CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Applicability of offshore software development best practices to AI-assisted software development

Master's thesis in Computer science and engineering

Ebba Ljung
Oliver Ljung

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2024

Master's thesis 2024

Applicability of offshore software development best practices to AI-assisted software development

EBBA LJUNG
OLIVER LJUNG

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Applicability of offshore software development best practices to AI-assisted software development

EBBA LJUNG
OLIVER LJUNG

Supervisor: Krishna Ronanki, Department of Computer Science and Engineering
Examiner: Farnaz Fotrousi, Department of Computer Science and Engineering

Master's Thesis 2024
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2024

Applicability of offshore software development best practices to AI-assisted software development

EBBA LJUNG
OLIVER LJUNG
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

In this study we explored the similarities between offshore software development and Artificial Intelligence (AI)-assisted software development. Given the current research gap regarding best practices for AI-assisted software development, this thesis aimed to explore the applicability of offshore software development best practices to AI-assisted software development. By gathering challenges within both areas through a literature review and using framework analysis, we were able to determine similarities between the two areas. With structured interviews we were then able to use the identified similarities to further examine the challenges and determine the applicability of offshore software development best practices to AI-assisted software development. The findings revealed three shared challenges: IP theft, code privacy, and tool incompatibility. Additionally, one best practice, confidentiality agreements, was determined to be directly applicable to AI-assisted software development based on the responses of participants who regularly use AI assistants for coding. The insights gained from this thesis provide valuable guidance for industry practitioners and contribute to further academia on optimising AI-assisted software development practices.

Keywords: Software, AI, offshore, challenges, best practices, software development, thesis.

Acknowledgements

We would like to express our sincere gratitude to our supervisor, Krishna Ronanki, for his invaluable guidance, support, and encouragement throughout the duration of this project. His expertise, insightful feedback, and unwavering dedication have been crucial in shaping this work.

Ebba Ljung & Oliver Ljung, Gothenburg, June 2024

Contents

List of Figures

List of Tables

1
Introduction

In the current day, artificial intelligence (AI) is an ever-evolving technology with the potential to transform a large number of fields [1]. AI and machine learning is currently used in the field of software engineering for a wide array of tasks [2]. One area within software engineering where AI has recently become relevant, is AI-assisted software development, with tools like GitHub Copilot and Amazon CodeWhisperer releasing in 2021 and 2023, respectively [3], [4]. These tools and others of similar nature can be used as coworkers for software developers, similar to pair programming [5]. They provide a multitude of benefits, such as improved code quality, higher productivity and cost benefits [6].

Using AI-assisted software development is beneficial in many ways but it also poses a number of challenges and limitations. These include AI assistants offering a low variety of solutions which limits code generation, generated code sometimes having poor quality and potential threats to privacy of the code when generating it with the assistance of AI [6].

With AI evolving and AI programming assistants launching, it is important to establish best practices [7] for AI-assisted software development to mitigate the challenges and limitations [8]. Our hypothesis is that there are some similarities between AI-assisted software development and offshore software development and these similarities can be used to investigate if best practices from offshore software development are applicable to AI assisted software development.

1.1 Aim of the study

The purpose of the study is to collect and present to what extent AI-assisted software development is similar to offshore software development. The evidence will be based on observed similarities between the two processes and the similarities between their challenges. This aims to evaluate if strategies employed in addressing challenges in offshore software development are applicable to the context of AI-assisted software development. The findings of this study are potentially beneficial for companies interested in incorporating AI assistants into their software development activities, as best practices are important [7]. This study also aims to further academic research by proposing best practices for working with AI-assisted software development.

1.2 Objectives

The study aimed to answer three main research questions. The first and second research questions, RQ1 and RQ2, were answered through a literature review covering offshore software development and AI-assisted software development. RQ2 was then further validated by conducting structured interviews with domain experts. The interviews also served to answer the third and final research question, RQ3,

RQ1: What are the similarities between AI-assisted software development practices and offshore software development practices?

RQ2: What are the shared challenges between offshore software development and AI-assisted software development?

RQ3: What offshore software development best practices are applicable to AI-assisted software development?

1.3 Delimitations

When determining applicability of best practices for AI-assisted software development, this study mainly focuses on third party AI tools as opposed to locally developed and maintained AI assistants. The study did not look at the effects an AI can have on humanity, both economic and psychological.

1.4 Thesis outline

The structure of the thesis is as follows:

Chapter 1 gives an introduction to AI-assisted software development, presents the thesis aim and its research questions and finally describes the thesis delimitations.

Chapter 2 presents background information about offshore software development and AI-assisted software development. It also covers related research within the same two topics.

Chapter 3 gives a detailed description about the scientific methods used to carry out the literature review, interviews and data analysis.

Chapter 4 presents the results of the data gathered from the literature review and interviews.

Chapter 5 contains a discussion of the results, threats to the validity and future work.

Chapter 6 summarises research findings, discussion and the potential for future work.

2
Theory

This section presents the necessary background information about the concepts and models discussed in this thesis as well as related works.

2.1 Background

This section provides an overview of two fundamental concepts discussed in this thesis: offshore software development and AI-assisted software development.

