Tampere University

Ashish Sonawane

SOFTWARE-PLATFORM BASED ECOSYSTEM IN HEAVY DUTY MOBILE MACHINE INDUSTRY

“A case study on ROS Ecosystem”

Master of Science Thesis
Faculty of Management and Business
Project Manager. Jukka Yrjänäinen
Associate Prof. Reza Ghabcheloo
Professor. Tuomas Ahola
October 2022

ABSTRACT

Ashish Sonawane: Software-Platform based Ecosystem in Heavy duty Mobile Machine Industry: a case study on ROS Ecosystem
Master of Science Thesis
Tampere University
Master's Program in Industrial Engineering and Management, Business and Technology
October 2022

The platform-based ecosystem theory is continuously evolving with a higher level of interdependence and interconnectedness in dynamic business surroundings. In the software context, the platform-based ecosystem provides a modular architecture that allows reusability of the core functionalities across different applications. The software-platform based ecosystem could make a huge difference in the heavy-duty mobile machine industry by reducing the R&D efforts in developing complex software systems to achieve smart functionalities in the mobile machines. The objective of the study is to determine the significance of the software platform ecosystem in the heavy-duty mobile machine industry and whether it could provide new prospects to this industry.

The research explores the Robot Operating System (ROS) ecosystem to address the ecosystem opportunities in the heavy-duty mobile machine industry. The ROS ecosystem is an open -source software platform offering a core set of software development kits for developing robotic applications. ROS has become a de facto middleware in robotics providing numerous software packages, algorithms, drivers, and a diverse community of developers.

The research utilized a qualitative case study approach to investigate the heavy-duty mobile machine sectors' perspectives on Software-platform based ecosystems. A total of 12 interviewees participated, involving 5 from software providers/consultants, 3 from embedded system providers, and 4 from manufacturing organizations expressing their opinions and current understanding of the Software-platform based ecosystem. The interviews were focused on understanding the use of software platforms and ecosystems in the heavy-duty mobile machine industry. The findings suggested the interest of the organizations into the ROS ecosystem. Additionally, the use of software platforms indicated reducing the complexity of developing complex software applications required for mobile work machines.

This study contributes to the software platform and software ecosystem literature by providing the possibility to collaborate across the players in the ecosystem and pursuing the integration benefits of the Software-platform based ecosystems in the heavy-duty mobile machine industry. Finally, this thesis proposes a few future research directions that can expand the understanding and applications of ROS ecosystem in heavy-duty mobile machine industry.

Keywords: Platform-based Ecosystem, Software Platforms, Software Ecosystem, ROS Ecosystem, Heavy-duty Mobile Machine Industry

PREFACE

This thesis is a part of the research group at Tampere university, which specializes in identifying and tackling the challenges in the software development of Intelligent Mobile machines. It has been an honor and a privilege to participate to the research study on understanding platform-based ecosystems and their potential in the heavy-duty mobile machine industry. Working on this thesis has allowed me to improve my problem-solving abilities and widen my horizons of accomplishing impossible goals.

I would like to thank Jukka Yrjänäinen, Professor Reza Ghabcheloo and Tuomas Ahola for supervising and continuously mentoring throughout the thesis work. I would also like to specially thank Jukka for his constant support, understanding, and motivation to write and complete my thesis work. In addition, I would like to thank the PEAMS research group members for supporting me, especially over the lunch and coffee breaks, and for tormenting to finish the thesis.

Finally, I want to express my gratitude to my entire family for believing in me and supporting me during my master's studies. Special thanks to my friend Prabhat Kiran Thakuri for always inspiring and advising me during the thesis writing process.

Tampere, October 2022

Ashish Sonawane

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AGV	Automated Guided Vehicle
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
COTS	Components off the Shelf
DDS	Data Distribution Service
FIMA	Forum of Intelligent Machines
IoT	Internet of things
OSRF	Open-Source Robotics Foundation
PEAMS	Platform Economy for Autonomous Mobile Machines Software Development
PLC	Programmable Logic Controller
ROS	Robot Operating System
SDK	Software Development Kits
SECO	Software Ecosystem
SW	Software
TSC	Technical Steering Committee

1. INTRODUCTION

In today's constantly evolving technological world, the platform-based ecosystem has restructured multiple major industries, transforming the labor and economic value generation dynamics [31]. Historically, humans have operated industrial machinery while maintaining constant supervision and oversight of their operations. However, with the growth of software and smart technology systems, equipment is being transformed into an intelligent machines, enhancing productivity and efficiency in a wide range of applications [3]. Especially the idea of self-driving and self-operating is no longer a vision but rather a reality due to the advancement of technology and research done by universities and organizations [30].

Software plays an essential role in the transition toward automation and autonomous industry [54]. 'Software platforms are called the powerful engines of change' due to the flexibility of code and of the essential functions they offer in revamping productivity and transformation across many industries [16]. The emergence of platforms has been observed in a variety of other domains such as autonomous vehicles, robotics, drones, critical systems (such as smart grids, power plants, and military applications), and Internet of things (IoT) [10]. Software Platforms not only accelerate technological innovation but have also been the source of creating new business opportunities [72]. For instance, the transaction industry has been transformed completely into digital platforms, creating new business opportunities [35].

Industries are now going for an ecosystem approach due to the lack of competencies, heavy R&D investments, and need the for constant customization [5]. The ecosystem method incorporates a number of participants who work together on innovation or development, including developers, partners, and users. While software ecosystems are gaining traction, their increasing complexity makes it challenging for potential users to envision and appreciate the benefits derived from an ecosystem [67]. Among different user groups of software ecosystems, this study attempts to focus on the heavy-duty mobile machine industry and intends to understand the significance of the platform-based ecosystem to them.

Heavy duty mobile machinery industry refers to the non-road mobile machinery specifically designed for use in various applications such as forestry, mining, warehousing, and construction [42]. The transition of industrial equipment manufacturers toward automation has been observed as a result of technological advancement and innovation [62]. Though these transitions offer significant advantages such as efficiency, productivity, safety, and profitability [50, 60], they do necessitate complex system engineering as well as extensive hardware and software architectural integration [27, 37]. The thesis explores into the Software-platform based ecosystem in an effort to better understand the software ecosystem interaction capabilities in the heavy-duty mobile machinery industry.

1.1 Research Context

The study was conducted as a part of the Platform Economy for Autonomous Mobile Machines (PEAMS) project, a Business Finland funded research consortium. The PEAMS project is a collaboration with Tampere University and the University of Helsinki, and it is coordinated by the Forum for Intelligent Machines (FIMA) [74]. The FIMA group is an industry-driven association that promotes the collaboration of mobile work machine manufacturers, specialist companies, system integrators, and research institutes in the Finnish region. The FIMA organization supports and defines research and product development in this industry, in accordance with industry expectations [17].

The PEAMS project goal is to do research on some of the most critical software development challenges for Heavy duty mobility machines. Software development is quite complex and needs to be compatible with other equipment or subsystems. One component of the research is to comprehend the relevance of a platform-based ecosystem in the heavy-duty machine industry. The core idea is to study the Robot Operating System (ROS) ecosystem and the opportunities it may offer to the heavy-duty machine industry.

1.2 Research Problems & Questions

The research problem of the thesis is stated as follows:

> *What potential value could a platform-based software ecosystem will bring to Heavy duty mobile machine industry?*

To address the research problem more comprehensively, the following research questions are formulated:

- **RQ.1**: *What do industries think about platform-based ecosystem, and ROS ecosystem in particular?*
- **RQ.2:** *What opportunities can the companies realize through the ROS ecosystem?*

The primary goal of this thesis is to understand the benefits and significance of using a software ecosystem and the related common platform for the heavy-duty machine industry. The research seeks to understand the ROS ecosystem and its existing state of knowledge in the heavy-duty mobile machine industry. Furthermore, the research tries to explore in depth the company perspectives and opinions on adopting software platforms and ecosystems related to the heavy-duty mobile machine industry.

1.3 Structure of Thesis

The thesis is organized into seven chapters. The first chapter describes the study's background, objectives, and research questions. The literature review is addressed in Chapters 2,3 and 4 by outlining the fundamental principles of the research study. First, an introduction to the heavy-duty mobile machine business is provided, followed by a review of the literature on software platforms and ecosystems, and lastly, a brief examination of the robot operating system is presented. The research strategy for reviewing literature and acquiring empirical data is described in Chapter 5. The outcomes of the qualitative interviews are presented and analyzed in Chapter 6. Finally, the conclusion chapter summarizes the most important findings and makes an attempt to answer the research questions in light of the study themes. Additionally, it further covers the theoretical and practical contributions to the literature and finally presents the limitations and future perspectives on research directions. Figure 1 summarizes the complete structure of the thesis.

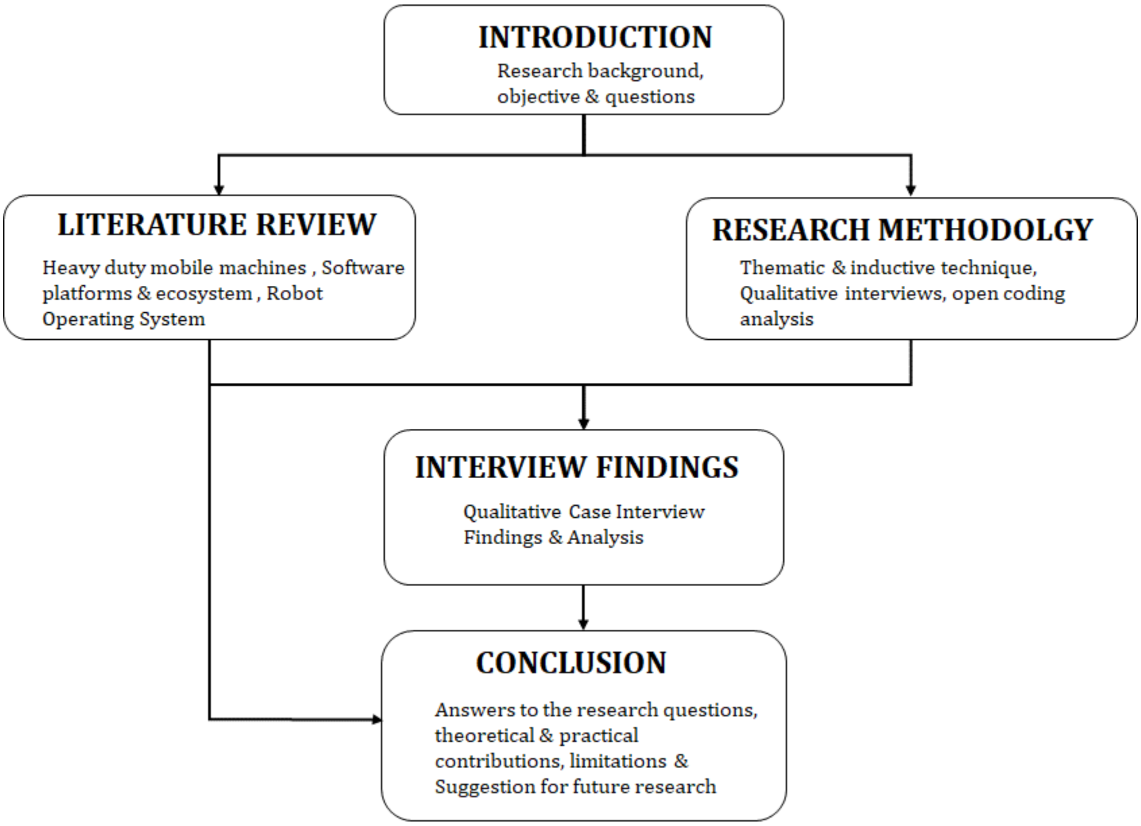

Figure 1. Thesis Structure.

2. HEAVY DUTY MOBILE MACHINE INDUSTRY

The use of heavy-duty mobile machinery is prominent in the material handling industry [20]. Heavy duty machines may not always be large and heavy, but rather specially designed machines for managing heavier loads and operating in difficult conditions. The term mobile here refers to any piece of equipment that is not intended for carrying passengers and may be drawn or driven in diverse terrains [81]. These involve a wide range of machinery used in diverse sectors such as construction, mining, agriculture, forestry, and warehouse. The most common example of heavy-duty mobile machines includes excavators, forklifts, loaders, forwarders, and dozers. Different heavy-duty machines serve different purposes based on the activities they are designed to perform. For example, loaders are used for earthmoving, road construction, agriculture, and also in large and small-scale industries [14]. It is used for moving heavy materials such as asphalt, demolition debris, snow, gravel, containers, and pallets.

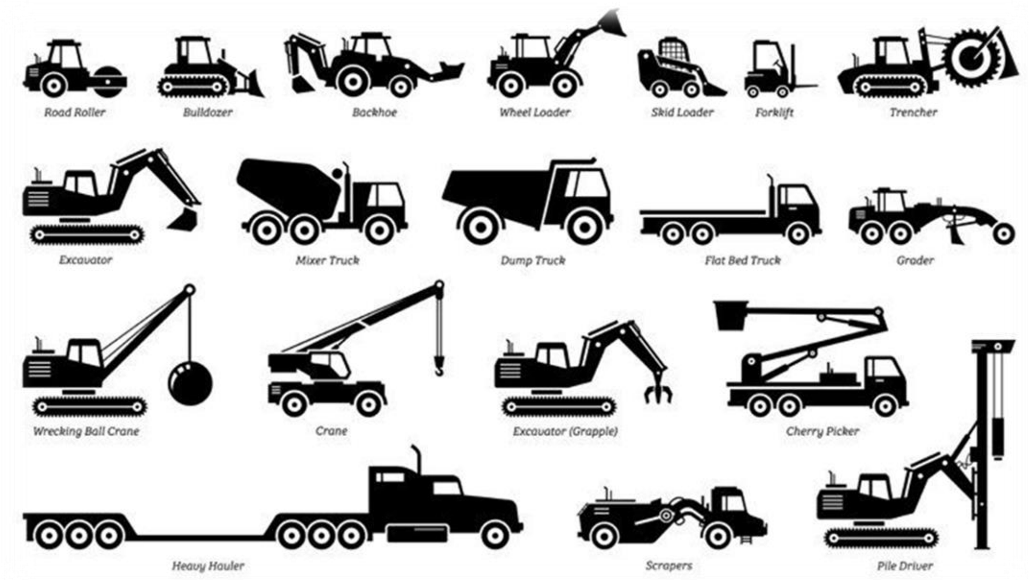

Figure 2. Different types of heavy-duty mobile machines [73].

The Figure 2 depicts the several heavy machinery types utilized in different contexts. Heavy duty machines were traditionally built entirely of mechanical components such as ropes, pulleys, and chains [25]. However, with the continuous evolution of technology and innovation, heavy-duty machines are completely revolutionized, enhancing their efficiency, agility, and productivity [25, 50]. For instance, the benefits of utilizing hydraulics and steering mechanisms have improved the machines' lifting capability and efficiency [20].

