research.

Users would also want to integrate tools from other sources. Aryal et al. (2023) results indicate that there is "a challenge to converge between third-party services". While interviewers mentioned that they would be interested in trying out the newly proposed open application platform they still would prefer to use their earlier adopted third-party tools and integrated solutions.

Aryal et al. (2023) mention operational costs as one of the challenges to adopt O-RAN. Interviewees mentioned cost as a critical factor when deciding which open source is generating a lot of revenue. Other than estimation of potential revenues, users also want to calculate how much the solution costs in the long run.

As mentioned earlier in analysis of interviews if there are no agreements, licenses, or costs associated with the use of platform, customers do not see any problem switching it to something new. If the switching costs are low the user experience should exceed the user's needs. To avoid operational costs that reflect to the price of platform use Aryal

et al. (2023) suggest platform owners to “continuously monitor the system and look for ways to cut costs”.

The theoretical contribution of this study extends to the role of community in motivating user engagement. Ye and Kishida (2003) found out that “learning is one of the major driving forces that motivate people to get involved in open-source software (OSS) communities with no monetary benefits”. Ye and Kishida (2003) state that learning gives instant satisfaction and provides possibility to evolve in the role of the platform community. Optional learning outside of mandatory work training is not mentioned in the survey or interviews. Platform community is however mentioned multiple times in interviews but not as a way that motivates through learning. While Ye and Kishida's research supports the idea that participation in these communities facilitates learning, research findings emphasize the role of learning as a primary motivator, underscoring this research significance in the context of open application platforms.

Ye and Kishida (2003) suggest that “by participating in the community, developers and users learn from the system, from each other, and share their learning with each other”. This is also what interviews indicate too as a couple of interviewees mention community as key critical factors when they decide to use an open application platform. Community is also seen as a place to ask for help and as one interviewee puts it develop the platform as well as the application. These commonalities between findings and Ye and Kishida's research emphasize the theoretical importance of community in user motivation.

One of the findings of this research is that more experienced developers have less expectations. During the investigation, no identical results from previous research were discovered, implying that this discovery is new in this field of study. Future study will be required to validate this finding.

5.2 Ishikawa fishbone diagram on motivational issues presenting investing into API applications

Ishikawa fishbone diagram, created by Dr. Kaoru Ishikawa (1994), also known as the cause-and-effect diagram or the fishbone diagram is a visual tool used for problem-solving and root cause analysis. The visual representation resembles a fish with a head as the problem and bones that represent what is causing the problem.

Figure 32 illustrates the Ishikawa fishbone diagram of the motivational issues and drivers that prevent investing into an open API platform that specializes in xApps (Ishikawa, 1994).

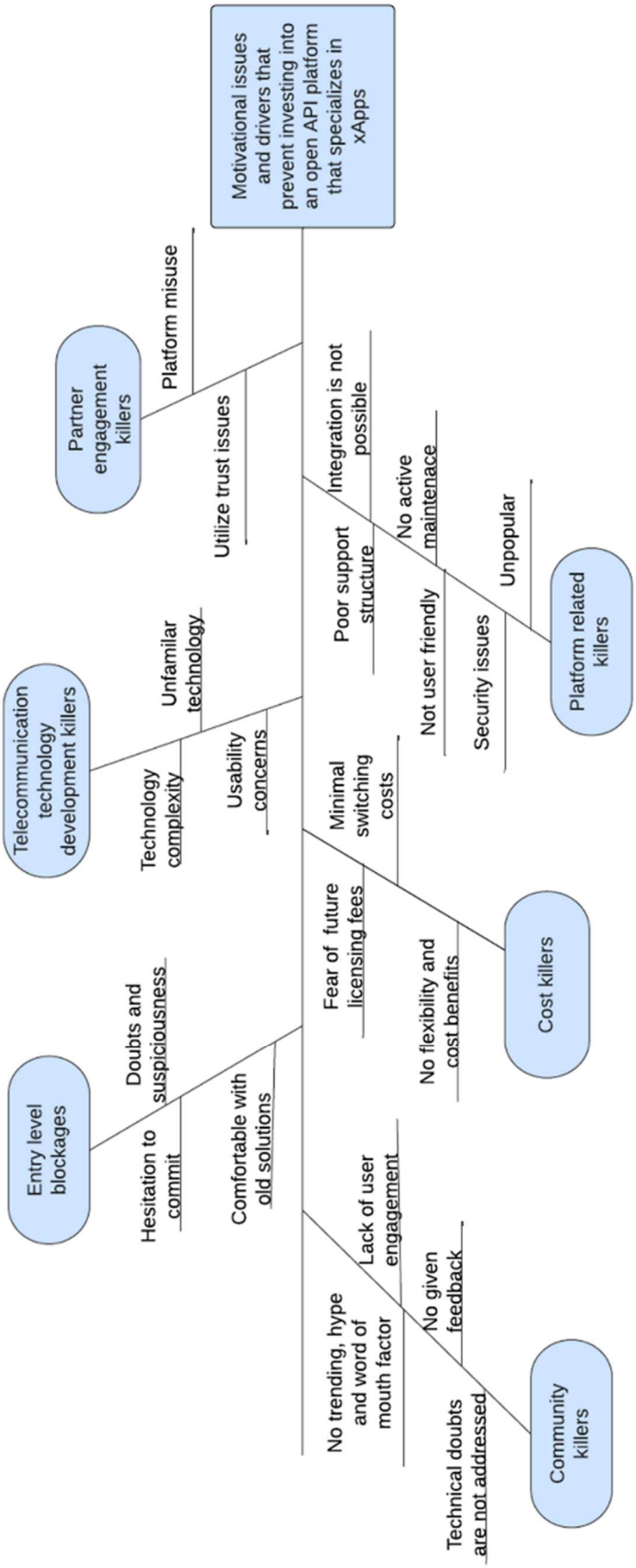


Figure 32. Ishikawa fishbone diagram of motivational issues of an open xApp API platform

As illustrated in the figure 32 the drivers and motivational issues have been divided into six groups that are entry level blockages, telecommunication technology development killers, partner engagement killers, community killers, cost killers and platform related killers:

- *Entry level blockages* are related to end user related issues that are linked to how they work and what they expect from the platform. Entry level blockages are hesitation to commit, doubts and suspiciousness and feeling comfortable with old solutions.
- *Telecommunication technology development killers* are related to companies and end users' beliefs about the development environment that the platform is arranging. Telecommunication technology development killers are unfamiliarity and complexity of the technology as well as usability concerns.
- *Partner engagement killers* are related to the concerns that end users have concerning platform partners work morals and contracts. Partner engagement killers are utilizing trust issues and misusing the platform.
- *Community killers* are associated with end users' wants and needs and how community aspect is covered in the platform. Community killers are lack of hype, trending, or word of mouth, not addressing technical doubts, lack of user engagement and no given feedback.
- *Cost killers* relate to other partner companies and end users' unwillingness to invest into the platform. Cost killers are fear of future licensing fees, no flexibility and cost benefits and minimal switching costs.
- *Platform related killers* are related to what the platform is offering. Platform killers are poor support structure, not being user friendly, security issues, integration is not possible, no active maintenance, and unpopularity.

5.3 Weakest link of future xApp marketing

In the previous chapter was illustrated the motivational factors that must be maintained under control. The weakest link must be identified. This is an ongoing task rather than a project.

Once the company finds a strategy to address these problems, being an xApp platform provider can become an asset in the future as the market around xApps is growing as shown in previous studies. Clearing doubts about the technology, showing what possibilities lie within xApp development, ensuring that every participant is treated equally, being an active platform, and trying to build a strong community around it helps in this process.

By creating a user friendly, easy to use, trustworthy and flexible xApp platform a company can gain an advantage over the other xApp platform creators. Business solutions do not need to be perfect, but it might be that it is enough to be equal or superior to its rival. There needs to be continuous research regarding the competitive situation and previously listed issues.

5.4 Vision, mission, and strategy for xApp

As discussed in the previous paragraphs, the tools of the company and how the company can monitor the changes in xApp business environment affect the current state of the company. Figure 33 illustrates how people, partners, mission, and the current state lead to the long-term aim or vision through the use of strategies. In this case, the company's vision for xApp should be to lead the way in providing state-of-the-art open application platform to customers.