2.1.1 Offshore software development

Offshore software development is the practice of using an onsite development team and a remote development team located in another country, that work together to create one product [9]. In the software development industry it is common to use offshore software development teams as a resource when developing new software [10]. Alongside offshore software development, other similar outsourcing models exist such as nearshore software development and onshore software development. The main functional difference between the three models is the location of the remote party [11]. In onshore software development, the remote development team is located within the same country as the onsite development team. For both nearshore software development and offshore software development, the remote development team is located in another country from the onsite development team. The differentiating factor between the two are the location of the country in which the remote team operates. In nearshore software development, the remote team's country will share a similar time-zone to the country where the onsite team is located. In offshore software development, the remote team's country will have a large time zone-difference to the onsite country [11].

There are many reasons why companies make use of offshore software development. One of these reasons is the wider array of competencies this practice allows for. A company using offshore software development is not restricted to to their own country's supply of qualified workers as they can make use of a larger supply which exist elsewhere [12]. Another reason is that the usage of offshore software development allows companies to produce their software at a lower cost by employing teams in countries with lower wages rather than individuals from their own country.

While a number of benefits to offshore software development exist, so does a number of challenges [12]. In addition to these challenges there also exist best practices for mitigating them. The challenges and best practices of offshore software development will be explored further later in this thesis.

2.1.2 AI-assisted software development

Artificial intelligence, also referred to as AI, is a concept that originated from fantasies and fiction, describing intelligent machines [13]. Over time, this concept has become reality, with current-day AI being able to perform a multitude of tasks in various fields that previously required human intelligence [13]. The most relevant of these fields for the purpose of this thesis is the potential use cases of AI in software engineering.

In recent years, AI with the ability to generate code has become readily available to the general public. AI assistants, like GitHub Copilot, can already successfully complete easier programming tasks on its own [14] and be used in the software development industry [15]. One of the potential ways in which companies could utilise AI assistants in software development is through pair-programming between onsite software engineers and the AI assistant. Effectively utilising AI assistants in this manner would require a set of best practices to be established. Current research on best practices for AI-assisted software development is sparse and the identification of said best practices would aid its continued development and use.

2.2 Related work

2.2.1 Offshore software development

Previous literature has found challenges and risks associated with offshore software development. Some of the mentioned challenges are "'language and cultural barriers', 'country instability', 'lack of project management', 'lack of protection for intellectual property rights' and 'lack of technical capability'" [16], [17], [18]. Best practices for offshore software development to manage these challenges have already been established [19]. One of the current best practices for using offshore software development teams in the development of new software appears to be the model proposed by [9]. It proposes that the offshore team is not a completely separate team but instead an "extension of the development team". Concept creation, analysis and design are the phases which should be mostly performed onsite while construction (coding, unit tests and code review) should be done offshore. A part of the offshore team is included in the planning, analysis and design phases to ensure that they can then carry out the construction and quality assurance phases with the same intent and vision as the onsite team had.

2.2.2 AI-assisted software development

One of the main challenges with integrating AI into software development is that common practices, such as agile models, need to be somewhat altered to fit this new modern take on software development [20]. The same literature also contains proposals for creating such alterations and seem positive that AI can be helpful in software development if used correctly. In previous literature it has been found that current AI-assisted tools may not produce the same code quality as human pair programming [21]. This challenge is similar to that of offshore software development and can be an indicator of more similarities in challenges between the two. Similarly to the risks found in offshore software development, some challenges regarding security have been found when using an AI-assistant in software development [21] which further proves the similarities between AI assistant software development and offshore software development. However, these similarities are yet to be studied to determine their full extent. To the best of our knowledge no best practices have been produced for AI-assisted software development as of February 2024.

3
Methods

This section presents the methods used for the literature review and interviews to address the research questions. The data gathered from the literature review was used to design the interviews. The interviews were then used to collect additional data to answer all three research questions.

3.1 Literature review

The literature review was conducted through a literature search using a qualitative approach. The goal of the literature review was to identify three things, *AI-assisted software development challenges*, *offshore software development challenges* and *offshore software development best practices*. Two separate literature reviews were conducted, one covering both offshore software development challenges and best practices and one covering AI-assisted software development. Offshore software development best practices and challenges were reviewed together because most literature which covered one of the two topics also covered the other.

3.1.1 Data collection

When collecting data for the literature review, Google Scholar was chosen because of its features showing citations and filtering by date of publication. The literature review made use of five different search-strings. The two search-strings used to find challenges and best practices for offshore software development can be found in table 3.1 and the three ones used to find literature on AI-assisted software development challenges can be found in table 3.2.

Table 3.1: Search strings used in the literature review to find offshore literature

	Offshore search strings
1	("offshore software development" AND "challenges")
2	("offshore software development" AND "best practices")

Table 3.2: Search strings used in the literature review to find literature on AI-assistants

	AI assistant search strings
1	("AI" AND "software development" AND "challenges")
2	("AI assisted" AND "software development" AND "challenges")
3	("copilot" AND "software development" AND "challenges")

For literature regarding AI-assisted software development it was necessary to use a variety of different search-terms because of the many names used to refer to the topic. Other than *AI-assisted* and *software development* which yielded very specific results *AI* was also used to ensure a wider range was covered.
Backwards and forwards snowballing was also performed on all found literature. When a paper was found through Google Scholar it was then also used for snowballing, when possible. Since the literature used in our study had to relevant to the topics, all literature used for snowballing was relevant to the study. Most of the offshore software development literature used for snowballing was highly cited indicating its influence and recognition in the field. This combined method of searching on Google Scholar and snowballing was conducted to optimise the results of the literature review [23].