The use of software technology has increased over the functioning of mobile working machines [20]. The employment of more powerful and intelligent onboard computers in these machines has modified their manner of working, improved efficiency, and allowed them to perform more intelligently [47]. But there are high risks that come along with these machines. Especially driving these machines demands highly competent operators because of the significant risk of injury and accidents [39]. To address issues like as skilled labor shortages, safety, and hostile environments, the industry has shifted its focus to the development of automated and autonomous heavy-duty mobile machinery [41, 70].

Automation and autonomous operations in heavy-duty machines involve extensive usage of onboard software and electronic control systems performing various roles. Modern machines are equipped with intelligent embedded systems that are controlled by engineering control units such as sensors, and actuators. Autonomous basically mean that heavy-duty machines are able to operate without active human intervention [41]. The literature states that autonomous machines involve the complex mechanism of system engineerings such as sensing, perception & planning, edge & cloud computing, and mechanical control [37]. Additionally, the equipment must have a control system that can comprehend, analyze, and take relevant actions based on the data. However, it is important to acknowledge that a single heavy-duty machine manufacturer may not have all the competencies to develop these capabilities by itself.

The heavy-duty mobile machine industry includes not just manufacturers, but also organizations that develop the software and hardware needed in the machines. For instance, a wheel loader comes with a physical body, combinations of robotic mechanisms powered by hydraulic systems, and complex software control systems to perform multiple functionalities [14]. In a way, different players with specific competencies are involved in manufacturing a mobile machine. For example, a company with software development expertise or a sensor manufacturer. Figure 3 depicts the significant players that are associated with the heavy-duty mobile machine industry.

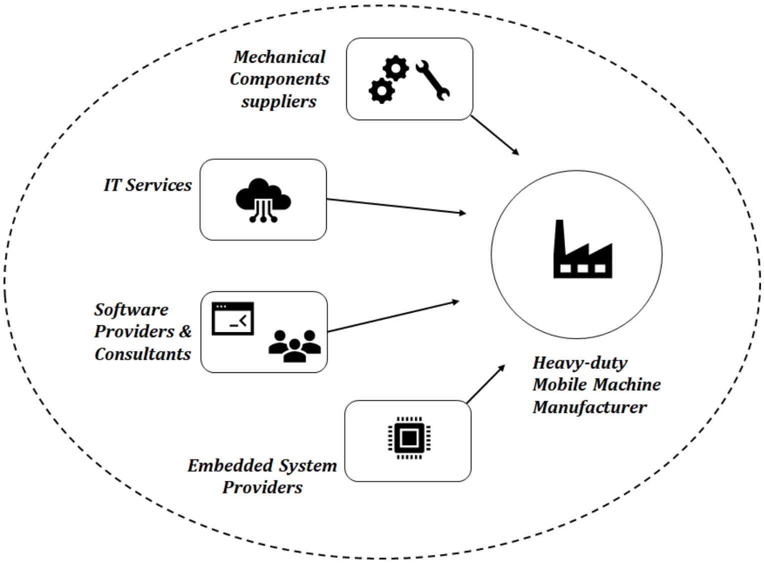

Figure 3. Major Companies associated with Heavy duty mobile machine industry.

The role of software and electronics is increasing with the complexity of achieving smart solutions [61]. In order to overcome software complexities and minimize software development efforts, industries are embracing software ecosystems [5, 55]. For example, Automotive Open System Architecture (AUTOSAR), a standardized software architecture developed to handle the software complexity in the automotive industry, uses a software ecosystem approach [33]. Similarly, robotics industries use the ecosystem approach like the ROS ecosystem to overcome the software complexities [13]. It is imperative to explore whether software ecosystems could have similar implications in the heavy-duty mobile machine industry.

3. PLATFORM BASED ECOSYSTEM

Globalization and digitization have accelerated the emergence of platforms that enable knowledge and resource sharing between diverse companies across similar or related industries [29]. As the name implies, a platform-based ecosystem constitutes two distinct concepts- namely, the platform and the ecosystem. Both the platform and ecosystem can be associated with different products or industries. However, this study particularly focuses on understanding the software platform and software ecosystem.

3.1 Software Platforms

In today's fast-paced and modernized world, the majority of products or applications are based on software platforms [16]. For the past few decades, " software platforms have been the invisible engines that have established, influenced or revolutionized practically every significant industry" [16]. From credit cards to mobile phones to games to vehicles, software platforms are utilized everywhere. Software platforms are not just used for commercial purposes; they are also used in research applications to facilitate software development and provide a stable environment for experimentation and rapid prototyping [27]. Software platforms provide the necessary configuration and a simplified environment for end users to develop their own applications [5].

The literature characterizes software platforms in a variety of ways, but the simplest definition is that they are a type of standard coded architecture on top of which you may construct various products or applications that do various tasks. According to Pohl et al. [52], a software platform is a collection of software subsystems and interfaces that constitute a common structure from which a range of derivative products may be developed and produced efficiently. Bosch [6] stated platform as an entity combines all functionality that is shared by all products or applications and often contains a standardized architecture. According to Tiwana et al. [63], a software platform is the expandable codebase of a software system that offers fundamental functionality shared by modules that communicate with one another via interfaces.

In the present era, software platforms appears to be the standard way of working in the software industry [57]. According to the literature, a software platform serves as a foundation for developing applications or technologies on top of a hardware and software framework [16]. There could be different types of software platforms such as utility platforms, technology and commercial platforms [80]. For instance, apple iOS or android are the commercial platforms that enable the interaction between the users and developers. Then there is the Application Programming Interface (API) development platform that provides the ability to interact with other software. Operating systems such as Mac OS, or Linux are also a kind of platform that supports different functions and applications [16]. Middleware is kind of a software that basically to offer services to application developers to control the hardware. Schmidt and Buschman [59], have highlighted how middleware and software frameworks complement one other and assist in increasing software reuse and quality. To be more specific, a software framework is an installation that streamlines application development by outlining a common environment across a group of software programs [27]. It is often distributed as source code with documentation.

The software functionality includes complex algorithms with a critical software application that needs to be continuously updated, reconfigured, and released [10]. The software platform can be seen as a bundle of functions that provides the framework and supporting tools for building applications [27]. Organizations typically build their own platforms or utilize an external platform if one that matches their needs is available. It has been noted the potential benefits that a platform may provide, such as cost savings, development efficiency through the reuse of common elements, and customization [79].

Another significant factor to look about software platforms is whether they are open or closed [16]. Open platforms are like operating systems for instance, Linux, where anybody can obtain its services to write applications that can support under the defined terms or license. Whereas only those who have the authorization to utilize a closed or proprietary platform may benefit from its services.

3.2 Software Ecosystem

Software Ecosystem (SECO) has been a long-running and rapidly evolving concept in the world of software engineering [28, 44]. Several recent studies have proposed software ecosystems (SECO) as a viable approach to build large software systems on top of software platforms by combining components developed by internal and external actors [44]. In the software business, a platform is typically found at the heart of an ecosystem [26], which consists of either a partial or full software product or a service upon which other providers might construct complementary applications and services. Many businesses, for example, base their software products or applications on the Linux or Windows platform. With the adaptation of platform evolution, organizations are observed shifting towards software ecosystem models, making it more effective to develop software products and platforms outside of an organization's conventional limits [10].

The software ecosystem is not just limited to the platform, but it involves the collaboration of different players, organizations, and individuals that might belong to different industries. Software ecosystem (SECO) is not a new term; the literature has characterized it in a variety of ways. Messerschmitt and Szypersky [48], initially introduced the concept of SECO in their book on software ecosystems. Table 1 summarizes the definitions of software ecosystems as they have evolved throughout time. Manikas and Hansen [44], identified three common elements present in SECO definitions in their literature: connecting relationships, business, and common software [44]. The relationship is seen between the different players that are a part of the ecosystem. The most common actors that are encountered in the ecosystem are orchestrators, niche players, External actors, vendors, and end users. The business aspects not only comprise financial benefits but also non-financial ones [44]. The software part is seen in different forms such as a collection of software projects, a set of software solutions, or as platforms.

This thesis focuses on the most recent definition of SECO put out by Manikas, which encompasses relationships between actors and software in association with software platforms that have an impact on ecosystems [43].

Table 1. Software Ecosystem definitions.

Definition	Author
"Traditionally, a software ecosystem refers to a collection of software products that have some given degree of symbiotic relationships."	[48]
"A software ecosystem is a collection of software projects which are developed, and which co-evolve together in the same environment"	[38]
"A software ecosystem consists of the set of software solutions that enable, support and automate the activities and transactions by the actors in the associated social or business ecosystem and the organizations that provide these solutions"	[5]
"We define a software ecosystem as a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them"	[26]

“A software ecosystem consists of a software platform, a set of internal and external developers and a community of domain experts in service to a community of users that compose relevant solution elements to satisfy their needs”	[7]
“The interaction of a set of actors on top of a common technological platform that results in a number of software solutions or services”	[44]
“The software and actor interaction in relation to a common technological infrastructure, that results in a set of contributions and influences directly or indirectly the ecosystem”	[43]

Kondel and Manikas *[32]*, emphasized the necessity for ecosystem typification based on the variations in the software platform *[32]*. They discovered four distinct ecosystem categories. The *cornerstone* ecosystem includes a typical type of software platform whose functionality is extended by the contributions of the actors. Here, a single significant player usually dominates the platform. The majority of research on this sort of ecosystems, such as the Android or iOS, is available in the literature. The next is the *standard-based* ecosystems, where the platform is replaced by standard specifications describing the requirement for actor contributions. A consortium with memberships maintains the ecosystem of this type. The *protocol-based* ecosystem is kind or more flexible and less constrained as compared to the standard-based ecosystem. Finally, *infrastructure-based* ecosystems share a similar technological environment, allowing players to contribute independently.*[32]*.

3.3 Relationships in an Ecosystem

Relationships and interactions between the many actors in an ecosystem are critical for the survival of the ecosystem. Whether from a technological, commercial, or social standpoint, the solutions developed in the SECO are established through partnerships or interactions amongst the actors involved [4]. The literature identifies the most common actors that are encountered in the ecosystem as *Orchestrator, niche player, external actor*, and *end user*. Based on the literature findings, Figure 4 depicts the contributions of the ecosystem actors to the software platform in the roles identified. For this thesis, the actor contribution model is designed to identify the various participants in a platform-based ecosystem. Furthermore, a new support coordinator role has been included that has not been specified in the literature according to the researcher's understanding.

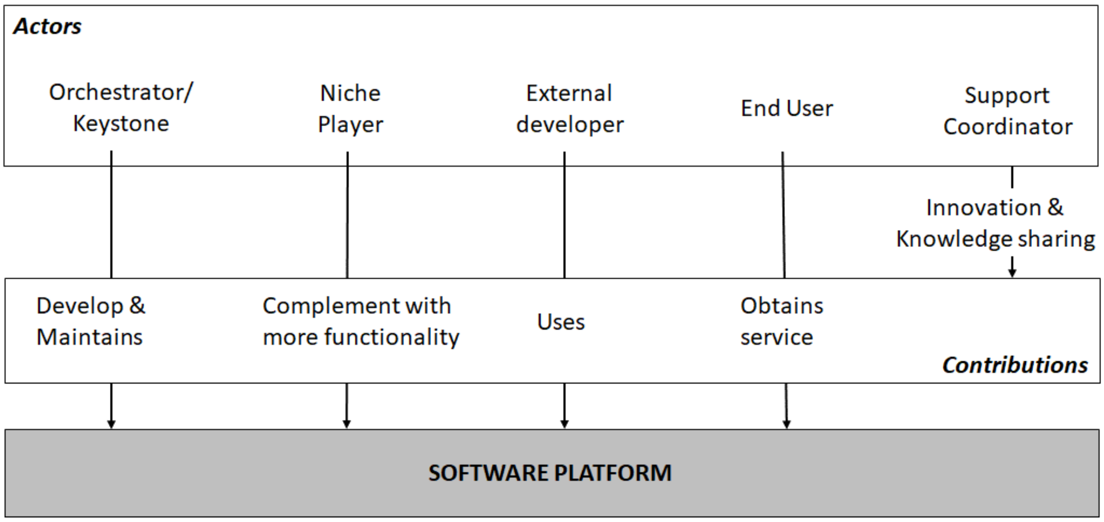

Figure 4. Actor Role contributions identified in an SECO.

The *orchestrator* or *keystone* player oversees the ecosystem. It might be an individual or a consortium that is in charge of administering the platform, developing and implementing rules, processes, and business procedures, as well as defining and monitoring quality standards [44]. The Apple ecosystem serves as the greatest and most frequently cited example, demonstrating how the individual corporation has total control over its ecosystem. The orchestrator plays a central role in an ecosystem, with multiple tasks ranging from platform development to monitoring ecosystem health.

The *niche* player is an actor who supplements an orchestrator's work by developing or adding components to the platform, resulting in functionality [44]. The niche players are the most developed and may have an impact on SECO management decision-making. They are often responsible for innovation, designing specific extensions to the platform and therefore adding value to the ecosystem.

External actors, often known by numerous names such as third-party developers, external parties, or external adopters, are the actors who take advantage of the opportunities provided by the ecosystem and provide indirect value to it. External actors may be developing on top of or parallel to the SECO platform, identifying flaws, promoting SECO, or even attempting to improve the platform. Vendor or resellers are mainly the organizations that make a profit from selling the products of the SECO to customers. These are the companies that enhance SECO's conversion into a new product by adding more capabilities to it. Then the role is the *end user* that obtains or purchases a complete or partial product of the SECO.[44]. An end user could be individual or even organizations that use the SECO platform for developing their products or application.

Furthermore, the last role listed is the *support coordinator*. Though this role does not seem to be covered in the literature, observation of the research collaborations at the university shows that such a role could have significant implications in a ROS ecosystem. The support coordinator's job could be viewed as the one that promotes collaborative innovation and knowledge sharing. The innovation is accomplished through bringing together multiple organizations and collaborating on research initiatives. Support coordinators could assist and enable other ecosystem players to fulfill their objectives, thereby indirectly contributing to the ecosystem. Universities and research groups are the best examples of these roles, producing innovations through cooperating on research projects.

4. ROS BASED ECOSYSTEM

Although the name implies, it is not an operating system, but rather an open-source framework that includes a collection of software libraries and development tools for building robotic applications. Basically it is a software development kit equipped with tools, packages, and frameworks that act as building blocks for developing robotics algorithms and hardware platform [13]. The usage of ROS is not restricted to robotics applications; it also includes tools for interacting with peripheral hardware such as sensors and actuators. Aside from its popularity in the robotics research community, it has also made its way into industrial environments [13]. It has gained traction in practically every intelligent machine area, enabling autonomous navigation simulation, visualization, control, and a variety of other services [11, 12].