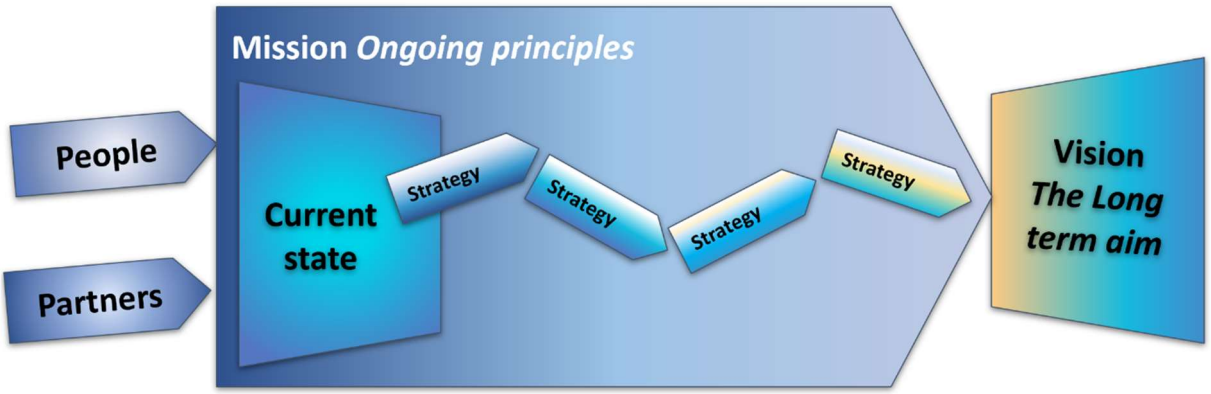


Figure 33. Illustration of vision, mission, and strategy

The company's mission for xApp should be to use unique expertise, business innovation and change management to help customers to communicate easily, efficiently and by friendly offers. This can be achieved with the help of network connections and communications services improving user experiences and productivity. As the competitive situation, technology and customer preferences are constantly changing, the xApp strategies must be updated.

On the path to the vision, it is essential to influence the xApp stakeholder groups around the platform. People who are developing the platform should have clear targets for their development efforts. On the other hand, the key partners should take part in this continuous development effort.

5.5 Recommendations for further research

This study's primary focus is discovering what user experiences developers have with open application platforms, xApps and what influences their decision to make the transition to an open application platform. Further research could be done regarding the user acceptance of rApps or similar kinds of solutions designed at developers and testers.

Because addressing these problems is never ending task, future study on the same target group might be conducted to determine how individuals have gained more expertise and familiarity with xApps and to determine what factors have the greatest impact to the

user experience. It would be interesting to see has the situation improved or gotten even worse. It would also be very interesting to study has these actions been effective.

This research focused on understanding the factors that affect xApp platform users who are also developers. Interest was particularly on those developing xApp applications that run on the platform. Additional research could be targeted to study subjects from developers and testers perspective to gain more knowledge about their user experience.

Additionally, it would be interesting to investigate if software developers with more experience tend to have lower job-related expectations compared to those with less experience. The expectations could be related to salary, job roles, work environment, or other factors. External factors like geographic location, industry, or company size could also be seen as a factor. Personal goals, job satisfaction, and external market conditions also play a role in shaping expectations. It would be good to include in the research a diversity of developers in terms of experience levels, backgrounds, and the companies that developers work for.

References

Alay, Ö., Lutu, A., Peón-Quirós, M., Mancuso, V., Hirsch, T., Evensen, K., Hansen, A., Alfredsson, S., Karlsson, J., & Brunstrom, A. (2017). Experience: An open platform for experimentation with commercial mobile broadband networks. *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 70–78.

Anerousis, N., Chemouil, P., Lazar, A. A., Mihai, N., & Weinstein, S. B. (2021). The Origin and Evolution of Open Programmable Networks and SDN. *IEEE Communications Surveys & Tutorials*, *23*(3), 1956–1971. https://doi.org/10.1109/COMST.2021.3060582

Antikainen, M. J., & Vaataja, H. K. (2010). Rewarding in open innovation communities–how to motivate members. *International Journal of Entrepreneurship and Innovation Management*, *11*(4), 440–456.

Antikainen, M., Mäkipää, M., & Ahonen, M. (2010). Motivating and supporting collaboration in open innovation. *European Journal of Innovation Management*, *13*(1), 100–119.

APIdays. (2019). *APIs + Business Models = New Ecosystems: Marjukka Niinioja, CEO of Osaango [video]. YouTube.* Retrieved 10.10.2022 from https://www.youtube.com/watch?v=7n-Q7Hr1HE0&t=176s&ab_channel=apidays

Aryal, N., Bertin, E., & Crespi, N. (2023). Open Radio Access Network challenges for Next Generation Mobile Network. *2023 26th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 90–94.

Azariah, W., Bimo, F. A., Lin, C.-W., Cheng, R.-G., Jana, R., & Nikaein, N. (2022). A survey on open radio access networks: Challenges, research directions, and open source approaches. *ArXiv Preprint ArXiv:2208.09125*.

Basole, R. C. (2016). Accelerating Digital Transformation: Visual Insights from the API Ecosystem. *IT Professional*, *18*(6), 20–25. https://doi.org/10.1109/MITP.2016.105

Benyon, D. (2014). *Designing interactive systems : a comprehensive guide to HCI, UX and interaction design* (3rd [rev.] ed). Pearson.

Benyon, D. (2019). *Designing user experience : a guide to HCI, UX and interaction design* (Fourth edition). Pearson Education Limited.

Biehl, M. (2015). *API architecture : the big picture for building APIs*. API-University Press.

Bitner, M. J., Ostrom, A. L., & Morgan, F. N. (2008). Service Blueprinting: A Practical Technique for Service Innovation. *California Management Review*, *50*(3), 66–94. https://doi.org/10.2307/41166446

Bodle, R. (2011). Regimes of sharing: Open APIs, interoperability, and Facebook. *Information, Communication & Society*, *14*(3), 320–337.

Bonati, L., Polese, M., D'Oro, S., Basagni, S., & Melodia, T. (2022). Intelligent Closed-loop RAN Control with xApps in OpenRAN Gym. *ArXiv Preprint ArXiv:2208.14877*.

Bork, D., Buchmann, R., Karagiannis, D., Lee, M., & Miron, E.-T. (2019). *An open platform for modeling method conceptualization: the OMiLAB digital ecosystem*.

Brannen, J., Bird, M., Bullock, R., Bryman, A., Hammersley, M., Laurie, H., Little, M., Millham, S., & Qureshi, H. (1992). *Mixing methods : qualitative and quantitative research*. Avebury.

Burrell, G., & Morgan, G. (1979). *Sociological paradigms and organisational analysis*. Heinemann.

Campbell, A. T., de Meer, H. G., Kounavis, M. E., Miki, K., Vicente, J. B., & Villela, D. (1999). A survey of programmable networks. *ACM SIGCOMM Computer Communication Review*, *29*(2), 7–23.

Chaffey, D., & Chadwick, F. (2022). *Digital marketing* (Eighth edition.). Pearson.

Chaffey, D., & Smith, P. R. (2023). *Digital marketing excellence : planning, optimizing and integrating online marketing* (Sixth edition). Routledge.

Chesbrough, H. (2004). Managing open innovation. *Research-Technology Management*, *47*(1), 23–26.

Chua, W. F. (1986). Radical Developments in Accounting Thought. *The Accounting Review*, *61*(4), 601–632.

Crawshaw, J. (2022a). *Self-Organizing Networks and RAN Intelligent Control – 2022*. https://omdia.tech.informa.com/OM026870/Self-Organizing-Networks-and-RAN-Intelligent-Control--2022

Crawshaw, J. (2022b). *Self-Organizing Networks and RAN Intelligent Control Market Forecast 2022*. https://omdia.tech.informa.com/OM026869/Self-Organizing-Networks-and-RAN-Intelligent-Control-Market-Forecast-2022

Crotty, M. J. (1998). The foundations of social research: Meaning and perspective in the research process. *The Foundations of Social Research*, 1–256.