To ensure that the found literature could be used to answer RQ1 and RQ2, it had to focus on any one of three different areas. These areas were best practices for offshore software development, challenges associated with offshore software development or challenges associated with AI-assisted software development. Additionally, if a challenge or best practice was only focusing on a specific scenario or method not used generally within the software development processes, it was excluded. This was decided to keep a broad view of the topic to not unintentionally go into more depth on challenges and best practices not applicable to the general software development processes.

In regards to requirements of publishing year and number of citations, offshore software development is not something new and has been used for decades [24] which is why literature from as early as 2004 were used. For literature from 2018 or later only three citations was required, while 20 citations was the minimum for literature older than 2018. This was to include newer literature while still setting higher standards for the literature in general. If 20 citation had been used as a requirement for all literature about offshore software development, no literature from the last years would have been included. While three citations for all literature would have included unwanted literature. Setting a threshold of at least 20 citations ensured that only widely recognised studies were included, while still including newer literature by only requiring three citations. AI-assisted software development is a newer topic and therefore had other requirements. The literature included had to be published between January 2022 and February 2024, when the literature review was conducted. Any literature published by February 2023 had to have at least 5 citations while literature published later had no requirement on number of citations. This was based on the same reasoning as the offshore software development literature, to include the newest literature while still only including

recognised studies where applicable.

Challenges and best practices were extracted from the literature by taking notes and citations into a document divided into two categories, offshore software development and AI-assisted software development. These were only added to the document if it was determined to be a challenge or best practice by the literature or if it was included as citations from participants in the study. The found challenges were then added to two different matrices, one for offshore software development and one for AI-assisted software development, to further visualise the results.

3.1.2 Data analysis

A framework analysis [25] was performed to analyse the data collected from the literature review. The results were comparatively synthesised to compile the similarities and differences between the two practices and thereafter used when designing the interview questions and drafting an initial conclusion to RQ1 and RQ2.

If the only mention of a potential challenge was also mentioned as a non-challenge in the same text, that potential challenge was excluded from the final results. This exclusion was made because the potential challenges which fit this criteria could not explicitly be stated as definite challenges.

3.2 Interviews

A set of structured interviews [26] were conducted with ten domain experts. The domain experts all had experience with either AI-assisted software development or offshore software development. These interviews served to validate the findings of the literature review and allow a final answer to all three research questions to be drawn. Interviews were chosen over similar methods, such as questionnaires or workshops for a number of reasons. The main reason for why questionnaires were not chosen was because they lack flexibility. In a questionnaire no clarifications can be made if the participants are unable to understand certain concepts or have any other questions. There were two main reasons for why workshops were not chosen. The first one was the potential bias a workshop can create. When multiple participants are asked to answer the same question together it is difficult to ensure no one participant is influenced by the others. The second reason for why workshops were not chosen was because they were seen as unrealistic in regards to participant recruitment. A workshop would require a number of participants to all be available at the same time which was deemed unrealistic because most participants had busy schedules. Structured interviews allows for a flexible approach and would only require each participant to be available for a shorter amount of time which could be adapted to fit their schedule. The interviews were conducted through Microsoft Teams [27]. Remote interviews were chosen over in-person interviews to enable a larger amount of potential candidates to participate.

3.2.1 Interview questions

The interview questions were created based on the data gathered from the literature review. The focus of the questions were identifying which of the found challenges and best practices from offshore software development could be applicable to AI-software development. The interview questions can divided into two sections, *Control questions* and *Interview questions*.

In the first section, participants were asked about their professional experience with AI-assisted software development and to what extent their current company utilises AI assistance in its software development. Participants were also asked about how the potential use of AI assistants would affect their current role in their company. The goal of this section was to identify any potential biases towards or against the use of AI assistants and to establish the interviewees reliability. Biases were measured through an open-ended question where participants described how their roles would be affected if their companies implemented AI in the software development process. Reliability was measured through two questions about their own and their company's experience with AI-assisted software development.

In the second section, participants were given a list of challenges. Participants were then asked to identify which of the challenges would be applicable to AI-assisted software development. This list contained the challenges for both offshore software development and AI-assisted software development which were identified in the literature review. Two additional questions were also asked in regards to this list. The first of these was an open-ended question about if the participant could think of any potential challenges that were not listed. The second one was if any of the chosen challenges would have been mitigated if the AI-assistant was developed and maintained locally instead of using a third party tool. Participants were then given a second list which contained best practices. The best practices in this list were all the identified best practices for offshore software development found in the literature review. The participants were asked which of these could be used to solve the identified challenges from the first list. Finally, participants were then asked if they could think of any other best practices that would be applicable to AI-assisted software development which were not on the list.

A full list of all the questions used for the interviews can be found in the appendix.