4.1 Origin of ROS

ROS originated as a personal project for Keenan Wyrobek and Eric Berger, who were looking for a solution to the robotics problem of reinventing the wheel. While at Stanford, these two were dealing with the most common challenge in robotics: devoting too much time to re-building software infrastructure for complex algorithms and not enough time to work on intelligent robotics systems. To solve this issue, Eric and Keenan established the Stanford Personal Robotics Program in 2006, with the purpose of creating a framework that will help in the design of code-based tools as well as the communication process. The plan at the same time was to develop 10 identical robots and distribute them across universities to help innovators develop software based on their framework. At the same time, many other projects were going on in the robotics community.[65].

The project received its first grant, which was utilized to construct the PR1 robot. Working on various pitches to raise funds to support ROS and construct the “Linux of Robotics”, Eric and Keenan finally received assistance from Scott Hassan, resulting in the formation of Willow Garage and the beginning of the ROS journey [65]. Willow Garage the robotic research lab and the technology incubator eventually concentrated its entire focus on developing of the personal robot program. It had progressed through working on a variety of projects and gaining a reputation in the market. The initial ROS distribution was launched in 2009, followed by PR2, the second edition of the Personal Robot [69]. Slowly the ROS platform started picking up with universities and businesses incorporating it into their products. Willow Garage announced its departure from the ROS in 2013, then the

Open Source Robotics Foundation (OSRF) took over and continued promoting it to become the industry standard for robotics [69]. Over time, ROS has developed into a thriving ecosystem of software algorithms, packages, tools, and a community of experienced developers working together to make it easier to create complex and reliable robot behaviour across a number of robotics and industrial platforms [34].

4.2 ROS as a communication middleware

ROS is a free and open source software framework that includes packages and a collection of tools for communicating between components and offering functionality for building advanced robots. ROS provides basic operating system services such as hardware abstraction, low-level device control, common-use functionality implementation, and message passing between processes and packages. According to the Quigley [53], ROS serves as a communication gateway between hardware and software, allowing advanced programming environments to manage low-level control. The Figure 5. explains the process of how ROS functions as a communication middleware.

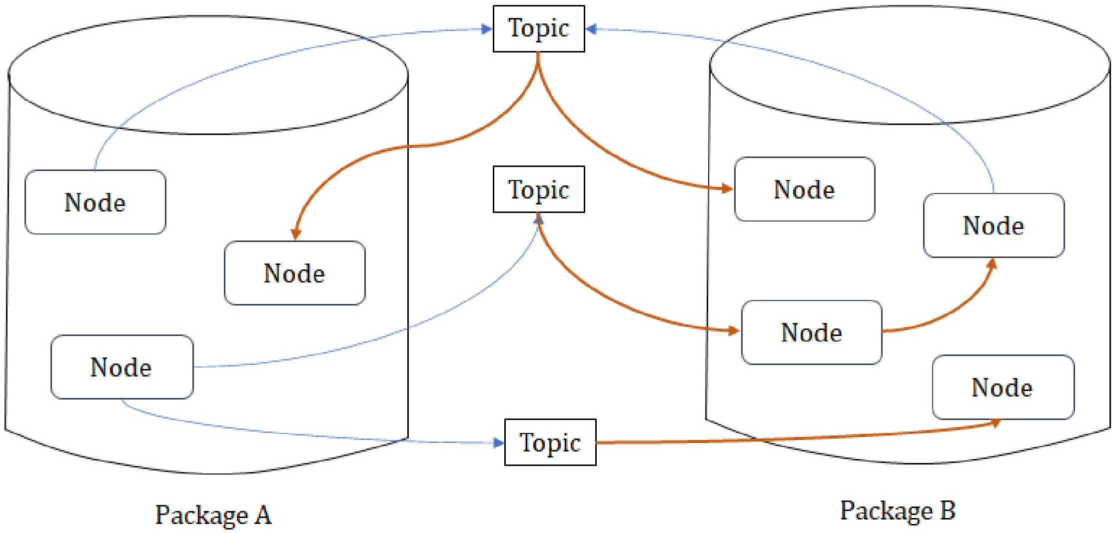

Figure 5. Working of ROS packages (modified from [78]).

The communication process starts with the ROS nodes. The node is an executable file that performs computation, and each node is assigned to a specific task [78]. These nodes communicate with one another by sending messages through logical channels known as topics. The topic is the information that is contained in the ROS. The topics can be published or subscribed by nodes to exchange or access the data between them. Single or multiple nodes could be bundled together to perform a specific task [78]. Such bundles are called ROS packages. From the figure 4 consider package A as a lidar sensor and package B as an Xbox controller package. Package A consists of several nodes with various tasks assigned to them. To provide the necessary functionality, multiple nodes could publish or subscribe to the topics and communicate between the packages. The packages can be arranged into ROS stacks, making the sharing of codes easier. Stacks basically are the primary means of distributing the software.

Evolution of ROS to ROS 2

In the recent years, significant advancements in industrial equipment manufacturers have been seen shifting towards automation and autonomous solutions. The growing need for advanced equipment makes it essential to have new software technology to control and communicate between the devices. Apart from this, the common characteristics found in autonomous machines, robots, drones, industrial equipment, etc, are the real time capabilities. For instance, it is necessary for an autonomous vehicle to tackle the challenges across a path that it is following such as obstacle avoidance, making a decision, and reacting. Real time constraints are quite common in robotics, requiring the capabilities like reliability, priority, and synchronization [56].

After many years of using and developing ROS, various problems arise in different use cases. ROS, in particular, did not support the real-time environment, which affected task prioritization and synchronization in multi-tasking scenarios [56]. To address the inadequacies of ROS, notably in the real-time domain, the ROS community upgraded to the ROS 2 version [40]. To facilitate ROS 2 delivery and widen the ROS community, the Technical Steering Committee (TSC) was established to focus on various topics. ROS 2 uses the Data Distribution Service (DDS) middleware expanding its capabilities such as high- level system integration and real time applications. Through DDS ROS 2 is able to provide serialization, transport, and discovery [40]. Since all implementations in ROS 2 are based on the DDS standard, it has enabled increased quality, performance, security, and communication across various networks [40].

4.3 ROS Ecosystem

As such, ROS meets the definition of software ecosystem- “the software and actor interaction in relation to a common technological infrastructure that results in a set of contributions and influences directly or indirectly the ecosystem” [43]. In comparison to the software ecosystem, the ROS ecosystem is a veritable cornucopia of open source robot software, that includes software packages, algorithms, communication interfaces, tools, community, and industrial support required to develop robotics applications in a variety of fields [76].

The ROS ecosystem is organized into four pillars that serve as the foundation for developing new applications and robots. Plumbing, Tools, Capabilities, and Community are the pillars that support the ROS ecosystem [76]. Plumbing is a messaging architecture provided by ROS to enable rapid and easy communication between various systems. The ecosystem is equipped with tools that can be used in several ways such as configuring, debugging, visualizing, monitoring, and testing robotic applications. The ecosystem reduces the barrier to producing robotic applications for both experienced and inexperienced workers by offering capabilities in building specific tasks such as control, planning, perception, and mapping. Finally, there is the community, which is supported by a big number of diverse and worldwide players and places a high focus on integration and documentation. The community helps in a number of ways, including offering basic and advanced ROS tutorials, answering questions in the discussion forum, and developing and maintaining distribution releases.[76].

Actors in ROS Ecosystem

Based on the literature, the players in an ecosystem have different roles and responsibilities based on their capabilities and needs. While some players spend more time and resources to the ecosystem, others do not. Every ecosystem, open or closed, has an orchestrator or a keystone that governs the ecosystem functions. Based on the literature review on ecosystem, several actors can be identified that are collaborating to the ROS platform forming a ROS ecosystem. Figure 6 represents the actor classification of the ROS ecosystem based on literature.

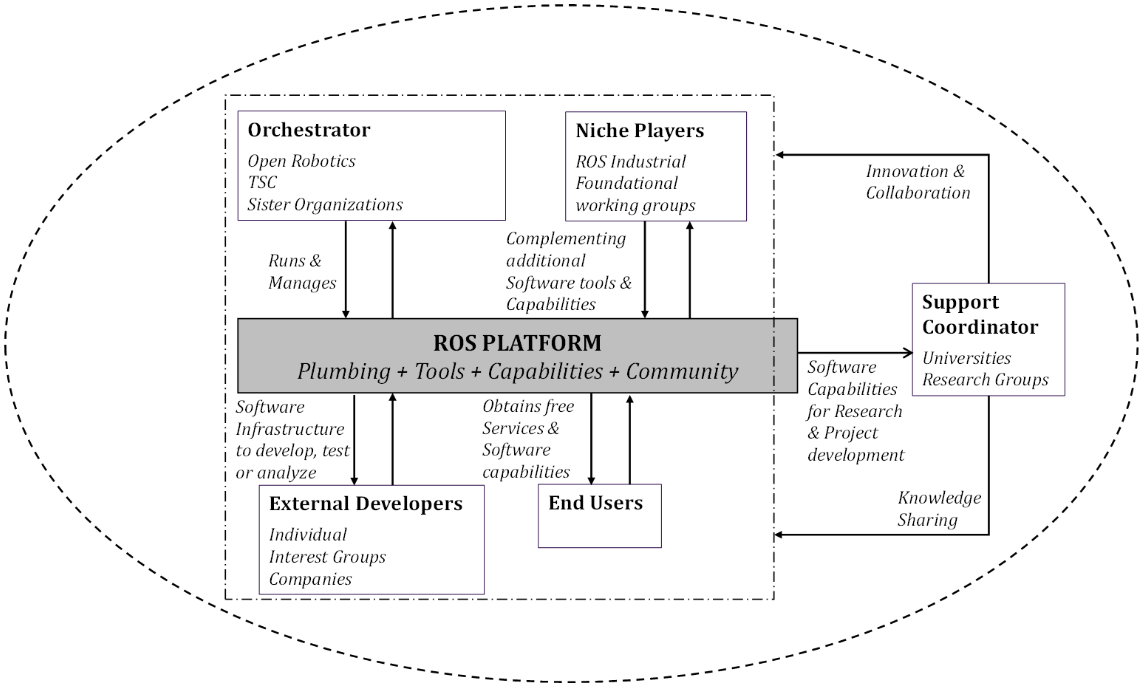

Figure 6. Actors identified in ROS Ecosystem based on the roles presented in 3.3.

The *orchestrator* is a non-profit organization consortium in charge of developing and maintaining the ROS platform, which includes the website, as well as developing and managing the distribution of major releases. The orchestrator comprises entities namely Open Robotics, Technical Steering Committee (TSC), and the Sister organizations. Open Robotics is a non-profit open source robotics foundation that *"support the development, distribution, and adoption of open source software for use in robotics research, education, and product development"* [82]. TSC is specifically established to work on building the roadmap, develop core tools, and libraries´, and establish working groups on ROS 2. The group is made up of officials from well-known companies like Microsoft and Amazon who are contributing materially towards ROS 2 and are in charge of determining the project's technical direction [75]. Sister organizations are the ROS groups that have grown to the size where they have formed into organizations. These companies develop and maintain the source code and packages for certain robotics applications, as well as work on diverse projects with businesses in some contexts [77].

The ROS industrial and Foundational working groups could be identified as the *niche* players complementing the platform. ROS industrial project adds advanced capabilities to the platform that can be further utilized in industrial relevant hardware and applications. The Foundational working groups work together with similar areas of expertise to develop standardized base applications in robotics such as navigation and perception [34]. There are many players that can be identified as *external* developers and *end users* of the platform. For instance, there are groups of interested people that are working on particular projects of shared interest. Even individuals could be seen that are trying to contribute to the platform for instance by developing their own packages or source codes. Various small-scale and large-scale industries could be end users endeavoring to obtain the services from the platform. Academic, research, or competition teams can act as an external developers or as end users in the ecosystem.

Finally, the role of the *support coordinator* might well be defined as sharing knowledge and facilitating collaborative among ecosystem participants. Universities or research groups are most likely that could be identified in this role. For example, the PEAMS group at the university may serve as a support coordinator, utilizing the platform's tools to facilitate collaboration among the many project participants. It can be observed from the study that a single player could act in multiple roles. Different roles demonstrate how the users of the platform utilize it to fulfill their objectives in multiple configurations.

5. RESEARCH METHODOLOGY

This chapter describes the research approach and technique used to investigate the significance and literature of the software-platform based ecosystem in the heavy-duty mobile machinery industry. It is then followed by a brief summary of the cases selected for this study. The subsequent sections explain how the data was gathered and analyzed.

5.1 Multiple Case Study approach

The research in this study aims to understand the significance of the software platforms and ecosystems in the organizations that are associated with the heavy-duty mobile machine industry. To develop a holistic understanding and to ensure the validity and generalizability of the findings, a multiple case study approach was used. The multiple case study method also enables the exploration and comprehension of complicated issues by examining facts in a particular context [68]. Figure 7, illustrates multiple case study approach implemented in this research context.

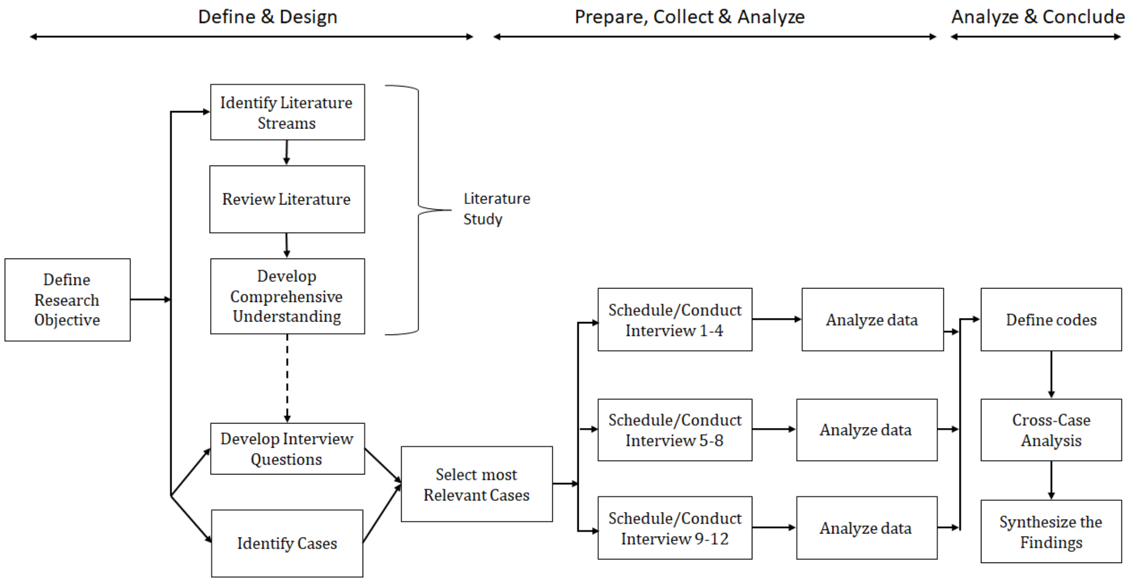

Figure 7. Multiple case study process (adapted from case study research [66]).