Crowley, P., Fluczynski, M. E., Baer, J.-L., & Bershad, B. N. (2000). Characterizing Processor Architectures for Programmable Network Interfaces. *ACM International Conference on Supercomputing 25th Anniversary Volume*, 287–298. https://doi.org/10.1145/2591635.2667178

Davis, F. D. (1985). *A technology acceptance model for empirically testing new end-user information systems: Theory and results*. Massachusetts Institute of Technology.

Denzin, N. K. (2008). *Collecting and interpreting qualitative materials* (Vol. 3). Sage.

Dryjański, M., Kułacz, Ł., & Kliks, A. (2021). Toward modular and flexible open ran implementations in 6g networks: Traffic steering use case and o-ran xapps. *Sensors*, *21*(24), 8173.

DuVander, A. (2012). *8,000 APIs: Rise of the Enterprise*. ProgrammableWeb Blog. https://www.programmableweb.com/news/8000-apis-rise-enterprise/2012/11/26

Feamster, N., Rexford, J., & Zegura, E. (2013). The road to SDN: An intellectual history of programmable networks. *Queue*, *11*(12), 20–40.

Feamster, N., Rexford, J., & Zegura, E. (2014). The Road to SDN: An Intellectual History of Programmable Networks. *SIGCOMM Comput. Commun. Rev.*, *44*(2), 87–98. https://doi.org/10.1145/2602204.2602219

Fernando, J. (2023). Compound Annual Growth Rate (CAGR) Formula and Calculation. *Investopedia*. https://www.investopedia.com/terms/c/cagr.asp

Foux, G. (2010). Integrating social into your business. *Journal of Direct, Data and Digital Marketing Practice*, *12*, 128–136.

Gobo, G. (2008). *Doing ethnography*. Sage.

Goldratt, E. M., & Cox, J. (2016). *The goal: a process of ongoing improvement*. Routledge.

Guba, E. G., & Lincoln, Y. S. (1994). Competing paradigms in qualitative research. *Handbook of Qualitative Research*, *2*(163–194), 105.

Hannes, V., Lasse, T., & Sami, S. (2018). The entrepreneurial ecosystem and born globals: the Estonian context. *Journal of Enterprising Communities*, *12*(2), 117–138. https://doi.org/10.1108/JEC-08-2017-0056

Hayes, A. (2022). Average Annual Growth Rate (AAGR): Definition and Calculation. *Investopedia*. https://www.investopedia.com/terms/a/aagr.asp

Hewett, T. T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G., & Verplank, W. (1992). *ACM SIGCHI Curricula for Human-Computer Interaction*. Association for Computing Machinery.

Hirsjärvi, S., & Hurme, H. (2008). *Tutkimushaastattelu : teemahaastattelun teoria ja käytäntö*. Gaudeamus.

Hollweck, T. (2015). Robert K. Yin.(2014). Case Study Research Design and Methods . *Canadian Journal of Program Evaluation*, *30*(1), 108–110.

Huhtamäki, J., Basole, R. C., Still, K., Russell, M. G., & Seppänen, M. (2017). *Visualizing the geography of platform boundary resources: The case of the global API ecosystem.*

International Organization for Standardization. (2018). *Ergonomics of human-system interaction — Part 171: Guidance on software accessibility (ISO Standard No.9241-171:2008)*. https://www.iso.org/standard/39080.html

Ishikawa, K. (1994). *Guide to Quality Control* (Twelfth Printing). Asian Productivity Organization.

Iturria-Rivera, P. E., Zhang, H., Zhou, H., Mollahasani, S., & Erol-Kantarci, M. (2022). Multi-agent team learning in virtualized open radio access networks (o-ran). *Sensors*, *22*(14), 5375.

Järvinen, P. (2001). *On research methods*.

Kak, A., Pham, V.-Q., Thieu, H.-T., & Choi, N. (2022). ProSLICE: An Open RAN-based approach to Programmable RAN Slicing. *GLOBECOM 2022-2022 IEEE Global Communications Conference*, 197–202.

Karhu, K., & Ritala, P. (2021). Slicing the cake without baking it: Opportunistic platform entry strategies in digital markets. *Long Range Planning*, *54*(5), 101988.

Kilpeläinen, O., yliopisto, V., Vaasa, U. of, yksikkö, J., & Management, S. of. (2021). *Value creation through API ecosystem : Case KONE*. https://osuva.uwasa.fi/handle/10024/12517

Kompany, R., Killeen, A., Gabriel, C., & Rao, A. (2021). Near-real-time RIC: enabling AI/ML-driven extreme automation and granular control of Open RAN. *Analysys Mason Limited*.

Kotler Hermawan; Setiawan Iwan, P. K., Kartajaya, H., Kotler, P., & Setiawan, I. (2016). *Marketing 4.0*. John Wiley & Sons, Incorporated. https://sfx.finna.fi/nelli07?url_ver=Z39.88-2004&ctx_ver=Z39.88-2004&ctx_enc=info:ofi/enc:UTF-8&rfr_id=info:sid/sfxit.com:opac_856&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx&sfx.ignore_date_threshold=1&rft.object_id=3710000000952392&svc_val_fmt=info:ofi/fmt:kev:mtx:sch_svc&

Koufos, K., Haloui, K., Dianati, M., Higgins, M., Elmirghani, J., Imran, M., & Tafazolli, R. (2021). Trends in intelligent communication systems: Review of standards, major research projects, and identification of research gaps. *Journal of Sensor and Actuator Networks*, *10*(4), 60. https://doi.org/10.3390/jsan10040060

Kougioumtzidis, G., Poulkov, V., Zaharis, Z. D., & Lazaridis, P. I. (2022). Intelligent and QoE-aware Open Radio Access Networks. *2022 3rd URSI Atlantic and Asia*

Pacific Radio Science Meeting (AT-AP-RASC), 1–4. https://doi.org/10.23919/AT-AP-RASC54737.2022.9814435

Labrecque, L. I., vor dem Esche, J., Mathwick, C., Novak, T. P., & Hofacker, C. F. (2013). Consumer Power: Evolution in the Digital Age. *Journal of Interactive Marketing*, *27*(4), 257–269. https://doi.org/https://doi.org/10.1016/j.intmar.2013.09.002

Lacava, A., Polese, M., Sivaraj, R., Soundrarajan, R., Bhati, B. S., Singh, T., Zugno, T., Cuomo, F., & Melodia, T. (2022). Programmable and customized intelligence for traffic steering in 5G networks using Open RAN architectures. *ArXiv Preprint ArXiv:2209.14171*.

Lahtinen, N., Pulkka, K., Karjaluoto, H., & Mero, J. (2022). *Digimarkkinointi : luo strategia, vie se käytäntöön, myy enemmän*. Alma Talent.

Latham, G. P., & Pinder, C. C. (2005). Work motivation theory and research at the dawn of the twenty-first century. *Annual Review of Psychology*, *56*(1), 485–516. https://doi.org/10.1146/annurev.psych.55.090902.142105

Laugwitz, B., Held, T., & Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. *HCI and Usability for Education and Work: 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008, Graz, Austria, November 20-21, 2008. Proceedings 4*, 63–76.

Lewko, C., & Parton, J. (2021). *Developer relations : how to build and grow a successful developer program*. Apress.

Lies, J. (2019). *Marketing intelligence and big data: Digital marketing techniques on their way to becoming social engineering techniques in marketing*.

LightCounting. (2020). *RAN Automation Report: When the SON sets, the RIC rises, and the RAN gets SDN'ized*.

Liu, M., Tu, Z., Zhu, Y., Xu, X., Wang, Z., & Sheng, Q. Z. (2021). Data correction and evolution analysis of the ProgrammableWeb service ecosystem. *Journal of Systems and Software*, *182*, 111066. https://doi.org/10.1016/J.JSS.2021.111066

Macedo, D. F., Guedes, D., Vieira, L. F. M., Vieira, M. A. M., & Nogueira, M. (2015). Programmable networks—From software-defined radio to software-defined networking. *IEEE Communications Surveys & Tutorials*, *17*(2), 1102–1125.

Mäkipää, J.-P., Norrgård, J., & Vartiainen, T. (2022). Factors Affecting the Accessibility of IT Artifacts: A Systematic Review. *Communications of the Association for Information Systems*, *51*, 666–702. https://doi.org/10.17705/1CAIS.05129

Maslow, A. H. (1943). Conflict, frustration, and the theory of threat. In *Contemporary Psychopathology: A Source Book* (pp. 588–594). Harvard University Press.