3.2.2 Participants

The interview participants were recruited through contacts at five different companies, LinkedIn [28] and a forum women in technology. To decrease the number of interviews necessary to receive reliable results, two criteria were set for the participants beforehand. The participants had to either work with the software development process at a company where parts of the software development was placed offshore or they had to use AI in some way in the software development process. Since we did not have anything to offer in exchange for the interviews, the

participants had to be willing to help us for free.

3.2.3 Data analysis

The interview questions can be divided into four different types based on the data they provided:

1. Two closed questions asked participants to select the option that applied to them on a scale of one to five.
2. One open-ended question asked participants how AI integration would affect their current role.
3. Three closed questions asked participants to select any number of options from a list provided to them.
4. Two open-ended questions asked participants if they could come up with any additional options which were not included in a previously provided list.

All four types of interview questions where analysed in different ways. This was done because each type provided data suitable for different data analysis methods.

The first question type provided quantitative data which was analysed using summary statistics [29]. This method was chosen because it allows for conclusions to be drawn about the general participants experience working with AI assistants. Specifically using AI assistants to generate code in a professional setting.

The second question type generated qualitative data which was analysed using thematic analysis [30]. The only question which falls under this question type asked participants how their current role would be affected if AI assistants were to be implemented. Thematic analysis was therefore chosen to find the common themes in the responses which could be used to determine any potential bias towards or against AI-assisted software development.

The third question type generated quantitative data which was analysed using frequency analysis [31]. This option was used to showcase the applicability of each challenge and best practice through tendencies among participants. This also allowed for comparison between what challenges and best practices were chosen in regards to the participants experience using AI assistants.

The forth and final question type generated qualitative data which was analysed using thematic analysis [30]. This method was chosen to provide valuable insight about any additional challenges or best practices that were not found during the literature review.

4
Results

The results are presented in the order of the Research questions. Starting with the literature review, which answers RQ1 and RQ2. The section is divided into three categories for identified challenges and one for identified best practices. Lastly, the results from the interviews are presented to answer RQ2 and RQ3 and are divided into control questions and interview questions.

4.1 Literature review

The literature review resulted in 19 papers, seven regarding offshore software development and 12 regarding AI assisted software development. The results gained from the seven papers about offshore software development challenges and best practices started to overlap. Challenges and best practices within offshore software development is not a new topic since it started to gain traction in the 1980s [32] and several challenges could be found in a fewer amount of papers. The search for literature was concluded when new found literature only presented challenges and best practices already found previously in other literature. This resulted in seven papers. To identify the challenges associated with AI-assisted software development, it was necessary to conduct a broader search of literature, as it was difficult to find papers that covered specifically the challenges with AI-assisted software development. Instead we had to take a different approach and search more broadly and find challenges from papers. The AI-assisted software development literature often addressed challenges as secondary findings within more specialised studies. As a result, we had to examine a larger number of papers to ensure a comprehensive understanding of the various challenges in this field. Finally, when new found literature only presented challenges already found previously in other literature, the search was concluded. This resulted in 12 papers containing challenges within AI-assisted software development.

Below are the identified challenges divided into three categories: security, communication, and quality and performance. These categories were specifically chosen based on the literature in which the challenges were identified. Out of 19 reviewed papers, 13 of them addressed challenges that fell exclusively within one of these three categories. Of the remaining six papers, three of them covered challenges which fell within two of these categories and three of them covered challenges which fell within all three categories. Consequently, it was both straightforward

and practical to adopt these categories, as the challenges were already portrayed accordingly in the literature. Each category is then divided into smaller categories or statements chosen to represent the found challenges and display the amount of papers finding said challenges.

To analyse the data from the literature review, framework analysis was used. This method was chosen for its structured approach, which allowed for a comparison of challenges in offshore software development and AI-assisted software development within the above defined categories. This method enabled a comprehensive examination of the data, together with clear visualisation and interpretation of the key themes and insights obtained from the literature.

After the analysis, the identified best practices for offshore software development are presented. The best practices were identified parallel with the literature review of challenges of offshore software development. Out of the seven papers in the literature review, six papers provided both challenges and corresponding best practices or mitigation strategies. One paper solely presented challenges without best practices or mitigation strategies.

4.1.1 Security challenges

The literature review found challenges surrounding security in both offshore software development and AI-assisted software development. [33] states that companies should "always be mindful of security" when outsourcing because of the risk of intellectual property (IP) theft. [34] mentions stolen data and code when using and personalising AI-assisted programming tools. [33], [35], [34], [36] all points out the challenge of ensuring the ownership of the code. It is therefore important to make sure the code is neither reused by another party [33], [35], [34], [36] nor owned by another party and used without permission [33] and [35]. After identifying these challenges they were divided into smaller categories to more precisely present how they were declared in the literature. The identified security challenges are presented in Table 4.1 below.

Table 4.1: Identified security challenges divided into smaller categories

Offshore software development	
Security Challenges	References
IP theft	[33], [35]
Use stolen code	[33], [35]
Ownership of code	[33], [35]
Company secrets shared unknowingly	[33]
AI-assisted software development	
Security Challenges	References
IP theft	[34]
Hard to ensure code is not reused by another party	[34], [36]

4.1.2 Communication challenges

The literature review identified two separate offshore software development challenges related to communication. Communication in this thesis refers to how the teams communicate and work together. According to 85% of the literature, there are significant communication challenges in offshore software development. No challenges relating to communication between developers could be found for AI-assisted software development. The only mention of communication challenges with AI was [37] which declared a research gap that "no studies has attempted to cover how such tools affect productivity and collaboration as a team". Therefore offshore software development and AI-assisted software development has no communication challenges listed in common. The identified communication challenges can be found in Table 4.2.