The utilization of multiple case studies in this research provides for a more comprehensive knowledge of the perspectives of various players in the heavy-duty mobile machine industry, making the findings strong and credible. Additionally, as shown in the Figure above, this case study approach allows the researcher to assess data within and across the cases, improving the research's validity [66]. Furthermore, it is possible for the researcher to analyze how different organizations use software platforms and ecosystems to accomplish their objectives.

According to Gillham [22], a multiple case study aids in investigating the answers to the required research questions using a variety of different types of evidence. This study intends to collect qualitative data in order to investigate and comprehend the significance of software and its role in research and development, as well as the role of software ecosystems in heavy-duty mobile machine industries. To address the research study's objective, multiple players are identified, to understand the various perspectives of the companies that perform distinct roles in the heavy-duty mobile machine industry. The study involves several systematic stages that are described in the following sections.

5.2 Case Selection

A purposeful sampling is used in this study to pick the most insightful cases to address the research questions through practical and pragmatic considerations of research aims [15]. Purposive sampling is a frequently used strategy in qualitative research that utilizes a specific set of criteria to ensure the most use of limited resources by determining the most 'information-rich' cases [15, 49]. The research study's focus is to understand the perspectives of the software environment in the heavy-duty mobile machine industry. As a result, a sample of cases was chosen that had diverse ways of engaging with the software component in their businesses. Through such a reasonable approach to case selection, the researcher hopes to comprehend the significance of the software platforms and ecosystems, as well as the potential that may be identified for organizations and consumers.

For selecting the case companies that would be suitable for the research study, data of PEAMS and FIMA members were utilized. The potential cases were then chosen using a set of criteria. First, the cases must be associated with the heavy-duty mobile machine industry. Second, the case companies need to have assumed the role in the industry either as a software provider, machine manufacturer, or embedded systems provider. Finally, the case companies should be able to appoint the company representatives, for an interview, who have working experience with the software. Figure 8, represents the identified categories among the potential cases.

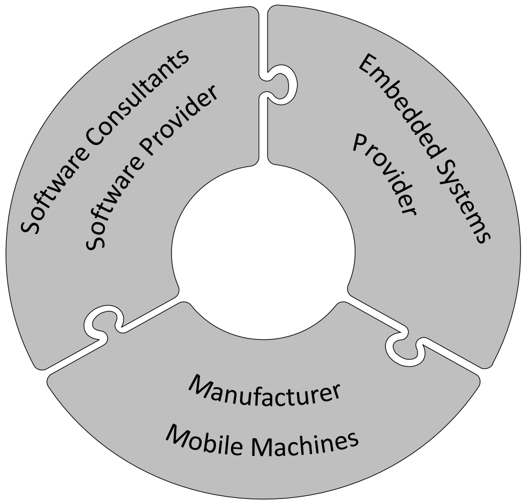

Figure 8. Segmentation of cases identified.

Firstly, it is the software consultants and the software provider that provide the necessary software expertise, resources, and strategic advice. It also includes delivering the complete software packages as per the requirement. The second category included companies that develop embedded software for the applications that were needed in the heavy-duty mobile machines. Many times, the companies included embedded software with the hardware. Finally, the third group included enterprises that were manufacturing their own products. Manufacturing specifically covered material handling equipment in several domains such as forestry, warehouse, construction, and shipyards. Most of the selected cases were Finland-based companies, that operate globally.

5.3 Semi-Structured Case Interviews

Qualitative approaches are descriptive and inferential in nature, with a primary focus on the type of evidence that will allow you to comprehend the practicality of what is going on [22]. Therefore, a qualitative method is used for collecting the data as the research study is both exploratory and explanatory. Qualitative data can be obtained in a variety of ways, including interviews, questionnaires, and observations [22]. Depending on the intricacy of the research, many approaches might be employed to collect data. However, for this study, qualitative interviews were considered the most suitable way to collect primary data.

Based on the literature there are three basic formats of interviews: structured interviews, unstructured interviews, and semi-structured interviews [21]. For the case study, a semi-structured interview methodology was adopted because it gives a series of essential questions that will serve to define the themes to be investigated and also allows the interviewer or interviewee to deviate in order to follow a concept or response in further detail. The interview questions were broken into distinct themes, with each subject having its own set of questions. To guarantee a seamless flow of interviews covering all of the subjects, a predefined sequence was followed throughout the interviews. The semi-structured interview provided the researcher with the option to offer more enhanced questions in order to gain deeper insights into certain themes highlighted throughout the dialogues [1]. Figure 9, illustrates the questions derived under the themes identified.

Semi-structured interview questions

Company & Interviewee Background

1. How would you describe your company's core business?
2. What are your company's main products?
3. What is your position in the company?
4. What are your key responsibilities?
5. How long have you been working in this field?

Software Development Process & Practices

6. What kind of software does your organization develop?
7. How is the software development organized in your company?
8. If your company lacks certain competencies or resources for software development, how do you solve them?
9. What are the most critical factors or criteria that influence your decisions to acquire such competencies or resources?
10. Are there other ways how your company obtains software needed in products?
11. What factors influence your decision when obtaining software in these ways?
12. How do you think the current approach for software development could change in future?

Software Platforms

13. What does Software Platform mean from your company's perspective?
14. Are there any Software Platforms that you are using?
15. Does the company see any benefits or risks of using a common software platform that is also used by other companies?

Software Ecosystems

16. What does software Ecosystem mean from your company's perspective?
17. Is your company currently a part of any software ecosystem?
18. Does your company collaborate with other companies while working on Software related matters?
19. What are the benefits of working collaboratively?
20. What are the risks or concerns of working together?

Robot Operating System

21. What idea do you have about the ROS in general?
22. Would the company consider using ROS for developing automation or robotics? If so, why or why not?
23. Do you think that ROS could be a source of creating new business opportunities or a competitive advantage for the company?

Figure 9. Questionnaire designed for the semi-structured interview.

The interviews were conducted professionally, in the English language. The interviewees were industry professionals with years of expertise who held high-level positions in case organizations. To give the interviewers a sense of the agenda, the confidential document along with the idea of interview themes were drafted and conveyed prior to the interviews through email. To get the perspectives of the interviewees that may be rather context-specific and coming from experience, the interview questions were not disclosed in advance to the interviewees. The interviews were conducted online using the zoom teleconferencing service [2], and with the interviewees' consent, they were recorded. The interviews took place over a four-month period, from March to June 2022. To minimize misunderstandings caused by language or technology barriers, the questions were expressed both verbally and in the form of a presentation during the interviews. Figure 10, shows an example of the zoom view with the question presented during the interviews.

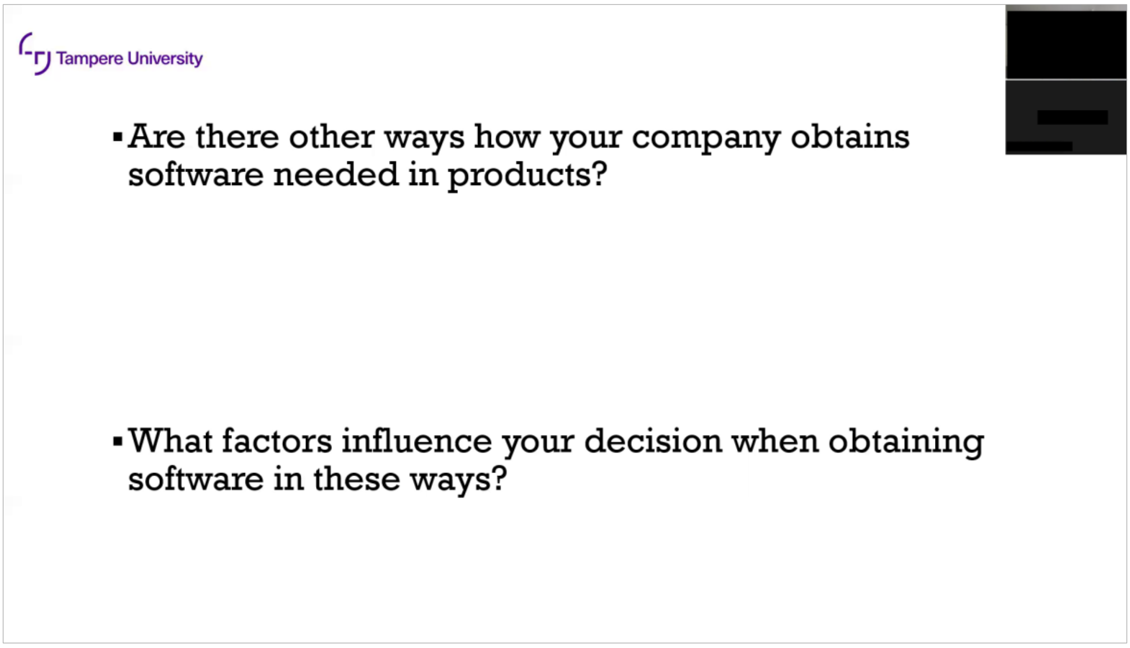

Figure 10. Example illustrating the use of presentation during interviews.

The recorded interviews were then further manually transcribed by the researcher for further analysis. The total time required for the transcription would be around 130 hours. The transcribed versions were analyzed using the atlas.ti application [71], which made it simple to code and extracts the highlights from the interviews. The real names of the case companies are kept confidential, and they are given distinct codes instead. The code S indicates that the organization is categorized as a software provider/consultant. The code E represents a company designated as a provider of embedded systems, while the code M represents the role of a machine manufacturer. Table 2 below presents the technical information of the interviews, including the positions of the interviewees in the case organizations.

Table 2. Interviewee Details.

Interviewee Code	Case Organizations	Current role	Experience (years)	Interview duration (min)
S1	Software Consultant	Business director	20	40
S2	Software Provider	Vice President	19	45
S3	Software Consultant	Business Director	15	45
S4	Software provider	Project Manager	4,5	60
S5	Software Provider	Product Manager	3,5	35
E1	Embedded systems provider	CTO	15	72
E2	Embedded systems provider	R&D Manager	15	55
E3	Embedded systems provider	Technology Director	20	45
M1	Mobile Machine Manufacturer	Project Manager	13	60
M1	Mobile Machine Manufacturer	Team Lead	10	60
M2	Mobile Machine Manufacturer	Manager	17	40
M3	Mobile Machine Manufacturer	Team Lead	6	85
M4	Mobile Machine Manufacturer	Research Director	19	90

As seen from the table, altogether, 12 interviews were conducted. The length of the interviews ranged between 30 to 90 minutes in total. The interviewees were from a range of organizations, for instance, 5 of them come from the software provider and software consultant sector.

5.4 Data Analysis

Data analysis seeks to explain, justify or address the research questions based on the assessment of available data [51]. In this study, both inductive and thematic techniques are used to qualitatively analyze the data. Qualitative data analysis is the process of transforming enormous amounts of data into organized, and comprehensible findings [36]. The inductive analysis method uses context-related categories or themes to build the research study on pre-existing literature or frameworks [24]. Inductive analysis uses the existing literature to formulate the categorizes and research questions to be addressed [58]. The thematic analysis is further applied to analyze the qualitative data. Thematic analysis is the technique for “identifying, analyzing, organizing, characterizing, and presenting findings discovered in a data collection” [8, 9]. The analysis intends to generate a comprehensive understanding of what was stated while also ensuring that all relevant aspects of the data are considered [19]. The data analysis in this study is a nonlinear process that requires numerous iterations of observing and analyzing the data at various levels to optimize the data.

To thoroughly understand the concept of software platforms and ecosystems, this research started with the review of relevant literature. Based on the literature review, specific categories were identified. Multiple categories were analyzed, and based on those, the interview questions were prepared. Data is obtained through conducting interviews and recording the material in the form of transcripts. The previous chapter 5.3 has a detailed discussion of gathering the interview data. Figure 11 below illustrates the process for analyzing the interview data.

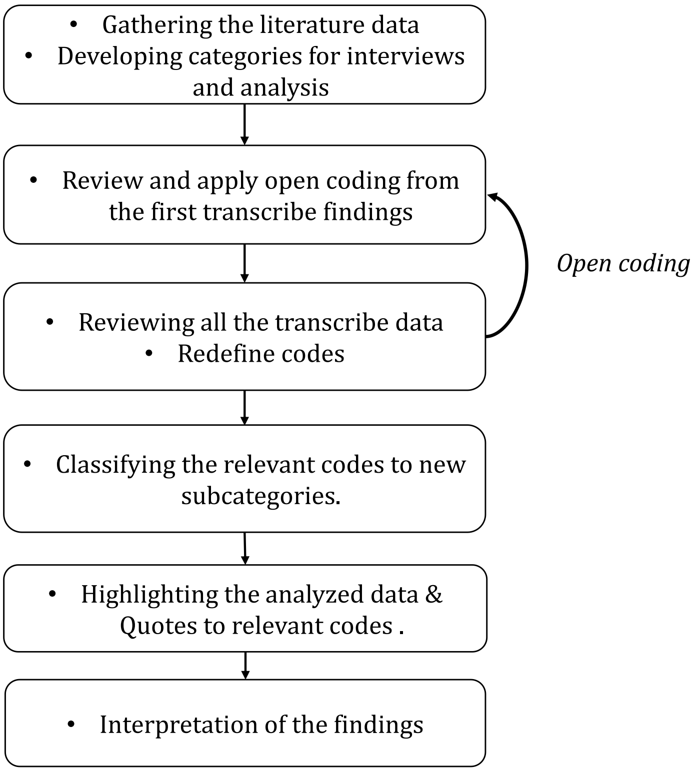

Figure 11. The data analysis process for the interview findings.

For this study, the open coding approach has been the key for evaluating transcripts and retrieving the data to address the research questions. Open coding is a qualitative research analysis technique that identifies and categorizes the recurring responses found across the research material [64]. The use of an open coding technique ensures that the researcher could achieve the validity and reliability of requirements associated with qualitative research data [51, 64].

An open coding strategy was preferred because the study is more exploratory rather than explanatory. The open coding technique allows the flexibility to the researcher for updating the codes depending on new relevant data obtained from later transcripts [18]. This method compares the codes constantly and focuses on examining commonly appearing contents across distinct transcripts [64]. A set of 'codes' in the form of terms or phrases were identified based on the interview themes and the first interview transcript. The common patterns of texts were then consolidated and analyzed across all the transcripts and assigned against the respective codes. For example, Figure 12 displays the application of the atlast.ti tool to analyze transcribes, that demonstrate the frequency of occurrence of identical remarks to a particular code.