Maslow, A. H. (1958). *A Dynamic Theory of Human Motivation.*

Mason, R. O. (1988). Experimentation and knowledge: A pragmatic perspective. *Knowledge*, *10*(1), 3–24.

Metsämuuronen, J. (2000). *Laadullisen tutkimuksen perusteet*. Methelp.

Miranda, J., Cabral, J., Wagner, S. R., Fischer Pedersen, C., Ravelo, B., Memon, M., & Mathiesen, M. (2016). An open platform for seamless sensor support in healthcare for the internet of things. *Sensors*, *16*(12), 2089.

Moilanen, J., Niinioja, M., Seppänen, M., & Honkanen, M. (2018). *API Economy 101*. Alma Talent.

Myers, M. D. (1997). Qualitative Research in Information Systems. *MIS Quarterly*, *21*(2), 241–242. https://doi.org/10.2307/249422

Myers, M. D. (2019). Qualitative research in business and management. *Qualitative Research in Business and Management*, 1–364.

Nambisan, S., Siegel, D., & Kenney, M. (2018). On open innovation, platforms, and entrepreneurship. *Strategic Entrepreneurship Journal*, *12*(3), 354–368.

Nielsen, J. (1994). *Usability Engineering*.

Nielsen, J. (2012). *Usability 101: Introduction to usability*. https://www.nngroup.com/articles/usability-101-introduction-to-usability/

Nielsen, J., & Norman, D. (n.d.). *The Definition of User Experience (UX)*. Retrieved February 22, 2023, from https://www.nngroup.com/articles/definition-user-experience/

Nikravesh, A., Yao, H., Xu, S., Choffnes, D., & Mao, Z. M. (2015). Mobilyzer: An open platform for controllable mobile network measurements. *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, 389–404.

Norman, D. A. (1986). Cognitive engineering. *User Centered System Design*, *31*(61), 2.

Norman, D. A. (1999). *The Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution* (Vol. 1). The MIT Press. https://EconPapers.repec.org/RePEc:mtp:titles:0262640414

Nunes, B. A. A., Mendonca, M., Nguyen, X.-N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, *16*(3), 1617–1634.

Orlikowski, W. J., & Baroudi, J. J. (1991). Studying information technology in organizations: Research approaches and assumptions. *Information Systems Research*, *2*(1), 1–28.

Osagie, E., Waqar, M., Adebayo, S., Stasiewicz, A., Porwol, L., & Ojo, A. (2017). Usability evaluation of an open data platform. *Proceedings of the 18th Annual International Conference on Digital Government Research*, 495–504.

Parker, G. G., van Alstyne, M. W., & Choudary, S. P. (2016). *Platform revolution : how networked markets are transforming the economy - and how to make them work for you* (First Edition). W. W. Norton & Company.

Parment, A., Kotler, P., & Armstrong, G. (2016). *Principles of Marketing Scandinavian Edition*. Pearson.

Parton, J. (2021). *Developer Relations: The Developer Journey Map*. CodeX. https://medium.com/codex/developer-relations-the-developer-journey-map-36bd4619f5f3

Persson, H., Åhman, H., Yngling, A. A., & Gulliksen, J. (2015). Universal design, inclusive design, accessible design, design for all: different concepts—one goal? On the concept of accessibility—historical, methodological and philosophical aspects. *Universal Access in the Information Society*, *14*(4), 505–526. https://doi.org/10.1007/s10209-014-0358-z

Polese, M., Bonati, L., D'Oro, S., Basagni, S., & Melodia, T. (2022). ColO-RAN: Developing machine learning-based xApps for open RAN closed-loop control on programmable experimental platforms. *IEEE Transactions on Mobile Computing*.

Pönnelin, V., Karppinen, J., korkeakoulu, S., Jäntti, R., Aalto-yliopisto, & University, A. (2023). *Optimized messaging framework for distributed radio access network*. https://aaltodoc.aalto.fi/handle/123456789/120221

ProgrammableWeb Staff. (2022). *ProgrammableWeb is Being Retired*. ProgrammableWeb Blog. https://www.programmableweb.com/news/programmableweb-being-retired/brief/2022/10/10

PRWeb. (2023). *Aira Technologies and VMware, in Collaboration with Intel, Demonstrate World's first xApp to Dramatically Improve 5G MU-MIMO Efficiency and Throughput using AI*.

RapidAPI. (2020). *Developer Survey 2020 Report*. https://rapidapi.com/wp-content/uploads/2021/01/2020-2021-Developer-Survey-Report.pdf

RapidAPI. (2022). *2022 State of APIs*. https://stateofapis.com/

RapidAPI. (2023). *RapidAPI Hub*. https://website.rapidapi.com/products/hub/

Robertson, S., & Robertson, J. (2012). *Mastering the requirements process: Getting requirements right*. Addison-wesley.

Rogers, E. M., Singhal, A., & Quinlan, M. M. (2014). Diffusion of innovations. In *An integrated approach to communication theory and research* (pp. 432–448). Routledge.

Saari, E. (2021). *Hyväksymismallien rooli ohjelmistotestauksessa* [Bachelor's thesis, University of Vaasa]. Osuva.

Santos, W. (2017). *ProgrammableWeb API Directory Eclipses 17,000 as API Economy Continues Surge*. ProgrammableWeb Blog. https://www.programmableweb.com/news/programmableweb-api-directory-eclipses-17000-api-economy-continues-surge/research/2017/03/13

Santos, W. (2019). *APIs show Faster Growth Rate in 2019 than Previous Years*. ProgrammableWeb Blog. https://www.programmableweb.com/news/apis-show-faster-growth-rate-2019-previous-years/research/2019/07/17

Schindelin, J., Rueden, C. T., Hiner, M. C., & Eliceiri, K. W. (2015). The ImageJ ecosystem: An open platform for biomedical image analysis. *Molecular Reproduction and Development*, *82*(7–8), 518–529.

Schrepp, M., Hinderks, A., & Thomaschewski, J. (2017). Design and evaluation of a short version of the user experience questionnaire (UEQ-S). *International Journal of Interactive Multimedia and Artificial Intelligence, 4 (6), 103-108.*

Sharma, C. (2019). Concentrated digital markets, restrictive apis, and the fight for internet interoperability. *U. Mem. L. Rev.*, *50*, 441.

Silverman, D. (2006). *Interpreting qualitative data : methods for analyzing talk, text and interaction* (3rd ed). SAGE Publications. http://www.loc.gov/catdir/enhancements/fy0660/2006920650-d.html

Silverman, D. (2015). *Interpreting qualitative data*. Sage.

Simon, J. P. (2021). APIs, the glue under the hood. Looking for the "API economy." *Digital Policy, Regulation and Governance*, *23*(5), 489–508.

Singh, S. K., Singh, R., & Kumbhani, B. (2020). The evolution of radio access network towards open-ran: Challenges and opportunities. *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 1–6.

Smith, S. (2022). *Architecting Modern Web Applications with ASP.NET Core and Microsoft Azure*. Microsoft Developer Division, .NET, and Visual Studio product teams.

Svensson, C., Udesen, J., & Webb, J. (2019). Alliances in Financial Ecosystems: A Source of Organizational Legitimacy for Fintech Startups and Incumbents. *Technology Innovation Management Review*, *9*(1), 20–32. https://doi.org/10.22215/timreview/1209

Venkatesh, V., Thong, J. Y. L., & Xu, X. (2012). Consumer Acceptance and Use of Information Technology: Extending the Unified Theory of Acceptance and Use of Technology. *MIS Quarterly*, *36*(1), 157–178. https://doi.org/10.2307/41410412

Venkatesh, V., Thong, J. Y. L., & Xu, X. (2016). Unified theory of acceptance and use of technology: A synthesis and the road ahead. *Journal of the Association for Information Systems*, *17*(5), 328–376. https://doi.org/10.17705/1jais.00428

Wäljas, M., Segerstahl, K., Väänänen-Vainio-Mattila, K., & Oinas-Kukkonen, H. (2010). Cross-Platform Service User Experience: A Field Study and an Initial Framework. *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, 219–228. https://doi.org/10.1145/1851600.1851637

Wang, J. (2009). New media technology and new business models: Speculations on 'post-advertising'paradigms. *Media International Australia*, *133*(1), 110–119.