Table 4.2: Identified communication challenges divided into two categories

Offshore software development	
Communication Challenges	References
Cultural differences	[33], [38], [39], [40]
Loss of informal communication	[35], [38], [39], [40], [41], [42]
AI-assisted software development	
Communication Challenges	References
No studies on communication and collaboration	[37]

4.1.3 Quality and performance challenges

For offshore software development, the literature review identified two separate challenges related to quality and performance. For AI-assisted software development, a total of five different challenges were identified with relation to quality and

performance. Seven out of twelve total literature references regarding AI-assisted software development mentioned that it was hard to ensure code is of good quality. Practitioners in the study conducted by [36] declared that "GitHub Copilot suggest solutions that don't work". [43] presents that 28.1% of participants in their study has "Difficulty of integration" as not all IDEs are supported as of now.

Another challenge found was *Can be considered distracting for some developers*. [44] stated that a few developers disabled the AI tool as "they found it noisy and intrusive". The identified quality and performance challenges can be found in Table 4.3.

Table 4.3: Identified quality and performance challenges divided into smaller categories

Offshore software development	
Quality and Performance Challenges	References
Loss of productivity due to time zone differences	[35], [38], [41], [42]
Tool incompatibility	[33], [42]
AI-assisted software development	
Quality and Performance Challenges	References
Hard to ensure code is of good quality	[36], [45], [46], [47], [48], [49], [50]
Can be considered distracting for some developers	[44]
Problems with using certain IDEs	[36], [43], [51]
Problematic when working with larger or multiple files	[36], [47]
Can not always incorporate highly specialised libraries and APIs	[44]

4.1.4 Data analysis

To summarise the findings and visualise the result, the found similarities between the challenges of offshore software development and AI-assisted software development have been grouped into three challenges. Three challenges were chosen to cover all found similarities while still maintaining the integrity of each challenge to accurately answer RQ1 and RQ2.

IP theft: As seen in Table 4.1 both offshore software development and AI-assisted software development has *IP theft* as a challenge and *Company secrets shared unknowingly* is an attribute of *IP theft* [52]. Both challenges are therefore presented together as *IP theft* in Table 4.4.

Code Privacy: *Use stolen code*, *Ownership of code* and *Hard to ensure code is not reused by another party* are three challenges listed in Table 4.1. The challenges are about who owns the code and who has the right to use it. [33] declares *Ownership of code* as an attribute of *IP theft*. However, [36] does not mention IP theft but only how "developers pay much attention to code privacy threat". Therefore these challenges are listed as one collective challenge, *Code privacy*, and remains separate from *IP theft* in Table 4.4.

Tool incompatibility: *Problems with using certain IDEs*, *Problematic when working with larger or multiple files* and *Can not always incorporate highly specialised libraries and APIs* is connected to the AI-assistant being compatible with tools used when coding. Resulting in issues for the users as it "is the most expected features of users" [43]. *Tool incompatibility* in regards to offshore software development refers to developers at different sites using different tools or different version of the same tool and causing issues [42]. Hence, *Tool incompatibility* in Table 4.4 refers to these mentioned challenges.

Table 4.4: Identified similarities between offshore software development challenges and AI-assisted software development challenges

Shared challenges	**Offshore software development**	**AI-assisted software development**
IP theft	[33], [35]	[34]
Code privacy	[33], [35]	[34], [36]
Tool incompatibility	[33], [42]	[36], [43], [44], [47], [51]

This results in three shared challenges between offshore software development and AI-assisted software development. The challenges are presented and described differently in the literature but still fall into the three categories listed in Table 4.4. Ten out of the 14 challenges identified were determined similar and then synthesised into *IP theft*, *Code Privacy* and *Tool incompatibility*. While no similarities were found for the remaining four challenges, determining them not to be similar.

4.1.5 Best practices

The literature review identified eleven best practices of offshore software development. Each of the identified best practices are connected to at least one challenge listed in Table 4.1 through 4.3 These best practices were used when conducting the interviews to answer RQ3. The identified best practices are listed below in Table 4.5.

Table 4.5: Identified best practices for offshore software development

Offshore software development	
Best practices	References
Get written agreements (who owns the code)	[33]
Ensure original code (no stolen code)	[33], [35]
Confidentiality agreements	[33]
Do not use it for highly sensitive code	[33]
Follow the sun method (Someone is always working on the code)	[38]
Keep core technologies and competencies close to home	[33], [41]
Log and/or watermark your products	[33]
Make use of global pairs	[38]
Using Agile Methods	[35], [38]
Optimise travelling and creating friendships	[41]
Use good communication tools	[39], [41], [42]

4.2 Interviews

Structured interviews were conducted with ten participants from different companies operating in the field of software development. The participants covered a wide variety of roles from developers to managers. All participants operated in Europe. By using the challenges and best practices gathered from the literature review we could verify the result and gather more information from professionals within the field.