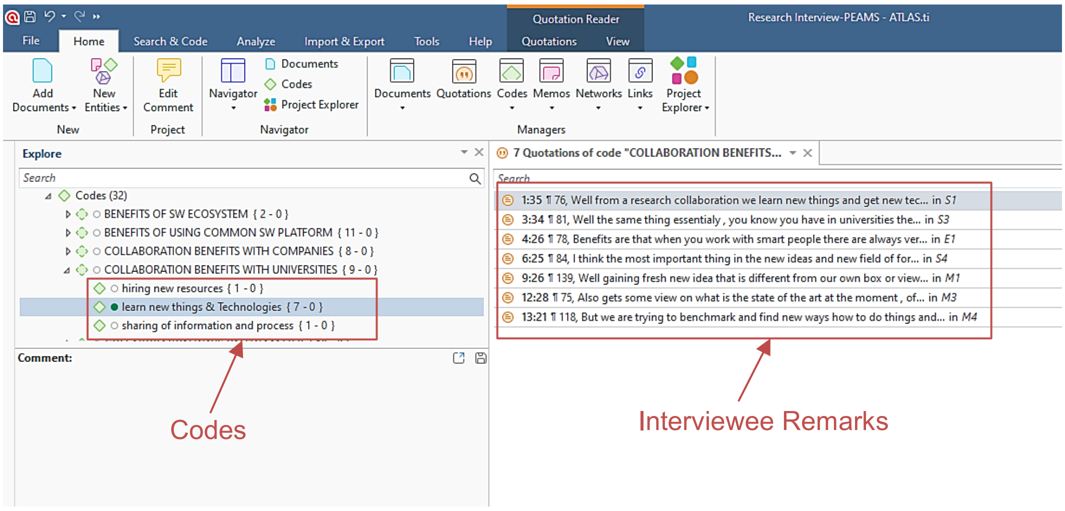

Figure 12. Atlast.ti tool used for the open coding analysis.

Next, the codes for each topic were then subcategorized based on transcript findings that were similar and relevant to the study. For example, all instances related to the SW role were assigned codes and classified under different subcategories. Figure 13 below demonstrates an example showing the analysis of one of the categories after going through all the transcripts.

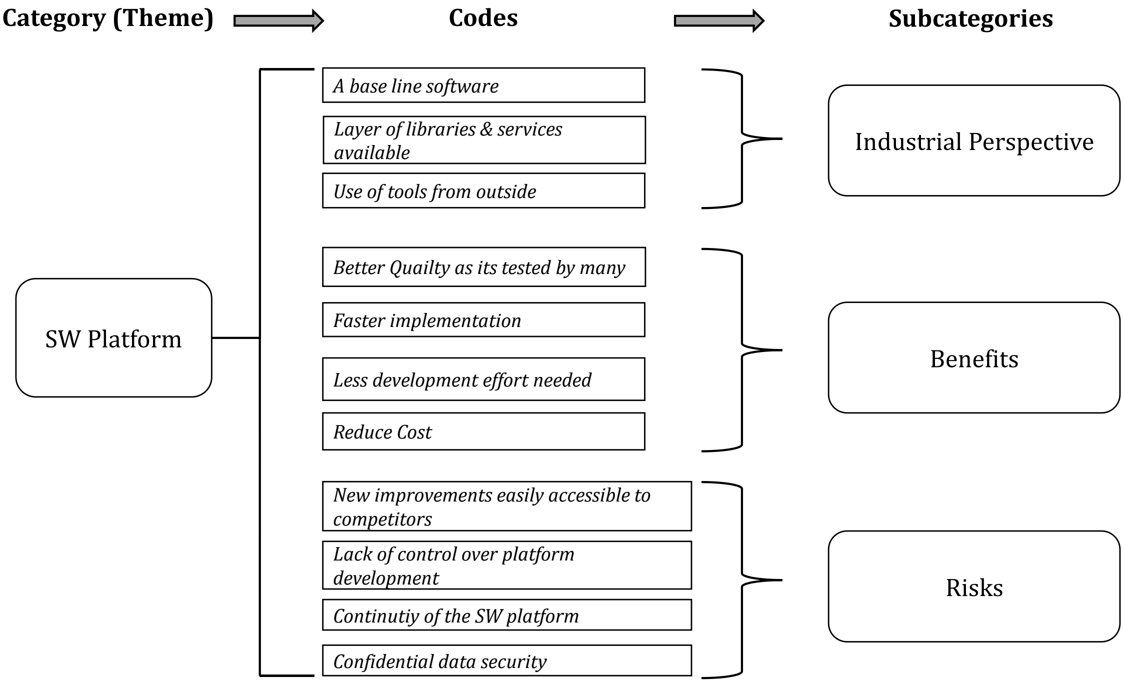

Figure 13. Example showing the category analysis of SW platform theme.

In all, 65 codes were derived from four categories and further categorized into 11 subcategories. The number of codes assigned to each subcategory were unequally distributed in order to cover all relevant information from the transcripts. A summary of the categories and subcategories is presented in table 4, chapter 6. The final part of data analysis involved cross-case analysis. The findings from different cases were compared and synthesized to answer the research questions. Inferences, to answer the research questions, were made based on the occurrences of the codes under the different categories and subcategories. The next chapter presents the findings from the empirical study.

6. INTERVIEW RESULTS AND ANALYSIS

This section covers the outcomes of empirical research compiled and analyzed through qualitative interviews with the selected case organizations. The findings are categorized under four different categories or themes addressing the discussion and summary of the relevant information. The subchapters in this section, from subchapter 6.1 to subchapter 6.4, discuss the four themes: software roles, software platform, software ecosystem, and ROS ecosystem, respectively. Table 4 outlines the categories and subcategories, as well as the associated interview questions and sample interviewee remarks.

Table 3. Outlined of the data analysis.

Categories	Sub-Categories	Definition	Interview Questions	Sample Interview Quotes under various Codes
Software Role	SW developing Ways	approaches for developing a software	6,7,8,9	*"...use OSS components…licensing or buying the product…"*
	SW development Future	how would the approach for SW development change in future	12	*"...using may be even much more networks and other companies to work with .."*
Software Platform	Industrial Perspective	Organization's view on SW Platform	13,14	*"...about a layer of libraries & services that application developers can utilize…"*
	Benefits	advantages of using SW platform	15	*"the development effort is done by somebody else... "*
	Risks	concerns of using SW platform	15	*"...the quality or how well can you know if something goes wrong"*
Software Ecosystem	Industrial Perspective	Organization's view on SW Ecosystem	16,17	*"there is somebody who take the burden of maintenance & continuity..."*
	Collaboration ways	cooperation with others when working on SW-related topics	18	*"have collaboration with universities but those comes from our research…"*
	Benefits	advantages of being a part of an ecosystem	19	*" the sharing of ideas, sharing experiences…"*
	Risks	concerns of being a part of an ecosystem	20	*"some of the parties can't invest time, energy or money in the collaboration…"*
ROS Ecosystem	Knowledge	company's relation or idea on ROS/ROS2	21	*" when developing new software it has quite nice tools and functionalities built in"*
	Opportunities	Opportunities that ROS can provide in the future	22,23	*"...good chances to get a head start for developing some new & applying it for our applications …"*

6.1 Software role in the industry

Based on the findings of the interviews, the role of software seems to be broadening in the industry, enabling it to streamline its operational activities and provide new business opportunities. The software has changed heavy-duty machines into intelligent machines by sensing and evaluating the operational circumstances under which they operate. The initial questions were addressed to the interviewees on the kind of software they developed or used in their business.

The majority of the software developed by the case industries were basically control systems software and different user interface to monitor and manage different environments for mobile work machines. The major role of the software developed by the case industries was in achieving the automation and autonomous functions in the heavy-duty mobile machines. Obstacle detection avoidance, road mapping, and navigation were some of the software applications that were developed by the case industries. These are the software that acts as a ready package that the customer can easily install on top of its hardware. The software consultant and providers were the major players in providing such type of ready software packages and also the software services such as resources and consulting. One of the respondents mentioned offboard and onboard software, particularly in the context of heavy-duty mobile machinery.

> *"the onboard software does the sensing and controls the machine movement direction locally on the local computer on the machine, whereas the offboard software typically decides the vehicle routes and assigns the machines to perform the task in coordination with others"-S1*

Then, as the term indicates, there were organizations known as embedded system providers, who provided both hardware and software. Such systems included the Programmable Logic Controller (PLC) programming and machine control system that are used on the machine to execute specific functions such as the boom control, hydraulic actuators, and sensor control. 3 out of 12 cases offered the embedded software as per the requirement of the customer. The software development also included software used in the braking systems, air conditioning, and all other subsystems that are used in a vehicle. In answer to questions about the role of software, mobile machine manufacturers addressed the usage of embedded software as well as the software necessary to evaluate data from sources such as service, maintenance, and customer side.

6.1.1 How the Software is developed

The objective here is to understand how software development is organized in the heavy-duty mobile machine industry. The interviewees suggested that there are different ways how companies develop the software they need. For instance, either the company can develop the complete software by themselves, or a part of the software or resources needed for software development can be sourced externally. Table 5 below depicts how the case interviewees source the parts or resources for the development of the software they require.

Table 4. Sourcing options for software development.

Software Development	Software consultants/ providers	Embedded systems provider	Mobile Machine Manufacturers
Open-Source Software (OSS)	S1, S2, S3, S4	E1	M1, M2, M3, M4
Components off the Shelf (COTS)	S1, S2, S4	E2, E3	M1, M2, M3, M4
Subcontracting resources	S2, S3, S5	E1, E2	M1, M2, M3, M4
Outsourcing to suppliers	S3, S5	E1, E3	M1, M2, M3

When it comes to software development, even the most software-intensive businesses use multiple approaches. Instead of developing the software from scratch, options like open source software or components off the shelf (COTS) are considered. For example one of the interviewees stated:

> *"Not necessarily the software is built in house, ofcourse we try to first look if there is open source alternative's which could be utilize for commercial purpose if not then we try to find certain partners that could have a ready-made product… " (S4).*

COTS are not only products that are pre-built and proprietary software, but it also involves software libraries that can be licensed or purchased. Subcontracting resources entails hiring the software specialists required for your project on a part-time or full-time basis. Often this is the most likely scenario that has been observed through the case interviewees. Finally, it's outsourcing the complete or either a part of a software to consultants, subcontractors, suppliers, or in some cases they are even partners. Based on the case interviews, the option to outsource to a supplier is more prevalent on the manufacturing side. For instance, one of the case interviewees from the mobile manufacturers quoted:

> *"Well all the production software that we are producing for the machines we are doing it externally with suppliers but we have a quite strict steering from our inhouse guys."-M2*

According to the interviewees, it is difficult to predict a percentage of how much software is completely developed in-house and how many other options are used. But the core part of the software is developed inhouse. One of the interviewees mentioned that it depends upon the project or the customer requirement where they might lookout for other options, otherwise it is developed in-house (S2). The interviewees mentioned several factors that led to exploring other options for software development. In addition, the factors influencing the decision to choose between multiple options were addressed. Table 6 below illustrates the most highlighted factors addressed through the discussion with the interviewees.

Table 5. Factors influencing external SW development.

Factors driving external SW development options	**Selection criteria across multiple options**
Lack of expertise	Pricing
Limited Timeline	Availability
Accelerate the development	Licensing model
Escape Vendor locking	Quality
	Technical Performance

The most often criteria (10 out of 12) that were addressed was the lack of expertise whether it's for the hardware or software part. Sometimes businesses lack certain competencies, particularly when dealing with emerging technologies or other novel concepts. In these situations, subcontracting resources are preferred. In some cases, there is a limited time frame in which the software component needs to be completed, so such situations often drive the requirement to seek external assistance. In particular, cases using help from outside accelerates the development rather than doing it from scratch. Additionally, outsourcing part or completed software also shortens the time needed to develop and release the software into the market. In the context of software development, long-term relationships make industries dependent on the same supplier or vendor. One of the factors driving the case interviewee to explore other options was vendor lock-in. These were the key elements that stood out during the case interviewees and prompted the industries to examine various options for software development.

6.1.2 Future of Software development

Through case interviews, it was seen how significantly the methods of software development have changed over time. For instance, using agile methodologies for development, open source software components, and model-based approaches while creating a code makes it much easier to build and test in the actual environment. Different tools and advanced systems have steadily evolved over time, providing flexibility for businesses and software developers. The key findings on the expectations of the future of software development addressed during the discussion are being identified as:

- Standardization of Software interfaces
- Outsourcing Software work
- Increased use of Open Source software components
- More open accessible software interfaces

Participants in the interviews expressed a need for software to become more standardized with interfaces and architectural designs that allow for reuse across various modules. Standardization facilitates communication between different interfaces or modules and makes it simple to replace a module of your software solutions with something else. One of the interviewees expressed the benefit of standardization quotes as

> *"…basically, you know what is the interface to that software and you can develop against that your own things and that way you could combine multiple different software's for that specific purpose to make you product so that you wouldn't need to have this huge integration project always to integrate software's from multiple vendors"-S4*

Software development has become more complex requiring complex algorithms with advanced skills and expertise. The interviewees expect a rise in using more networks and partners rather than doing the complete software by themselves. The use of open source components seems to be beneficial as it saves time, cost, and effort in developing new software. In future the open source software will be a viable option in many fields, for instance, one of the interviewees mentioned "*I think for PLC programming open source will become more and more viable option*" (M3). The industries would like to see more open and easily accessible software interfaces. More open interfaces would make it easier to understand the system's input and output, speeding up the development and integration process.

6.2 Software Platform

6.2.1 Industrial view on software platforms

The empirical analysis suggests three distinct viewpoints on software platforms, namely baseline software, software libraries & services, and external tools. The following Figure 14 illustrates the perspectives of the interviewees identified.

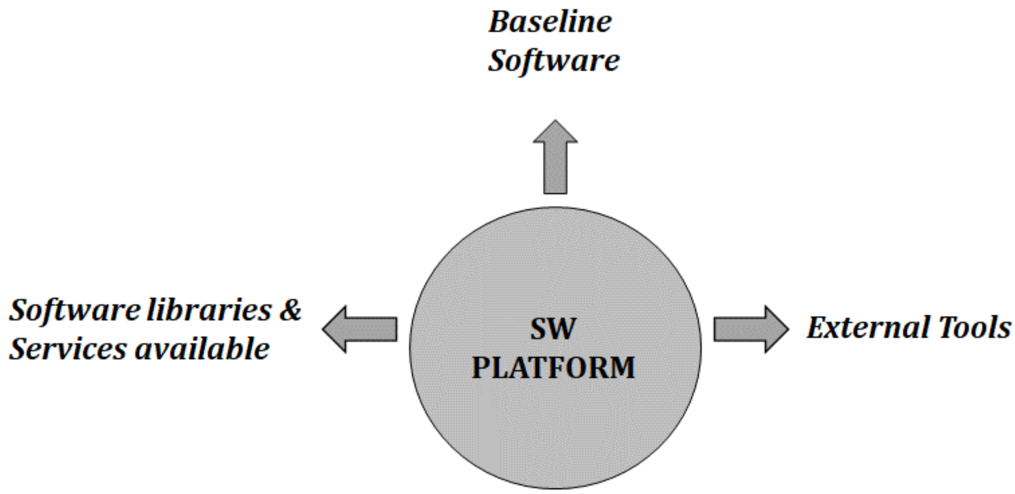

Figure 14. Differing perspectives on software platforms.