Wang, T., Zhu, H., Ruffy, F., Jin, X., Sivaraman, A., Ports, D. R. K., & Panda, A. (2020). Multitenancy for fast and programmable networks in the cloud. *USENIX Workshop on Hot Topics in Cloud Computing (HotCloud)*.

Web Accessibility Initiative. (2008). *Web Content Accessibility Guidelines (WCAG) 2.0*. https://www.w3.org/TR/WCAG20/

Web Accessibility Initiative. (2018). *Web Content Accessibility Guidelines (WCAG) 2.1*. https://www.w3.org/TR/WCAG21/

Wen, H., Porras, P., Yegneswaran, V., & Lin, Z. (2022). A Fine-Grained Telemetry Stream for Security Services in 5G Open Radio Access Networks. *Proceedings*

of the 1st International Workshop on Emerging Topics in Wireless, 18–23. https://doi.org/10.1145/3565474.3569070

Wypiór, D., Klinkowski, M., & Michalski, I. (2022). Open RAN—Radio Access Network Evolution, Benefits and Market Trends. *Applied Sciences*, *12*(1), 408. https://doi.org/10.3390/app12010408

Ye, Y., & Kishida, K. (2003). Toward an understanding of the motivation of open source software developers. *25th International Conference on Software Engineering, 2003. Proceedings.*, 419–429.

Yemini, Y., & Silva, S. da. (1996). Towards programmable networks. *In IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*.

Yin, R. K., & Campbell, D. T. (2003). *Case study research : design and methods* (3rd ed). Sage Publications.

Zachariadis, M., & Ozcan, P. (2017). *The API economy and digital transformation in financial services: The case of open banking*.

Appendixes

Annex 1. The key concepts of the thesis

The main terminology of this thesis is introduced here to make the topic easier to understand. Each term is going to be explained later in the thesis more thoroughly.

Human-Computer Interaction (HCI) is defined as *a field dealing with the design, assessment, and deployment of interactive computing systems for human use, as well as the study of key phenomena surrounding them* (Hewett et al., 1992).

Application programming interface (API) is "*a programming interface that allows different applications to share information, functionality, and other resources with each other*" (Moilanen et al., 2018, p. 211). The term API leads often to wrong expectations. APIs are confused with applications like Apps or often mislead to concern only integrated products while APIs can be different things (Moilanen et al., 2018, p. 19).

An open data interface is *an interface that is being used only for providing open data content in terms of use* (Moilanen et al., 2018, p. 213).

Open API is "*a classification that includes both public interfaces and partner interfaces*" (Moilanen et al., 2018, p. 213).

Artificial Intelligence (AI) "*allows machines, equipment, programs, systems, and services to function rationally according to mission and situation*" which "*requires a lot of data, computing power, and algorithms*" (Moilanen et al., 2018, p. 219).

Business model answers to at least the following questions: *who is the customer, what is the value proposition, and how value can be produced for the customer at an appropriate cost* (Moilanen et al., 2018, p. 216)?

Platform is "*a business model that uses technology to connect people, organizations, and resources in an interactive ecosystem in which amazing amounts of value can be created and exchanged*" (Parker et al., 2016, p. 3).

Digital platform is *a management system to control, interact and refine value from data and resources outside the platform* (Moilanen et al., 2018, p. 214).

Cloud platform provides "*data storage, software driving, machine learning, and artificial intelligence services*" (Moilanen et al., 2018, p. 218).

Platform economy is "*based on dynamic, multi-party innovation cooperation and marketplaces for the exchange of innovations, where it is possible to achieve large-scale and rapidly scaling network effects*" (Moilanen et al., 2018, p. 211).

Brand is "*the sum of characteristics of a product or service perceived by a user"* (Chaffey & Chadwick, 2022, p. 496).

Customer loyalty is "*the desire on the part of the customer to continue to do business with a given supplier over time*" (Chaffey & Chadwick, 2022, p. 500).

Value creation refers to *how an API provides value to consumers, owners, and other stakeholders* (Moilanen et al., 2018, p. 213).

Vision is *the long-term aim or the future target state where the company wants to aim someday.*

Mission is about *the ongoing principles that lead to the road to the mission*. Mission is about *why the company exists, what are the tasks and tools the organization's activities want to achieve and refers more to the reason for existence*.

Strategies are *the course of action*. When the company's vision and the purpose of the operation are known, the company's strategies can be defined. If the implementation of the strategy is *done successfully, the vision is achieved*.

Weakest link is "*the bottleneck of the chain*" as **"***the strength of the chain is determined by the weakest link*" (Goldratt & Cox, 2016, pp. 306–338).

Ecosystem consists of *a multi-layered set of partners, all of whom and their interaction are needed to materialize the ecosystem's central value proposition* (Moilanen et al., 2018, p. 214).

Digital marketing is "*achieving marketing objectives with the application of digital media, data, and technology*" (Chaffey & Chadwick, 2022, p. 502).

Network effects "*refers to the impact that the number of users of a platform has on the value created for each user*" (Parker et al., 2016, p. 17).

Hype cycle is "*a graphic representation of the maturity, adoption, and business application of specific technologies*" (Chaffey & Chadwick, 2022, p. 506).

Segmentation is "*identifying different user groups within a target market to develop differentiated offerings for each group*" (Chaffey & Chadwick, 2022, p. 516).

Paradigms are "*basic belief systems based on ontological, epistemological, and methodological assumptions*" (Guba & Lincoln, 1994, p. 107).

Concept is "*an idea deriving from a given model*" (Silverman, 2015, p. 53).

Ontology describes "*what reality is like and the basic elements it contains*" (Silverman, 2006, p. 13).

Epistemology describes "*what is the nature and status of knowledge*" (Silverman, 2006, p. 13).

Theory is "*a set of concepts used to define and/or explain some phenomenon*" (Silverman, 2015, p. 53).

Hypothesis is "*a testable proposition in research*" (Silverman, 2015, pp. 53–54).

Methodology outlines "*the choices we make about cases to study, methods of data gathering, forms of data analysis, etc. in planning and executing a research study*" (Silverman, 2015, p. 54).

Model is "*an overall framework for looking at reality*" (Silverman, 2015, p. 53).

Method is "*a specific research technique*" that should be a "*good fit with model, theory, hypothesis and methodology*" (Silverman, 2015, p. 53).

User experience (UX) is **"***concerned with developing high-quality interactive systems, products and services that fit with people and their ways of living*" while it **"***encompasses all the feelings, thoughts, sensations, and actions of engaging in some activity*" (Benyon, 2019, pp. 1–2).

Motivation is "*a psychological process where the individual wants to attain a goal based on their intensity, direction, and persistence*" (Latham & Pinder, 2005, p. 486).