4.2.1 Control questions

When answering two of the control questions the participants were given a table which defined a scale of one to five. This was done to ensure that all participants interpreted the numbers on the scale in the same way when asked about their experience using AI. The definition of each number is shown in Figure 4.1.

1	Never used an AI based tool
2	Use an AI based tool as an alternative search engine
3	Use an AI based tool to sometimes generate code
4	Use an AI assistant for coding on a monthly basis
5	Use an AI assistant for coding on a weekly basis

Figure 4.1: A scale from 1 to 5 to show participants what each level indicates when answering control questions

The two previously mentioned control questions asked the participants about their personal and their companies' use of AI-assisted software development. The result of these two questions are shown in Figure 4.2 and Figure 4.3. Figure 4.2 shows that 50% of the companies where participants work use AI-assistants for coding on a weekly basis while 30% of the companies only use AI as an alternative search engine. The remaining participants are in between and have used an AI based tool to sometimes generate code. Similar AI usage can be seen in the participants' personal work. In Figure 4.3 it is shown that 40% are using AI-assistants for coding on a weekly basis and 30% use AI as an alternative search engine, while the remaining participants are using it to generate code but on a monthly basis or less often.

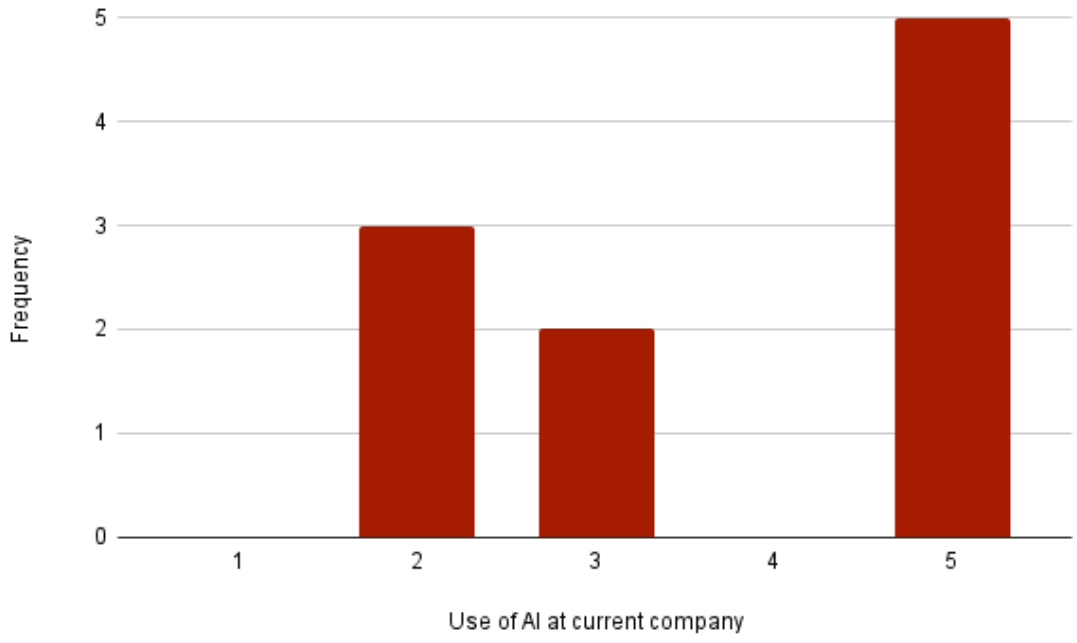

Figure 4.2: On a scale from 1-5, to what extent does the participant's company use AI in the software development process

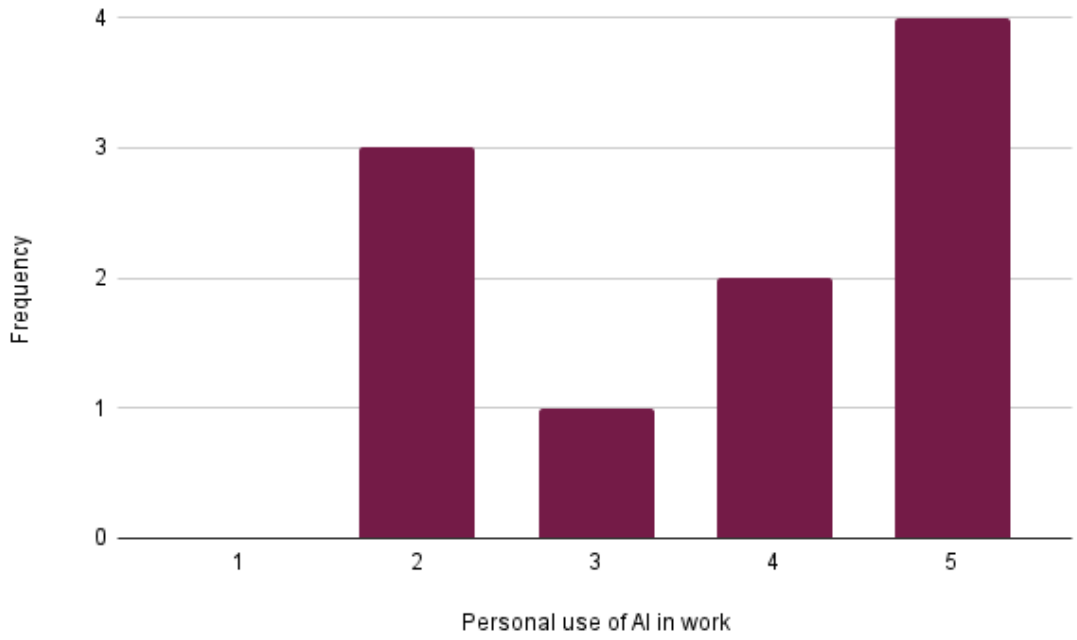

Figure 4.3: On a scale from 1-5 of how much the participant personally is using or have previously used AI in the software development process