First, 5 out of 12 interviewees, mentioned software platform as baseline software. The software platform act as a kind of middleware or set of development tools upon which multiple applications can be built. As some mentioned, it is the core software that can be found in the majority of their applications. Such software platforms could be developed internally, or they can be sourced from outside, on top of which the industries build their own software. For instance, one of the interviewees mentioned their view about the software platform quoted:

> *"I guess in our case it's just mean that we reduced a lot of the modules withing different applications what we have. In some places we do utilize*

> *actually the same components or whatever you would call them and change just the configurations to adapt to different solutions or applications"-M4.*

Second, 4 out of 12 of the interviewees mentioned that a software platform is a layer of software libraries and services that applications developers can utilize. One of the interviewees mentioned ROS middleware as s kind of platform where you have access to specific software libraries. The use of azure services for cloud computing was also mentioned as a part of the platform. Accessibility to software codes, software development kit (SDK), and API that developers can utilize to develop various algorithms and integrate it with the system. As an illustrative example, the interviewee stated his opinion:

> *"So thats really what I see as an software platform essentially providing means and abstractions for applications developers that make their life a lot easier"-S3*

Finally, a few interviewees (3 out of 12) mentioned using external tools to support software development that they regarded as software platforms. The external tools included different programming languages or databases that are used for developing the software. On the embedded side, the tools used for developing PLC programming are similar to what the interviewee referred to as software platforms. The tools are the means by which you build software code, conduct analyses, and carry out tests. The tools cannot be used as a foundation for developing software. For instance, an interviewee states his perspective quoted as:

> *"But like I said we started from very scratch on our software, so kind of we only take the programming languages and tools or databases from outside, but we do not take like the core function as they started building on top of this, we build the core function in house and maintain that also"-S2*

Software platforms promote growth and add value through integration with various software development components, regardless of the perspective from which they are viewed. Software platforms have the ability to address software development challenges, but they will come with some risks and rewards.

6.2.2 Benefits & Risks

According to the interviewees, software platform has advantages whether an industry has its own or utilizes an external platform. 5 out of 12 respondents cited quality as a major advantage of using external platforms that are also used by other companies. The quality is significantly greater because the platforms have been tried, evaluated, and documented, making it easy for developers to implement and use. For instance, one of the interviewees from the manufacturing side stated as:

> *"Well the benefit is probably that if they are mulitple users of platform then its probaly being developed and there are resources behind it to be developed and if there is different organizations and companies making request for it then it can also make it more general and make it easier for us to use it in new ways"-M1*

Another benefit mentioned (4 out of 12) was the lesser effort needed to develop the software. The platforms are so well developed and can be utilized directly, which makes it less risky for companies to use them. It becomes easier for the software developer because the need to start from scratch is avoided causing lesser effort and manpower. For example, a case the interviewee from software industry stated:

> *"so the development effort is done by somebody else or at least shared with somebody else and if it's also used by somebody else its more used and therefore better tested"-S1.*

In some cases, using a software platform speeds up implementation and reduces time to market, providing more value to the customer. Other potential benefits identified included cost savings and easier integration when a similar platform is used by the end user. For example, few of the respondents mentioned the advantages of adopting a platform as follows:

> *"so those are the three main factors that we see benefit in. time to market is shorter, expenses are smaller, quailty is better and then may be a plus for developer experience is also better when you are using a software platform and you dont have do anything from scratch"-S3*
> *"…very much like that since we are very flexible a customer centric company I see as a benefit for our customers we can move very quickly if there is something that a particular customer or a group of customers would need from the platform"-E1*

Risks on using software platforms

The use of software platforms also came with certain risks, especially while using external platforms. One of the major risks identified was the continuity of the software platform. The risk is huge when someone is heavily using an external software platform and then suddenly due to certain reasons the platform is not available without any updates or upgrades available. The continuity also includes the maintenance of the software platform. For an example, an interviewee expresses concern of the software platform as:

> *"so a software platform needs active developer ecosystem, so you wanne keep this software platform alive. If you don't have maintainers, you might end up with a bunch of crap that not updated and have bugs in all that"-S3*

A few of the other risks noted were the lack of control over the platform's developments. It cannot be tailored to the user's individual solutions or applications. Then, in one of the cases discussed, if the company has also invested in developing a software platform, the new advancements may be easily accessible to competitors. One of the interviewees mentioned securities as also one of the risks that could expose confidential data on the platform. Further, the challenge addressed was the need for customization of the platform to integrate with the software. For instance, as mentioned by the interviewee the risk of customization quoted as:

> *"In order to get a platform in use we would need to change a whole lot of other things in our system and then we kind of lose the benefits out of it"-M4*

6.3 Software Ecosystem

6.3.1 Industrial view on software ecosystem

Similar to software platform, multiple perspectives on the software ecosystem were identified during the case study. 3 out of 12 respondents defined a software ecosystem as a collection of software that can be integrated and compatible with one another. Aside from the numerous software, the greater emphasis is placed on the collaboration between software vendors, subcontractors, large and small companies that are involved in those softwares. For instance, one of the interviewees expressed his views as quoted:

> *"I understand it as a cooperation of big and small companies involved in software enabled products…"-M3*

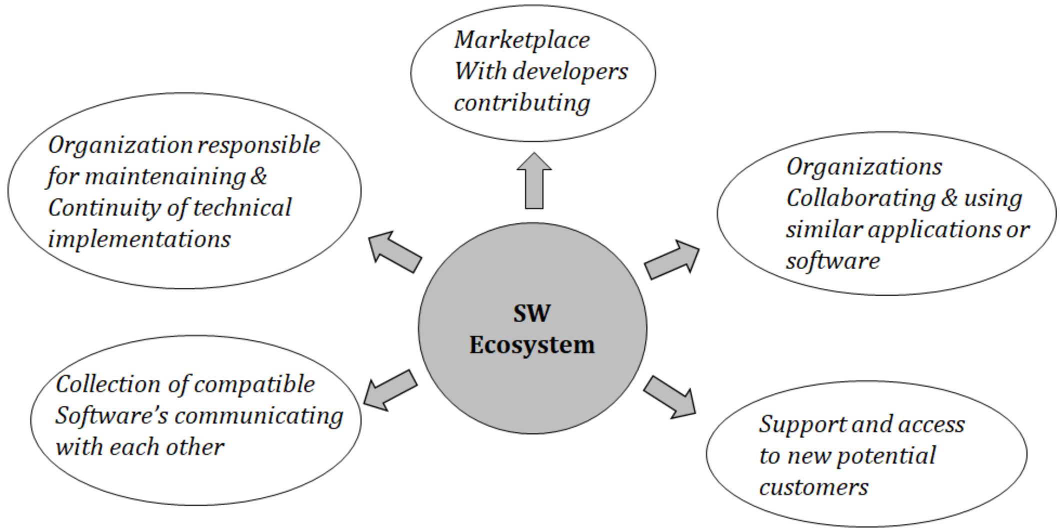

Figure 15.Multiple perspectives on software ecosystem.

Then 2 out of 12 interviewees addressed software ecosystem as a marketplace or kind of a platform. The marketplace allows developers to contribute their software to such an ecosystem. ROS was stated as an example of such an ecosystem where you have multiple developers contributing and also the software's are compatible with each other. For example, the respondent statement is quoted as:

> *"I think ROS is a good example in a sense you have a software from multiple vendors or companies and also open source community and all that software works nicely together"*
> *-S4*

Further viewpoint described the ecosystem as an organization that is responsible for ensuring the continuity of its technical implementation. The organization ensures the maintenance for example the modifications, bug removal, and upgrades required. One of the case interviewees described themselves as an ecosystem because they provide the entire software stack required to control an automated guided vehicle. The ecosystem comprised of the supporting tools along with the customers they served. Finally, it was about the ecosystem that offers support and access to customers. This sort of ecosystem provides applications and software platform that allows for increased scalability and access to new markets. The viewpoint addressed both closed and open software ecosystems. The open software ecosystem was referred to as being free accessibility to what it has to offer. Figure 15 summarizes the multiple perspectives identified on software ecosystems. A few of the viewpoints addressed by the interviewee are quoted below:

“Hopefully ecosystems means that there is somebody who take the burden of maintenance and continuity of the technical implementations utilized in such an ecosystem”-E1

“it’s a bigger version of software stack, so somehow connected set of software or programs and companies that are working with them and that set somehow should be compatible working together”-S1.

6.3.2 Collaboration on Software related aspects

The primary objectives of this question were to learn how industries interact on software-related topics. Different ways were identified how companies collaborate with software related issues. It was observed that businesses frequently work with their partners, such as vendors and service providers. A greater percentage (3 out of 4) of respondents from the manufacturing side stated that they collaborate with their partners or look for a partnership in which the partner would handle the software work. For instance, one interviewee expressed about collaboration quoted:

“…we see that further development requires different types of competences and we might make a partnership agreement with some supplier that they will take over the software and continue to develop that software”-M2

The other way the industries collaborate is with universities on research aspects. It was observed that a greater proportion of respondents stated that they work with universities but, in varying degrees. While some partnerships are continuous and last for a long time, others are project-based and last just a short time until a mutual objective is achieved. For example, an interviewee addressed about working with universities on research projects quoted:

“Yes we have collaboration with universities but those comes from our research, we have research projects were we have, there is usually those universities and so on but not in other projects”-E2

The final aspect of engagement identified was with companies on research initiatives. This collaboration occurs in a way, where several companies collaborate with universities, to execute research programs or achieve some specific objectives. Two of the case participants stated that they do not frequently collaborate with firms on developing common software, but rather they search for cooperation where the other party can provide additional functions to the product, such as developing software parts for its hardware part. For example, a interviewee made the following remarked:

> *"The biggest things (collaborations) are these research projects that we are participating-PEAMS ad MORO project definitely. …so at the moment, we don not develop same software together with some other company."*

6.3.3 Benefits & Risk of collaborations

According to the interviewees, partnership or collaboration brings a different set of values and competencies. Companies are seeking to collaborate in order to achieve their objectives and gain mutual benefits. Most participants (7 out of 12) cited the sharing of best practices and ideas as the most significant outcome of collaboration. Collaborative effort generates innovative thinking and access to technology for deploying product utilization in many scenarios Sometimes, industries lack specific experience or competencies necessary to develop a product or application. Therefore, collaborating is one such strategy identified to achieve the goals through leveraging one another's skills. For instance, one of the interviewees from the manufacturing industry mentioned the benefits as:

> *"we can benefit from the different fields of expertise, as we cannot be the experts in every aspects to the detail level. for ex, with these motor controllers we get the benefit from the guys who know most of how to control different semiconductor in most efficient way and how to do robust code for their platform its important for us we get that part from them"-M3*

The other potential benefit identified was sharing of the investment cost when industries collaborate with each other. It can also be a source of providing additional capital resources through collaboration. Combining more resources might just not give an opportunity to reduce the cost of development but also the amount of time required. Two of the participants mentioned that collaboration frequently delivers faster outcomes since the task may be split among the various expertise. One of the respondents emphasized the benefit of accessing non-domain markets and expanding the business networks. Collaboration enables growth and support in entering markets that are difficult to access as an individual. Below are a few of the statements addressed by the interviewees on the benefits of access through collaboration with companies:

> *"When making joint offerings with other companies we can participate in bigger projects then we could otherwise do or sometimes if the other company has some special domain expertise, then we can go to a domain where we could not go alone."-S1*

"we work collaboratively in many cases we can achieve the results much faster and then ofcourse when you can combine resources with someone else it could save you also the development cost and time."-E3

The benefits of collaboration extended beyond working with companies to include collaborating with academics as well. Working with universities gives businesses access to cutting-edge technology and innovative concepts. Universities have a wealth of knowledge and expertise in a variety of fields that industries may utilize to assess new developments and conduct collaborative research. For instance, one of the statements by an interviewee is quoted below:

"we are trying to benchmark and find new ways how to do things and so on and that's one thing why we are doing together with universities"-M4

The other benefit identified with collaborating with the universities is access to new talents. By having access to a talent pool of untapped individuals with both theoretical and practical expertise, industries might get over the obstacle of attracting new hire.

Table 6. Benefits and risks identified of collaborative working.

Collaboration with …	**Benefits**	**Risks**
Companies	• Sharing of best practices (S3, S4, E1, E2, M1, M2 M3) • Sharing of cost (S4, E3) • Faster results (E3, M3) • Access to non-domain & new networks (S1)	• Project failure & economic losses (S1, S2, E3, M4) • Multiple Companies makes it difficult to capture needs (E1, S4, M3) • Dependency on others (S2, M3, E1, S3)

Universities	• Access to emerging technology & innovative concepts (S1, S3, E1, S4, M1, M3, M4) • Hiring new resources (M4)	• Research quitting the project (E3, M3) • Publishing research results (M3)

Risks of collaboration

Though there are several benefits to collaboration, it does not come without risks. The risk of project failure was highlighted by a few of the interviewees. Two reasons were identified for project failure; one because the outcomes are not obtained, and the other because the counterpart is failing to execute their part. The project failure further leads to economic loss. The economic loss includes not only the capital invested, but also the resources, time, and development effort employed on the project. For instance, an interviewee stated the risk of collaboration as quoted:

> *"it could be that the risks is you dont get the results that you want so if you are requirement are not aligned then ofcourse it could be that end results is not something that you expected"-E3*

Then few of the interviewees mentioned the risk of having multiple companies on board. Too many organizations working collaboratively generate too many viewpoints, making it harder to reach conclusions. One respondent stated that it is difficult to maintain the balance of distributing tasks owing to the risks of disagreement regarding who should do what. The other risk identified was the risk of depending on others to perform their task. The factors identified in this dependency were quality, time, and trust. Quality refers to whether the copartner's contribution meets the expected standard and is completed within the agreed time frame. The trust factor is whether or not the businesses are recognized for their efforts and investments in the project. For example, the statement provided by an interviewee is quoted:

> *"the biggest risk that are that you rely on someone else and it depends again that how well can you trust your partner to do their part"-S2*

The risk associated from collaborating with universities were not regarded as a major issue by the interviewees. One of the risks highlighted was the researcher quitting the project and getting recruited by other companies. The other concern identified related to agreeing on publishing the research results. Table 7 summarizes the collaboration benefits and risks identified during the interviews. The table also provides the data supported by different interviewees on the benefits and risks.