Annex 2. Developer journey map

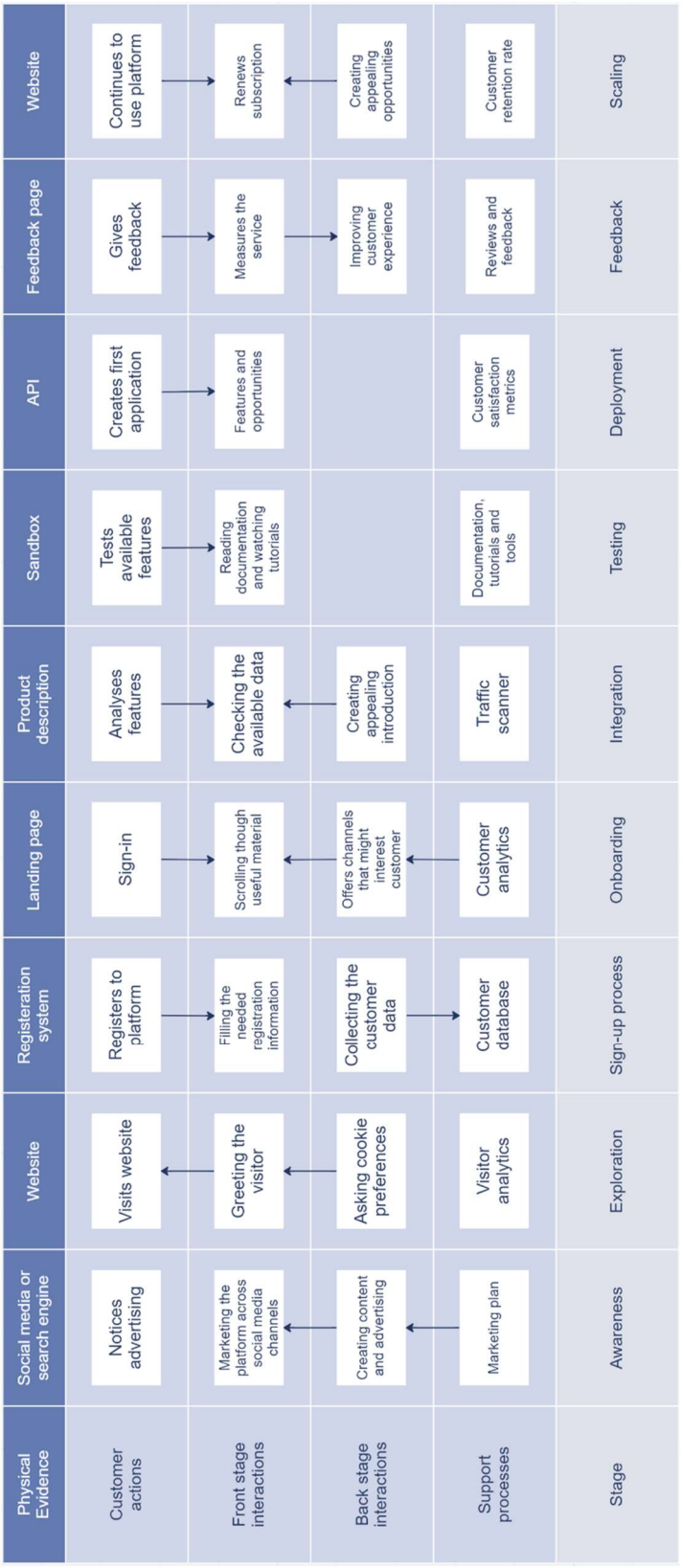
DEVELOPER JOURNEY MAP

Stages	Awareness	Exploration	Integration	Testing	Deployment	Feedback	Scaling
Goals/needs	Could this be useful for me?	Will it meet my needs?	How does it work?	How well does it work?	Can I build a proof of concept?	Was this worth trying?	Can I build to scale?
Questions	1. What is it? 2. Could it solve my problem? 3. Is it credible?	1. Is it useful? 2. Is it easy to use? 3. Is there already something that I do not like?	1. Could this work? 2. Do I have enough confidence? 3. Should I try this?	1. Is coding made easy and efficient? 2. Are the documents good enough? 3. Is there a community I can rely on?	1. Is the user experience good enough? 2. Do I get support? 3. Are my expectations met?	1. How do I give feedback? 2. Is this worth the money? 3. Am I satisfied?	1. How can I contribute? 2. Can I do more? 3. Will I keep using this?
Internal touchpoints	• Developer portal • Search engine results • Social media (YouTube, Facebook, Twitter, Instagram, Twitch, LinkedIn, Discord, Reddit etc.) • Events (face-to-face and virtual) • Blog • Newsletter • Case studies • Articles • Podcasts • Reviews	• FAQ • Q&A • Documentation • Use cases • Market place • Tech talks, webinars, shows and announcements	• Tutorials • Code samples • Cheat sheets • Guides • Solutions • Documentation • Whitepapers • Support desk	• Registration/sign up • Developer tools • Quick start guides • Documentation • Training • Learning store • Educator center • Learning paths	• Extensions • Guides • Sandbox • Changelog • Support • Workshops	• Developer success • Measuring the service (SLAs) • Showcase • Communities	• Subscription • Ambassador program • Partner program • Developer spotlights • Badges • Rewards • Certifications
External touchpoints	• Advertisements and news (digital and traditional marketing) • Word-of-mouth (discussions, events, meetups, referrals etc.) • Online and offline groups • Events, competitions and challenges (hackathons etc.)	• Communities (GitHub, Stack Overflow) • Reviews	• Technology dependencies • Articles	• Questions tagged (Stack Overflow) • Tutorials (YouTube)		• Reviews	• Alternative solutions

Annex 3. Benchmarking

Factors	COMPANIES			
	AWS	Google	RedHat	Windows
Supported front-end languages	JavaScript	JavaScript, TypeScript	JavaScript	Any programming language
Supported back-end languages	.NET, C++, Go, Java, Kotlin, PHP, Python, Ruby, Swift, Rust	C++, Go, Java, Kotlin, Python, Ruby, Swift, Node.js, CSS, Dart, HTML	.NET, C, C#, C++, Go, Kotlin, PHP, Python, Ruby, Node.js, Rust	Any programming language
Social channels	• Multiple podcasts (multiple music service providers) • Announcements, knowledge solutions and founders (YouTube) • Advice, news and announcements (Facebook) • Discussions (Twitch) • Community to help and gain knowledge (re:Post) • Podcasts (Soundcloud) • Questions tagged (Stack Overflow) • Discussions (Reddit) • Labs (GitHub) • Development focused content and resources (Twitter) • Reviews (Peer Insights, G2)	• Blog • Podcasts and shows (YouTube) • Advice and resources (Facebook) • Discussions and instructions (Twitter)	• Blog • Tech talks, weekly live chats and instructions (YouTube) • Latest news (Facebook) • Repositories and projects (GitHub) • Tutorials and career insights (Twitter)	• Blog • Shows and tutorials (YouTube) • Stories, advises and memes (Instagram) • Projects (GitHub) • Training resources, events and news (LinkedIn) • Advice, research and case studies (Twitter)
Engaging with other developers	• Articles • Communities • Events like community day, developer day and virtual tech talks • Learning by community (re:Skill) • Study groups • Connecting with experts	• Articles • Communities • Face-to-face events globally • Competitions • Challenges • Startup Accelerators	• Articles • Communities • Events like summits and developer days • Workshops and labs	• Startup Founders • Communities • Virtual events like virtual training days • Certified trainers • Learning partners • Connecting with likeminded people and startups with the same business idea • Community-led after parties
Useful material	• Learning store • Q&A and FAQs • Knowledge and support center • Analyst reports (re:Invent) • Tutorials, guides, whitepapers, videos and solutions • Architecture and diagram examples • Explanations of cloud computing, DevOps, a container and data lake etc. • Careers	• Learning store • Case studies	• Learning store • F&Qs • Cheat sheets • Developer Tools • E-books • Tutorials • Use cases • Sandbox • Live events to go through fundamentals • Learning paths	• Learning store • Q&A • Code samples • Educator center • Documentation • Developer support • Student Hub • Assessments
Qualifications	Certifications, Courses (Digital, classroom and partner), Rewards	Badges, Developer expert profiles	Certifications, Quizzes, Contests	Certifications, Student certifications, Courses

Annex 4. Service blueprint



Annex 5. Survey questions

Background check

First some basic questions

1. How many years have you worked in software development/testing?
   - 1—3
   - 4—6
   - 7—10
   - Over 11

2. Your role at your company
   - Software architect
   - Software developer intern
   - Software developer
   - Senior software developer
   - Software development team leader
   - Software development manager or similar
   - Role above software development manager
   - Software tester intern / Software QA engineer intern
   - Software tester / Software QA engineer
   - Senior software tester / Senior software QA engineer
   - Software testing team leader
   - Software testing manager or similar
   - Role above software testing manager

3. Industry or trade area
   - Agriculture, Forestry, and Fisheries
   - Manufacturing
   - Energy sector
   - Infrastructure

- Construction
- Wholesale and retail
- Transportation
- Information and communication
- Telecommunication
- Accommodation and catering
- Administration, law, and support services
- Financing and insurance
- Real estate
- Education
- Healthcare and social services
- Entertainment and recreation

4. Company size
- Micro - Less than 10 persons working and annual turnover less or equal to 2 M€
- Small - Less than 50 persons working and annual turnover less or equal to 10 M€
- Medium-sized - Less than 250 persons working and annual turnover less or equal to 50 M€
- Large - More or equal to 1000 persons working and annual turnover less or equal to 50 M€
- Multinational enterprise - More than 10 000 persons working and annual turnover less or equal to 1 000 M€

5. Your SW knowledge and testing domains

   Description: The user can choose multiple options
- Cloud
- Verification
- Radio technology
- IT administration / Operations
- Continuous integration / Continuous delivery

- System testing
- Embedded
- Network
- Machine learning
- Fastpath
- Databases
- Telecommunication
- Performance testing
- Functional testing
- Other (user can write)

6. Are you familiar with

   Description: The user can choose multiple options

- Kubernetes
- Container development
- Dev/Ops
- Agile process
- OpenStack
- GitOps
- SecOps
- No familiar options
- Other (user can write)

7. Do you have experience in xApps?

   Description: An xApp is a software tool used by a RAN Intelligent Controller (RIC) to manage network functions in near-real time.