The final control question asked the participant to predict how their current role would be affected by implementing AI in the software development process. For this question thematic analysis was used to determine and group the responses into four themes. Out of the 10 participants three of them answered that there would be no changes since AI was already implemented. Half of the participants stated that the AI would assist them with their regular tasks to make them more efficient and produce higher quality work and one participant stated that it would make their job outright easier. One participant answered in a neutral tone stating that it would change their job in some way but they were unsure how it would affect them. No participant predicted that their role would be negatively impacted or become redundant. The response rates for each theme can be found in Figure 4.4.

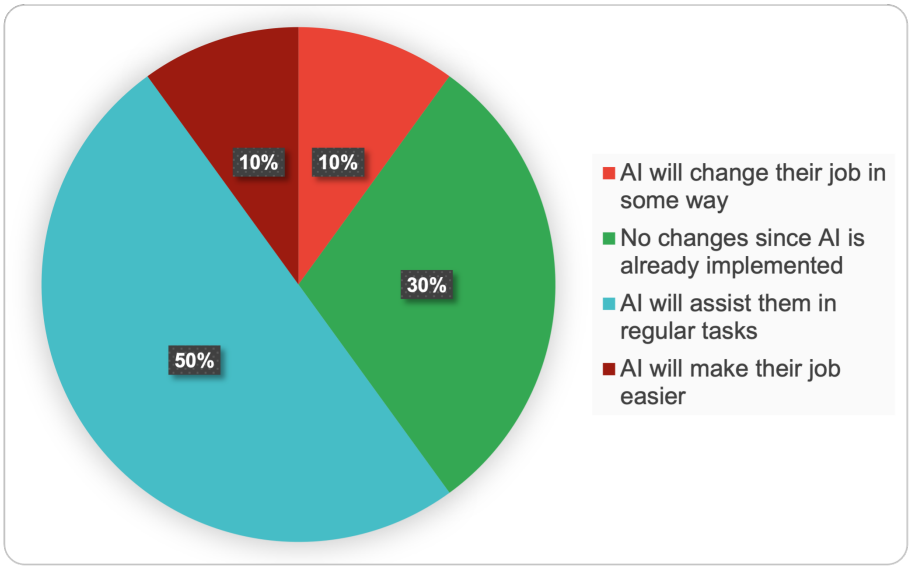

Figure 4.4: How the participants predicts their job will be affected when implementing AI in the software development process

4.2.2 Interview questions

The interview questions consisted of five questions. Questions one, three and four were multiple response questions while question two and five were open-ended questions. To answer these questions, participants were given two lists: one list of challenges which were identified in the literature review and one list of best practices identified in the literature review. Each challenge and best practice was given a label to make it easier to reference it. A list of the labels used to reference these can be found in Table 4.6 and Table 4.7 below.

Table 4.6: Challenges and their labels

Label	Challenge
C1	Ownership of code
C2	Tool incompatibility (uses different versions)
C3	Loss of informal communication
C4	Cultural differences
C5	Hard to ensure code is of good quality
C6	Use stolen code
C7	Loss of productivity due to time zone differences
C8	Hard to ensure code is not reused by another party
C9	Problems with using certain IDEs
C10	Problematic when working with larger or multiple files
C11	IP theft
C12	Company secrets shared unknowingly
C13	Cannot always incorporate highly specialised libraries and APIs
C14	Can be considered distracting for some developers

Table 4.7: Best practices and their labels

Label	Best practice
BP1	Get written agreements (who owns the code)
BP2	Ensure original code (no stolen code)
BP3	Confidentiality agreements
BP4	Do not use it for highly sensitive code
BP5	Follow the sun method (Someone is always working on the code)
BP6	Keep core technologies and competencies close to home
BP7	Log and/or watermark your products
BP8	Make use of global pairs
BP9	Using Agile Methods
BP10	Optimise travelling and creating friendships
BP11	Use good communication tools

Question one asked which of the listed challenges participants believed to be applicable to AI-assisted software development. The results for this question can be seen in Figure 4.5. They show that four of the identified 14 challenges were seen as applicable to AI-assisted software development by half or more of the 10 industry professionals interviewed. C5, C12 and C13 were picked as applicable to AI assisted software development in 80% or more of the interviews.