6.4 Robot Operating System

6.4.1 Knowledge about Robot operating system

The Robot Operating System (ROS) has been there for a while and has widely been used for research and academic purposes. Nevertheless, it was observed in this study that some of the interviewees were not much aware about ROS. Whereas other interviewees from different industries had differing viewpoints of what it is. First, 3 out of 12 interviewees mentioned ROS as a communication interface. In this regard, ROS helps to communicate different software with each other. For instance, one interviewee stated ROS as communication middleware, as cited:

> *"in some cases we have to integrate different subsystems to our platform and it could be that this system already have some kind of functionality that need to be able to communicate with the vehicle platform so we see that ROS is a interface for that"-E3*

Next, 3 out of 12 interviewees considered ROS as a platform or an ecosystem, that is, an open platform that includes software libraries and development tools for the robotics and autonomous domains. Furthermore, it has a consortium of members who innovate and maintain this platform. The community around this platform includes many players such as industries, individuals, groups, and academics that contribute to the ecosystem. One respondent emphasized its growing use in the Automated Guided Vehicle (AGV) area, where corporations rely on ROS. For example, an interviewee expressed his opinion about ROS as:

> *"I personally do have rather good technical view on the different building blocks of the ROS, but for ex, that its actually not an operating system but it's a platform or an ecosystem."-E1*

Finally, most of the interviewees, 6 of them, stated ROS as a technological platform or a technological base upon which further software or functionalities can be developed. One of the interviewees mentioned using ROS in different domains, such as robotics, mobile work machinery, and process automation. The platform offers technological characteristics such as the ability to build codes for onboard units on machines, which makes it rather simple. For instance, one of the interviewees provided his viewpoint as:

“ROS 2 specially is proving to be a matured technology, efficient technology and a technology that is supported by many vendors so that's why we want to understand it”-S3

6.4.2 Opportunities of ROS Ecosystem

As discussed in the previous chapter, there do not appear to be much common understanding of ROS among the industries. That implies, that the opportunities they recognize, from the ROS ecosystem, are also varied. An important observation was that most of the case companies (9 out of 12) did not have long experience with ROS. Many of the interviewees suggested that their familiarity with ROS is due to their association with the PEAMS project at the university, which is relatively new. Nonetheless, the interviewees suggested a few potential opportunities that are a result of being a part of ROS ecosystem.

First, 4 out of 12 interviewees suggested that ROS provides an opportunity to develop the software inhouse. It offers a chance to replace the option of purchasing the software since it comes with the software packages needed to develop the application. Furthermore, one respondent stated that it may help businesses avoid vendor lock-in. For example, one of the interviewees stated the use of ROS as:

“…ROS is more a flexible platform we can use it to run more advance analytics and algorithm & it enables you to quite rapidly develop new functionalities and test them”-M1

Next, the ROS ecosystem makes the integration work much simpler and faster. One of the respondents commented that there is more development effort involved with developing the integration of any subsystems in a vehicle. In such circumstances, employing ROS simplifies the integration process, saving time and resources. Finally, the ROS ecosystem decreases the barrier to entry into the world of robotics and automation. Building a robot or automating an application requires a significant amount of effort and money since all of the software must be developed from scratch. But the open architecture of the ROS ecosystem reduces the development efforts to build such software. For instance, an interviewee mentioned:

“ROS lowers the barriers to market entry… from ROS you can let say get 80% of your software already build and you just have to add the missing 20% that you need for your robot and your application and that hugely decreases the initial barrier or threshold of entry”-S1

7. CONCLUSIONS

The master's thesis investigated how the heavy-duty mobile machine industry views a platform-based ecosystem. According to Yin [23], using multiple case studies and drawing logical inferences leads to findings that represent substantial topics or concerns. In that context, the final chapters attempt to answer the research questions. Furthermore, the subsequent sections describe the study's theoretical and practical contributions, as well as the research limits and future potential.

Answers to the Research Questions

RQ 1: What do industries think about platform-based ecosystem, and ROS ecosystem in particular?

To answer this question, the researcher conducts empirical research to better understand industrial perspectives on software platforms and ecosystems. Based on the actual findings, the industries have various perspectives on the platform and ecosystem, which correspond to the findings from literature research. It could be observed from the literature, that the definitions of platform and ecosystem have evolved throughout time. For instance, software platforms are used as a basis or reused across many applications, or they provide software tools and libraries that may be utilized to develop various applications. A similar understanding is with the software ecosystem, where some see it as a marketplace of developers contributing or else different players contributing towards a common technological platform. A similar, varying, understanding is seen across multiple companies that are part of the heavy-duty mobile machine industry.

In terms of the benefits of adopting software platforms, particularly open-source ones, it is difficult to evaluate if the benefits outweigh the risks. Indeed, whether open source or closed platforms are used, they make it simple to add new functionality on top of them, saving time and resources. There are benefits such as cost sharing and high quality, but the most significant observation was its promising future. According to literature, the ecosystem is not just the platform or the software, but it is more about the network and relations between its participants. According to empirical data, when it comes to collaboration, it is more like give-and-take relationship where organizations are seeking to obtain certain elements of their products or applications from each other. For instance, a software supplier may collaborate with a sensor manufacturer to verify the sensor's compatibility with their interface, offering input and updating the sensors as needed (Interviewee representing as a software provider, S2).

Regarding the ROS ecosystem, empirical findings suggest that companies in heavy-duty mobile machine industry are interested in learning more about it. The companies are eager to understand what ROS has to offer, how to pursue it, and what additional benefits it will bring to the table. It can be observed that, despite the existence of ROS for more than a decade, there is not yet a clear and common understanding of its characteristics and applications. This is most possibly because the companies are only recently getting to know and be associated with the ROS ecosystems. Nevertheless, the study revealed that currently, the companies have multiple opinions about the ROS ecosystem. For some, it serves more as a technological platform that will make it easier to develop and maintain technical solutions. Whereas, few see it more as a communication middleware in domains such as robotics, mobile machinery, and process automation. In addition, some see it as an ecosystem that not only provides the building blocks in different environments such as real-time or motion control, but also a network of actors collaborating together.

RQ 2: What opportunities can companies realize through ROS ecosystem?

According to the case interview findings, despite being unsure of the full potential of the ROS ecosystem, companies have pointed out a few possible opportunities that may be realized by adapting it. Based on the findings, the ROS platform simplifies the development work required to build a software application. The ecosystem does offer an opportunity to get a head start when industries look for developing new applications. It was observed that some manufacturing businesses are heavily dependent on a single provider or partner for software development. However, employing an open-source platform like ROS allows businesses to create their software in-house and avoid vendor lock-in.

Next opportunity identified is that the ROS ecosystem lowers the entry barriers for firms looking to enter the robotics or automation sector. Developing software for automation or autonomous operation necessitates a significant investment of capital and resources. Small-scale industries find it quite challenging to invest in such developments. However, with the support of free software tools and the knowledge that the ROS ecosystem provides, it is possible to enter this type of domain. This could, for instance, be observed among the members of the FIMA association and the PEAMS project, who are trying to utilize the ROS platform in different contexts.

The opportunity for collaboration offered by the ROS platform was another finding highlighted during the empirical analysis. The collaboration between the platform's members and the orchestrator is frequently mentioned in the literature. It is the orchestrator that always generates opportunities for its members to collaborate and work on new innovative research or projects. However, a horizontal collaboration between the participants of the PEAMS projects has been observed during this study. Being a part of a ROS ecosystem increases corporate value by indirectly motivating players in that ecosystem to innovate, collaborate, and co-invest to deliver the needs of the future. Based on the observations around the research study, ROS platform has served as a catalyst to encourage horizontal collaboration across the heavy-duty mobile machinery industry. Figure 16 summarizes the potential benefits that are identified from the research findings.

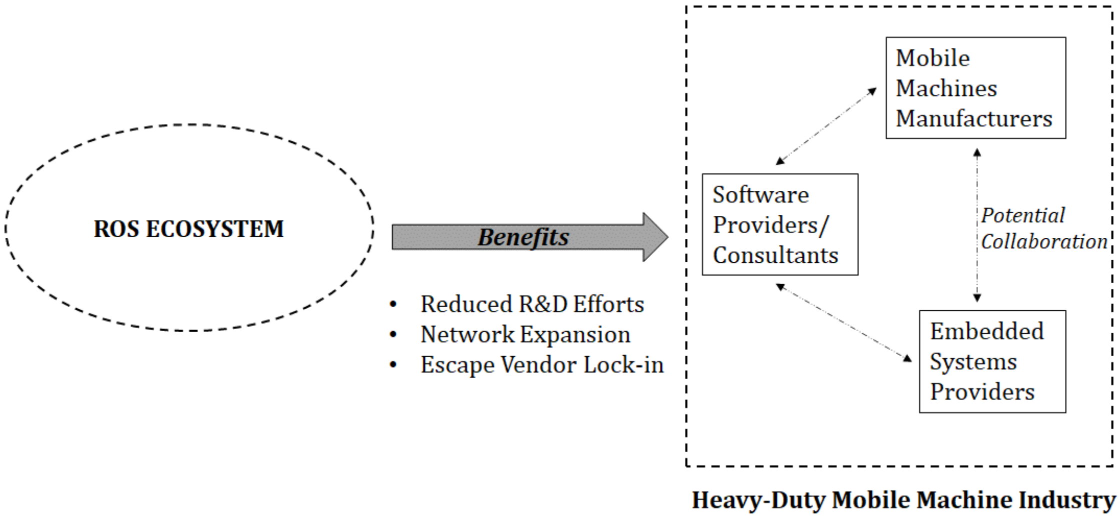

Figure 16. Significance of ROS ecosystem in heavy duty mobile machine industry

7.1 Theoretical Contributions

In recent years, there has been a steady increase in the amount of research conducted on the topics of platform and ecosystem architecture, with an immediate focus on understanding the core components of platform architecture and the influence of its networks [29, 46, 63]. This study tries to contribute to the existing literature on platform-based ecosystem and the heavy-duty mobile machine industry in multiple ways. First, it uses a wider and holistic approach to understand the perspectives of the heavy-duty mobile machine industry on the platform-based ecosystem. This study seeks to determine whether the different players in the heavy-duty mobile machine industry have common reasons and concerns towards the use of Software-platform based ecosystem and in particular, the ROS ecosystem. Furthermore, the research intends to incorporate several perspectives on the significance of the ROS ecosystem to the heavy-duty mobile machine industry.

Second, this study in heavy-duty mobile machine industry validates the existing literature that there is not yet a consensus among practitioners on what the software platform and ecosystem are. This study further highlights that the companies in the heavy-duty mobile machine industry lack a common understanding of what the ROS ecosystem means.

Third, this study is one of the first attempts to exclusively study the business implications of ROS ecosystem in context of heavy-duty mobile machine industry. The findings do highlight the interest of the companies in exploring the services offered by the ROS ecosystem. Though only a few of the companies have practically adopted the ROS platform, other companies are working together with universities to understand how the platform might benefit them. Surprisingly, despite the widespread use of the ROS platform in various industries [40, 45, 53], the heavy duty mobile machine industry in this study did not recognize the ROS ecosystem as a potential source of new business opportunities.

Finally, this study contributes to the existing literature on the software ecosystem by proposing a new role- support coordinator. This role normally undertaken by different actors, such as universities and research groups, has not been explicitly recognized in the previous literature. Under this support coordinator role, the actors indirectly contribute to the ecosystem by encouraging collaborative innovation and disseminating knowledge awareness.

7.2 Practical Contributions

This study highlights a few important practical implications for companies in the heavy-duty mobile machine industry. The findings of this study will help the companies to gain a comprehensive understanding on the benefits one can receive by utilizing Software-platform based ecosystem, especially the free and open-source ROS ecosystem. The findings particularly highlighted the benefits such as simplifying software development and speeding up the implementation process by minimizing the resources and efforts needed for developing new software or applications in different contexts.

Additionally, this study shows that the applications of ROS platform is not limited to pure robotics [53], but it does have the capabilities that could be utilized in the heavy-duty mobile machinery context. This has been observed from the interest of the companies coming from different backgrounds and collaborating with the universities on the ROS related topics. On the other hand, one novel finding that could be observed in this study was the horizontal collaboration seen across the players in the industry. However, it is important that the players are open to recognize and collaborate with other players who have the complementary capabilities.

7.3 Research Limitations and Future topics

Unlike every other qualitative research, the results of this research are based on different perspectives of the researcher and the interviewees that might suffer from their subjectivity [58]. First, the qualitative findings are based on a small number of participants and a limited geographical range. The sample size remains fairly small due to time restrictions and the researcher's limited access to relevant case companies. The case companies primarily belong to the Finland region, and though few of them were operating globally, this might undermine the opinions and the generalizability. Although the literature does not specify any limited number of cases to be conducted [66], the purposive sampling used for the case selection would be enough to validate the findings and contributions.

Second, a few company representatives were not very knowledgeable about the ROS platform, and more than half of the case companies were yet to fully understand its potential. Based on the understanding of empirical findings companies are still seeking a complete understanding and how it can add value to the business. Additionally, the lack of literature in the context of heavy-duty mobile machine industry has been limited. Therefore, the perspectives on the ROS ecosystem may not be entirely based on the practicality of using the ROS platform.

Finally, the semi-structured interview theme may exclude some essential subjects that might give useful insights into the research findings. Furthermore, there may be a biased understanding among the interviewees, and each individual may seek a different understanding than the others.

Based on the findings and limitations discussed, there is a future scope to study the significance of the ROS ecosystem in the heavy-duty mobile machine industry on a larger scale. This will assist in understanding more functional capabilities achieved through the ROS platform and will help organizations make effective decisions to achieve their objectives. Second, a more detailed questionaries, specifically linked to the ROS ecosystem, could be addressed to the respondents in order to better grasp the potential benefits and challenges on a more practical level. The current study recognized the lack of complete understanding of the services or the capabilities that ROS has to offer particularly in the context of heavy-duty mobile machine industry. Consequently, research further in this area should be a logical extension of ongoing research.

Finally, the future research could also seek to investigate whether ROS ecosystem could create any new business opportunities for the companies operating in the heavy-duty mobile machine industry domain. This study pointed out that most of the case companies presently do not perceive any such new business opportunities, although this might be because the industry is only recently becoming aware of the ROS ecosystem. It would be quite interesting to know what valuable insights does ROS ecosystem, especially the ROS 2 capabilities could offer and the findings that could improve organizations of all sizes.