- Yes
- No

Based on the answer to question 7 if the user chooses to answer

a) Yes, they will fill out question 8
b) No, move to answer questions 50–96

8. What is your primary role when working with xApps?
   Description: Your choice will affect the next set of questions.

- xApp testing
- xApp development

Based on the answer to question 8 if the user chooses to answer

a) xApp testing, the user answers first xApp testing-related question (questions 9–28) and then questions related to documentation (questions 66–96)
b) xApp development, the user answers first xApp development-related questions (questions 29–49) and then questions related to documentation (questions 66–96)

<u>xApp testing-related questions</u>

9. Please rate your current testing experience.
   Description: Linear scale from 1 to 7, where 1 is a beginner and 7 is expert

10. Attractiveness - How satisfied are you with your xApp testing experience?
    Description: Linear scale from 1 to 7, where 1 is dissatisfied and 7 is satisfied

11. Minimal action - How much time and effort did it take you to learn to use and test an xApp?
    Description: Linear scale from 1 to 7, where 1 is a lot and 7 is none

12. Consistency - From a satisfaction point of view, how clear and familiar is the xApp compared to other platforms or programs you have already used?

Description: Linear scale from 1 to 7, where 1 is a non-similar and 7 is clear and familiar

13. Self-descriptiveness - Is there enough help for you in the form of information, documentation, and tool tips available so that you do not have to contact the help desk or similar to test an xApp on the platform?
    Description: Linear scale from 1 to 7, where 1 is a disagree and 7 is agree

14. Accuracy - The xApp services should help the end customer to search the needed content and accomplish the desired goals. Accuracy is about decreasing the probability of the customer to complain accomplishing these goals. How the xApp service is filling this accuracy criteria?
    Description: Linear scale from 1 to 7, where 1 is not filling and 7 is filling

15. Readability - How readable or interpretable are the xApp service and the API documentation data tables and flow charts?
    Description: Linear scale from 1 to 7, where 1 is non-readable and 7 is readable

16. Simplicity - Are the xApp service and the API documentation data tables and charts simple and fast to read?
    Description: Linear scale from 1 to 7, where 1 is complicated and 7 is simple

17. Familiarity - How familiar are the provided xApp platform tools to the ones you have previously used?
    Description: Linear scale from 1 to 7, where 1 is non-familiar and 7 is familiar

18. Effectiveness - Are the xApps helping you to achieve your goals with accuracy and relevance?
    Description: Linear scale from 1 to 7, where 1 is not helping and 7 is helping

19. Satisfaction - Are you satisfied after working with xApps?
    Description: Linear scale from 1 to 7, where 1 is not satisfied and 7 is satisfied

20. Learnability - How quick and easy it was for you to adopt the xApp functionalities while achieving tasks compared to your previous experiences with information systems?
    Description: Linear scale from 1 to 7, where 1 is difficult and 7 is easy

21. Accessibility - How accessible is both information and the tools needed to test an xApp?
    Description: Linear scale from 1 to 7, where 1 is non-accessible and 7 is accessible

22. Do you think a third-party tester could test an xApp based on the provided documentation?
    Description: Linear scale from 1 to 7, where 1 is disagree and 7 is agree

23. How skilled should the tester be to test xApps based on the documentation?
    Description: Linear scale from 1 to 7, where 1 is beginner and 7 is expert

24. What experience do you have using the xApp application development environment in testing?

25. Please explain the challenges you have faced when testing xApps.

26. Which problem investigation tools have you usually used testing xApps for analysis and log collection?
    Description: User can choose multiple options. Columns mostly used, often used, never used, and wanted in xApp testing

- ss/netstat
- nc/ncat/netcat

- ifconfig/ip
- nsenter
- mtr
- tcpdump/Wireshark
- dcap
- syslog/BTSLog
- emil log/cell trace

27. Is there some testing tool you have used and would like to mention besides previously stated?
- No
- Other (user can write)

28. What capabilities in testing tools are important to you?

xApp development related questions

29. Please rate your current developer experience.
    Description: Linear scale from 1 to 7, where 1 is beginner and 7 is expert

30. Attractiveness - How satisfied are you with your xApp development experience?
    Description: Linear scale from 1 to 7, where 1 is dissatisfied and 7 is satisfied

31. Minimal action - How much time and effort did it take you to learn to use SEP - Service Enablement Platform?
    Description: Linear scale from 1 to 7, where 1 is a lot and 7 is none

32. Minimal action - How much time and effort did it take you to develop an xApp running in SEP?
    Description: Linear scale from 1 to 7, where 1 is a lot and 7 is none

33. Consistency - From satisfaction point of view, how clear and familiar is xApp developing compared to other platforms or programs you have already used?
    Description: Linear scale from 1 to 7, where 1 is a non-similar and 7 is clear and familiar

34. Self-descriptiveness - Is there enough help for you in the form of information, documentation, and tool-tips available so that you do not have to contact the help desk or similar to develop an xApp on the platform?
    Description: Linear scale from 1 to 7, where 1 is a disagree and 7 is agree

35. Accuracy - The SEP application services, and xApp-frame should help the end customer to search the needed content and build the desired use case and give the ability to develop it. Accuracy is about decreasing the probability of the customer to fail accomplishing the target. How do the SEP application services and xApp-frame fill this accuracy criteria?
    Description: Linear scale from 1 to 7, where 1 is not filling and 7 is filling

36. Readability - How readable or interpretable are the xApp service and the API documentation data tables and flow charts?
    Description: Linear scale from 1 to 7, where 1 is non-readable and 7 is readable

37. Simplicity - Are the xApp service and the API documentation data tables and charts simple and fast to read?
    Description: Linear scale from 1 to 7, where 1 is complicated and 7 is simple

38. Familiarity - How familiar are the provided xApp platform tools to the ones you have previously used?
    Description: Linear scale from 1 to 7, where 1 is non-familiar and 7 is familiar

39. Effectiveness - Are the xApp APIs helping you to achieve your goals with accuracy and relevance?
    Description: Linear scale from 1 to 7, where 1 is not helping and 7 is helping

40. Satisfaction - Are you satisfied after working with xApp API?
    Description: Linear scale from 1 to 7, where 1 is not satisfied and 7 is satisfied

41. Learnability - How quick and easy it was for you to adopt the xApp APIs and the service functionalities while achieving tasks compared to your previous experiences with information systems?
    Description: Linear scale from 1 to 7, where 1 is difficult and 7 is easy

42. Accessibility - How accessible is both information and the tools needed to create an xApp?
    Description: Linear scale from 1 to 7, where 1 is non-accessible and 7 is accessible

43. Do you think a third-party developer could implement an xApp based on the provided documentation?
    Description: Linear scale from 1 to 7, where 1 is disagree and 7 is agree

44. How skilled should the developer be to implement xApps based on the documentation?
    Description: Linear scale from 1 to 7, where 1 is beginner and 7 is expert

45. What experience do you have using the xApp application development environment?

46. Please explain the challenges you have faced when developing xApps.

47. Which tools have you previously used developing xApps and what tools would you like to have?

    Description: User can choose multiple options. Columns mostly used, often used, never used, and wanted in xApp development

- Software integrated development editor (IDE)
- Running the application instantly to test the implementation changes
- Running unit tests to validate the implemented test case
- Networking investigation tools to trace the IP traffic and network statistics
- Traffic simulator tools to test the application use case
- Profiler to investigate the program execution
- Debugger to race and monitor the execution of the application
- Memory tracking investigation tools to track resource usage