REFERENCES

1. Adhabi, E., Anozie, C.: Literature Review for the Type of Interview in Qualitative Research. Int. J. Educ. 9, 86 (2017). https://doi.org/10.5296/ije.v9i3.11483.
2. Archibald, M. et al.: Using Zoom Videoconferencing for Qualitative Data Collection: Perceptions and Experiences of Researchers and Participants. Int. J. Qual. Methods. 18, 160940691987459 (2019). https://doi.org/10.1177/1609406919874596.
3. Barbosa, F. et al.: Reinventing construction: A route of higher productivity. McKinsey Global Institute (2017).
4. Belo, Í., Alves, C.: How to Create a Software Ecosystem? A Partnership Meta-Model and Strategic Patterns. Inf. Basel. 12, 6, 240- (2021). https://doi.org/10.3390/info12060240.
5. Bosch, J.: From software product lines to software ecosystem. Presented at the Proceedings of the 13th International Software Product Line Conference January 1 (2009). https://doi.org/10.1145/1753235.1753251.
6. Bosch, J.: Maturity and Evolution in Software Product Lines: Approaches, Artefacts and Organization. Presented at the , Berlin, Heidelberg (2002). https://doi.org/10.1007/3-540-45652-X_16.
7. Bosch, J., Bosch-Sijtsema, P.: From integration to composition: On the impact of software product lines, global development and ecosystems. J. Syst. Softw. 83, 1, 67–76 (2010). https://doi.org/10.1016/j.jss.2009.06.051.
8. Boyatzis, R.E.: Transforming qualitative information: thematic analysis and code development. Sage, Thousand Oaks (1998).
9. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual. Res. Psychol. 3, 2, 77–101 (2006). https://doi.org/10.1191/1478088706qp063oa.
10. Capilla, R. et al.: On Autonomous Dynamic Software Ecosystems. IEEE Trans. Eng. Manag. 1–15 (2021). https://doi.org/10.1109/TEM.2021.3116873.
11. Chitta, S. et al.: ros_control: A generic and simple control framework for ROS. J. Open Source Softw. 2, 20, 456–456 (2017). https://doi.org/10.21105/joss.00456.
12. Coleman, D. et al.: Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study. (2014).
13. Curran, W. et al.: Evaluating impact in the ROS ecosystem. Proc. - IEEE Int. Conf. Robot. Autom. 2015, 6213–6219 (2015). https://doi.org/10.1109/ICRA.2015.7140071.
14. Dadhich, S. et al.: Key challenges in automation of earth-moving machines. Autom. Constr. 68, 212–222 (2016). https://doi.org/10.1016/j.autcon.2016.05.009.
15. Emmel, N.: Sampling and Choosing Cases in Qualitative Research: A Realist Approach. SAGE Publications, Limited, London (2013). https://doi.org/10.4135/9781473913882.
16. Evans, D.S. et al.: Invisible engines: how software platforms drive innovation and transform industries. MIT Press, Cambridge, Mass (2006).
17. FIMA: FIMA, https://www.fima.fi/about/, last accessed 2022/07/26.
18. Flick, U.: The Sage handbook of qualitative data collection. SAGE Publications Ltd, London (2018).
19. Gale, N.K. et al.: Using the framework method for the analysis of qualitative data in multidisciplinary health research. BMC Med. Res. Methodol. 13, 1, 117–117 (2013). https://doi.org/10.1186/1471-2288-13-117.
20. Geimer, M.: Mobile working machines. SAE International, Warrendale, PA (2020).
21. Gill, P. et al.: Methods of data collection in qualitative research: interviews and focus groups. Br. Dent. J. 204, 6, 291–295 (2008). https://doi.org/10.1038/bdj.2008.192.
22. Gillham, B.: Case Study Research Methods. Bloomsbury Publishing Plc, London, UNITED KINGDOM (2000).
23. Green, J.L. et al.: Handbook of Complementary Methods in Education Research. Taylor & Francis Group, Florence, UNITED STATES (2006).
24. Haenssgen, M.J.: Interdisciplinary Qualitative Research in Global Development: A Concise Guide. Emerald Publishing Limited, Bingley (2019).
25. Haycraft, W.R.: History of Construction Equipment. J. Constr. Eng. Manag. 137, 10, 720–723 (2011). https://doi.org/10.1061/(ASCE)CO.1943-7862.0000374.
26. Jansen, S. et al.: A sense of community: A research agenda for software ecosystems. Presented at the (2009). https://doi.org/10.1109/ICSE-COMPANION.2009.5070978.

27. Joshi, R. et al.: Software Development Platforms for Autonomous Systems. Presented at the August 5 (2002). https://doi.org/10.2514/6.2002-4484.
28. Joshua, J.V. et al.: Software Ecosystem: Features, Benefits and Challenges. Int. J. Adv. Comput. Sci. Appl. IJACSA. 4, 8, (2013). https://doi.org/10.14569/IJACSA.2013.040833.
29. Kapoor, K. et al.: A socio-technical view of platform ecosystems: Systematic review and research agenda. J. Bus. Res. 128, 94–108 (2021). https://doi.org/10.1016/j.jbusres.2021.01.060.
30. Kato, S. et al.: An Open Approach to Autonomous Vehicles. IEEE Micro. 35, 6, 60–68 (2015). https://doi.org/10.1109/MM.2015.133.
31. Kenney, M. et al.: Platforms and industrial change. Ind. Innov. 26, 8, 871–879 (2019). https://doi.org/10.1080/13662716.2019.1602514.
32. Knodel, J., Manikas, K.: Towards a Typification of Software Ecosystems. In: Fernandes, J.M. et al. (eds.) Software Business. pp. 60–65 Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-19593-3_5.
33. Knüchel, C. et al.: Artop – an ecosystem approach for collaborative AUTOSAR tool development. In: ERTS2 2010, Embedded Real Time Software & Systems. , Toulouse, France (2010).
34. Kolak, S. et al.: It Takes a Village to Build a Robot: An Empirical Study of The ROS Ecosystem. In: 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME). pp. 430–440 (2020). https://doi.org/10.1109/ICSME46990.2020.00048.
35. Lee, S.M., Lim, S.: Living Innovation: From Value Creation to the Greater Good. Emerald Publishing Limited, Bingley (2018).
36. Liamputtong, P.: Qualitative Data Analysis: Conceptual and Practical Considerations. Health Promot. J. Austr. 20, 2, 133–139 (2009). https://doi.org/10.1071/HE09133.
37. Liu, S. et al.: Engineering Education in the Age of Autonomous Machines. Computer. 54, 4, 66–69 (2021). https://doi.org/10.1109/MC.2021.3057407.
38. Lungu, M.: Towards reverse engineering software ecosystems. In: 2008 IEEE International Conference on Software Maintenance. pp. 428–431 (2008). https://doi.org/10.1109/ICSM.2008.4658096.
39. Ma Cecilia C Carlos, Honey Jericah D Lucero: Risk Assessment of the Job Tasks for Heavy Equipment Operators. IIE Annu. Conf. Proc. 1- (2012).
40. Macenski, S. et al.: Robot Operating System 2: Design, architecture, and uses in the wild. Sci. Robot. 7, 66, eabm6074 (2022). https://doi.org/10.1126/scirobotics.abm6074.
41. Machado, T. et al.: Autonomous Heavy-Duty Mobile Machinery: A Multidisciplinary Collaborative Challenge. Presented at the , Piscataway (2021). https://doi.org/10.1109/ICTE51655.2021.9584498.
42. Machado, T. et al.: TOWARDS A STANDARD TAXONOMY FOR LEVELS OF AUTOMATION IN HEAVY- DUTY MOBILE MACHINERY. 10 (2021).
43. Manikas, K.: Revisiting software ecosystems Research: A longitudinal literature study. J. Syst. Softw. 117, 84–103 (2016). https://doi.org/10.1016/j.jss.2016.02.003.
44. Manikas, K., Hansen, K.M.: Software ecosystems – A systematic literature review. J. Syst. Softw. 86, 5, 1294–1306 (2013). https://doi.org/10.1016/j.jss.2012.12.026.
45. Mayoral-Vilches, V. et al.: Can ROS be used securely in industry? Red teaming ROS-Industrial. (2020).
46. McIntyre, D.P., Srinivasan, A.: Networks, platforms, and strategy: Emerging views and next steps: Networks, Platforms, and Strategy. Strateg. Manag. J. 38, 1, 141–160 (2017). https://doi.org/10.1002/smj.2596.
47. Media, O.: Heavy Mobile Equipment: Transforming Industries, http://embeddedcomputing.com/application/automotive/adas-autonomous-drive/heavy-mobile-equipment-transforming-industries, last accessed 2022/09/09.
48. Messerschmitt, D.G.: Software ecosystem: understanding an indispensable technology and industry. MIT Press, Cambridge, Massachusetts (2003).
49. Palinkas, L.A. et al.: Purposeful sampling for qualitative data collection and analysis in mixed method implementation research. Adm. Policy Ment. Health. 42, 5, 533–544 (2015). https://doi.org/10.1007/s10488-013-0528-y.
50. Parida, V. et al.: Reviewing Literature on Digitalization, Business Model Innovation, and Sustainable Industry: Past Achievements and Future Promises. Sustainability. 11, 391 (2019). https://doi.org/10.3390/su11020391.
51. Patton, M.Q.: Qualitative research & evaluation methods. Sage, Thousand Oaks (Calif.) (2002).

52. Pohl, K. et al.: Software Product Line Engineering Foundations, Principles and Techniques. Springer Berlin Heidelberg, Berlin, Heidelberg (2005). https://doi.org/10.1007/3-540-28901-1.
53. Quigley, M. et al.: ROS: an open-source Robot Operating System. 6.
54. Rantala, A. et al.: Inter-firm software development: the case of mobile work machines. Int. J. Value Chain Manag. 13, 2, 197–216 (2022). https://doi.org/10.1504/IJVCM.2022.123550.
55. Rausch, A. et al.: From Software Systems to Complex Software Ecosystems: Model- and Constraint-Based Engineering of Ecosystems. In: Perspectives on the Future of Software Engineering: Essays in Honor of Dieter Rombach. pp. 61–80 (2013). https://doi.org/10.1007/978-3-642-37395-4_5.
56. Saito, Y. et al.: Priority and Synchronization Support for ROS. In: 2016 IEEE 4th International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA). pp. 77–82 (2016). https://doi.org/10.1109/CPSNA.2016.24.
57. Sami, H. et al.: A Review of Ethical Discussions on Platforms and Ecosystems. CEUR-WS. 11 (2019).
58. Saunders, M. et al.: “Research Methods for Business Students” Chapter 4: Understanding research philosophy and approaches to theory development. Presented at the March 1 (2019).
59. Schmidt, D.C., Buschmann, F.: Patterns, frameworks, and middleware: their synergistic relationships. In: 25th International Conference on Software Engineering, 2003. Proceedings. pp. 694–704 IEEE, Portland, OR, USA (2003). https://doi.org/10.1109/ICSE.2003.1201256.
60. Sjödin, D. et al.: Smart Factory Implementation and Process Innovation: A Preliminary Maturity Model for Leveraging Digitalization in Manufacturing. Res. Technol. Manag. 61, 22–31 (2018). https://doi.org/10.1080/08956308.2018.1471277.
61. Sniderman, B. et al.: Deloitte-Industry-4-0-and-manufacturing-ecosystems.pdf, https://www2.deloitte.com/content/dam/Deloitte/de/Documents/consumer-industrial-products/Deloitte-Industry-4-0-and-manufacturing-ecosystems.pdf, (2016).
62. Thomson, L. et al.: A maturity framework for autonomous solutions in manufacturing firms: The interplay of technology, ecosystem, and business model. Int. Entrep. Manag. J. 18, 1, 125–152 (2022). https://doi.org/10.1007/s11365-020-00717-3.
63. Tiwana, A. et al.: Research Commentary—Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. Inf. Syst. Res. 21, 4, 675–687 (2010). https://doi.org/10.1287/isre.1100.0323.
64. Williams, M., Moser, T.: The Art of Coding and Thematic Exploration in Qualitative Research. Int. Manag. Rev. 15, 1, 45–72 (2019).
65. Wyrobek, K.: The Origin Story of ROS, the Linux of Robotics, https://spectrum.ieee.org/the-origin-story-of-ros-the-linux-of-robotics, last accessed 2022/08/04.
66. Yin, R.K.: Case study research: design and methods. Sage, Thousand Oaks (Calif.) (2003).
67. Yu, E., Deng, S.: Understanding Software Ecosystems: A Strategic Modeling Approach. CEUR Workshop Proc. 746, (2011).
68. Zainal, Z.: Case study as a research method. 6 (2007).
69. A History of ROS (Robot Operating System), https://www.theconstructsim.com/history-ros/, last accessed 2022/09/11.
70. A literature review on the levels of automation during the years. What are the different taxonomies that have been proposed? | Elsevier Enhanced Reader, https://reader.elsevier.com/reader/sd/pii/S0003687015300855?token=946F61579A786EA1DCEE058177D9F42C0A1049F0A1B824FFD72D0CE45317E1A6957364E1938A10E5D5C9CE8263273F13&originRegion=eu-west-1&originCreation=20220909110634, last accessed 2022/09/09. https://doi.org/10.1016/j.apergo.2015.09.013.
71. ATLAS.ti | The Qualitative Data Analysis & Research Software, https://atlasti.com, last accessed 2022/10/18.
72. How Software Platforms Revolutionize Business, http://hbswk.hbs.edu/item/how-software-platforms-revolutionize-business, last accessed 2022/09/17.
73. Pinterest, https://in.pinterest.com/, last accessed 2022/09/09.
74. Platform Economy for Autonomous Mobile Machines Software Development, https://www.fima.fi/projects/platform-economy-for-autonomous-mobile-machines-software-development/, last accessed 2022/09/09.

75. Project Governance — ROS 2 Documentation: Rolling documentation, https://docs.ros.org/en/rolling/The-ROS2-Project/Governance.html, last accessed 2022/08/30.
76. ROS: The ROS Ecosystem, https://www.ros.org/blog/ecosystem/, last accessed 2022/08/25.
77. ROS: Welcome to the ROS Community, https://www.ros.org/blog/community/, last accessed 2022/08/30.
78. ROS/Tutorials - ROS Wiki, http://wiki.ros.org/ROS/Tutorials, last accessed 2022/09/11.
79. The case for an automotive software platform | McKinsey, https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-case-for-an-end-to-end-automotive-software-platform, last accessed 2022/09/10.
80. Types of Software Platforms, https://www.geeksforgeeks.org/types-of-software-platforms/, last accessed 2022/09/10.
81. Vehicle Certification Agency, https://www.vehicle-certification-agency.gov.uk/, last accessed 2022/09/09.
82. https://www.openrobotics.org/company, Last accessed 2022/09/01.