48. Is there some developing tool you have used and would like to mention besides previously stated?

- No
- Other (user can write)

49. What capabilities in development tools are important to you?

<u>Users previous experience</u>

50. Please rate your current testing experience.

    Description: Linear scale from 1 to 7, where 1 is beginner and 7 is expert

51. Please rate your current developer experience.

    Description: Linear scale from 1 to 7, where 1 is beginner and 7 is expert

52. Which problem investigation tools do you usually use when testing in analysis and log collection?

Description: User can choose multiple options. Columns mostly used, often used, and never used

- ss/netstat
- nc/ncat/netcat
- ifconfig/ip
- nsenter
- mtr
- tcpdump/Wireshark
- syslog
- Jenkins

53. Is there some testing tool you have used and would like to mention besides previously stated?
- No
- Other (user can write)

54. Which tools are you able to use when developing?
Description: User can choose multiple options. Columns mostly used, often used, and never used
- Running the application instantly to test the implementation changes
- Running unit tests to validate the implemented test case
- Networking investigation tools to trace the IP traffic and network statistics
- Traffic simulator tools to test the application use case
- Profiler to investigate the program execution
- Debugger to race and monitor the execution of the application
- Memory tracking investigation tools to track resource usage

55. Is there some developing tool you have used and would like to mention besides previously stated?
- No

- Other (user can write)

56. The best tools that I have used when creating SW are…
    Description: Complete the sentence

57. The most important thing to me when I debug is to…
    Description: Complete the sentence

58. The general goals that I am trying to solve with my application…
    Description: Complete the sentence

Thoughts about the idea

59. What is your first impression?
    Description: Linear scale from 1 to 7, where 1 is bad and 7 is good

60. Does this compare to your expectations?
    Description: Linear scale from 1 to 7, where 1 is does not fill expectations and 7 is meets expectations

61. How much do you think xApp development environment is going to help you?
    Description: Linear scale from 1 to 7, where 1 is no help and 7 is lots of help

62. Please tell why you think xApp development environment is going/not going to help you.

63. Does this feel like it was designed for you?
    Description: Linear scale from 1 to 7, where 1 is a disagree and 7 is agree

64. Please tell why you think xApp development environment is designed/not designed for you.

65. How motivated would you be to develop/test an xApp?
    Description: Linear scale from 1 to 7, where 1 is a not motivated and 7 is highly motivated

Documentation

66. Have you taken a good look at the documentation now?
- Yes
- No

Users can express their agreement with the attributes by ticking the circle that most closely reflects their impression. Users are asked to decide their choices spontaneously.

What are your thoughts about the current documentation?

67. Description: Linear scale from 1 to 7, where 1 is obstructive and 7 is supportive
68. Description: Linear scale from 1 to 7, where 1 is complicated and 7 is easy
69. Description: Linear scale from 1 to 7, where 1 is inefficient and 7 is efficient
70. Description: Linear scale from 1 to 7, where 1 is confusing and 7 is clear
71. Description: Linear scale from 1 to 7, where 1 is boring and 7 is exciting
72. Description: Linear scale from 1 to 7, where 1 is not interesting and 7 is interesting
73. Description: Linear scale from 1 to 7, where 1 is conventional and 7 is inventive
74. Description: Linear scale from 1 to 7, where 1 is usual and 7 is leading edge

75. Attractiveness - How satisfied are you with the documentation?
    Description: Linear scale from 1 to 7, where 1 is dissatisfied and 7 is satisfied

76. Minimal action - How much time and effort do you think it would take for you to learn to use the documentation?
    Description: Linear scale from 1 to 7, where 1 is a lot and 7 is none

77. Consistency - From a satisfaction point of view, how clear and familiar is the documentation compared to other documentations you have seen and used?
    Description: Linear scale from 1 to 7, where 1 is non-similar and 7 is clear and familiar

78. Self-descriptiveness - How much improvement does the documentation need?
    Description: Linear scale from 1 to 7, where 1 is a lot and 7 is none

79. Accuracy - The documentation should help the end customer to search for the needed content and accomplish the desired goals. Accuracy is about decreasing the probability of the customer to complain accomplishing these goals. How is the documentation filling this accuracy criteria?
    Description: Linear scale from 1 to 7, where 1 is not filling and 7 is filling

80. Readability - How readable or interpretable documentation datasets are?
    Description: Linear scale from 1 to 7, where 1 is non-readable and 7 is readable

81. Simplicity - Are the documentation data tables and charts simple and fast to read?
    Description: Linear scale from 1 to 7, where 1 is complicated and 7 is simple

82. Familiarity - How familiar are the provided documentation tools to the ones you have previously used?
    Description: Linear scale from 1 to 7, where 1 is non-familiar and 7 is familiar

83. Effectiveness - Is the documentation helping you to achieve your goals with accuracy and relevance?

Description: Linear scale from 1 to 7, where 1 is not helping and 7 is helping

84. Satisfaction - Are you satisfied with the documentation?
    Description: Linear scale from 1 to 7, where 1 is not satisfied and 7 is satisfied

85. Learnability - How quick and easy it would be for you to adopt the documentation while achieving tasks compared to your previous experiences with documentation?
    Description: Linear scale from 1 to 7, where 1 is difficult and 7 is easy

86. Accessibility - How accessible are both information and the tools in the documentation?
    Description: Linear scale from 1 to 7, where 1 is non-accessible and 7 is accessible

87. What do you think should be added/improved in the documentation?

88. Indicate the xApp API abstraction level.
    Description: Linear scale from 1 to 7, where 1 is not understandable and 7 is understandable

89. Would you like an API for use cases?
    Description: Linear scale from 1 to 7, where 1 is disagree and 7 is agree

90. Do you think you could develop/test xApp based on this documentation?
    Description: Linear scale from 1 to 7, where 1 is disagree and 7 is agree

91. Does the documentation describe the application use case details at such a level that the functionality is understandable?
    Description: Linear scale from 1 to 7, where 1 is not understandable and 7 is understandable

92. Please tell why you think application use case details are understandable/not understandable.

93. Based on this documentation are you able to understand what logs need to be taken for problem solving?
    Description: Linear scale from 1 to 7, where 1 is not understandable and 7 is understandable

Finish line

94. Which of these channels/provided material would you visit if they were provided by a developer portal?
    Description: Pick at least three most important for you

- Learning store
- Learning paths
- Blogs
- Podcasts
- Shows
- Stories
- Tutorials
- Reports
- Articles
- Use cases
- Code samples
- Cheat sheets
- Sandbox/Test environment
- Case studies
- E-books
- Q&A, FAQs

- Reviews
- Challenges
- Quizzes
- Competitions
- Virtual events
- Face-to-face events
- Workshops and labs
- Certifications
- Courses
- Badges
- Rewards
- Career possibilities
- Game to develop coding skills

95. Which social channels would you visit if they were provided by a developer portal?

    Description: Pick at least three most important for you

- YouTube
- Facebook
- Instagram
- Twitch
- Twitter
- LinkedIn
- GitHub
- Discord
- Stack Overflow
- Reddit
- Soundcloud, Spotify, etc.

96. How was the survey?

    Description: Linear scale from 1 to 7, where 1 is bad and 7 is good

Annex 6. Interview questions

1. In which channels do you seek or come across information related to new technologies and opportunities?
    a. Television
    b. Social media
    c. Research articles
    d. Newspapers
    e. Blog
    f. Magazines
    g. Influencers
    h. Events
    i. Networking
    j. Communities
    k. YouTube
    l. Radio
    m. Podcasts
    n. Word of mouth
    o. Or other

2. What are the critical factors when you decide to use an open application platform? Can you explain and walk through this process?

3. From where and how do you measure the end customer satisfaction against what you develop? Do you have for example customer key performance indicators (KPIs), CVPI, etc.?

4. What is the overall score you would give for the open-ecosystem idea from 1–7? What factors do you emphasize in this assessment? Can you give an example of a portal or developer community that has provided useful support for the developer journey – content, user-friendliness, or something like that?

5. What kind of other competing solutions have you used before? What are their pros and cons? What factors do you emphasize?

6. In general, what are the characteristics of a good partner that is assisting your developer/testing efforts? On the other hand, what would be the characteristics of a bad partner?