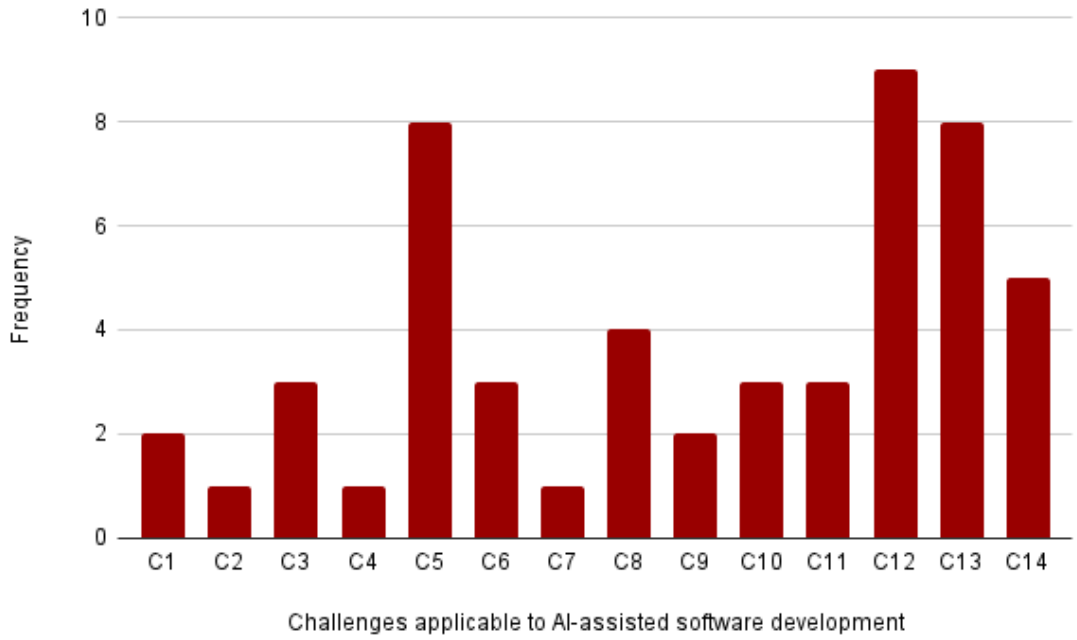

Figure 4.5: Challenges applicable to AI-assisted software development

Question two asked the participants if they could think of any challenges that were applicable to AI-assisted software development that were not included in the provided list. This question resulted in a large verity of answers. A Majority of the participants answered with challenges which were already on the list using different wording. Two of the participants did however answer with a challenge which were not already listed. That challenge was that continuous utilisation if AI assistants could cause the user to become either lazy or forget how to perform simple software development tasks.

Question three asked which of the listed challenges participants believed would be mitigated if the AI assistant were to be developed and maintained locally by their company instead of using a third party tool. The results to this question can be found in Figure 4.6. Out of 14 shown challenges, three were picked by 50% or more of the participants.

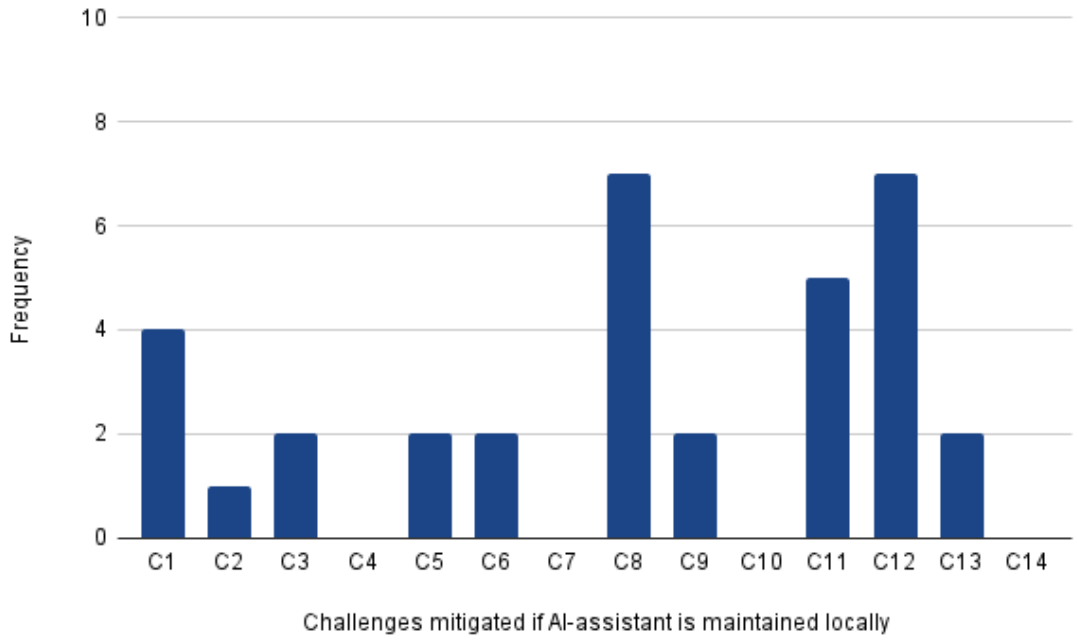

Figure 4.6: Challenges that would be mitigated if the AI was maintained locally

Question four asked which of the listed best practices participants believed to be applicable to AI-assisted software development. The results for this question can be seen in Figure 4.7. Four of the 11 options, BP3, BP4, BP6 and BP7 where picked by 50% or more of the participants.

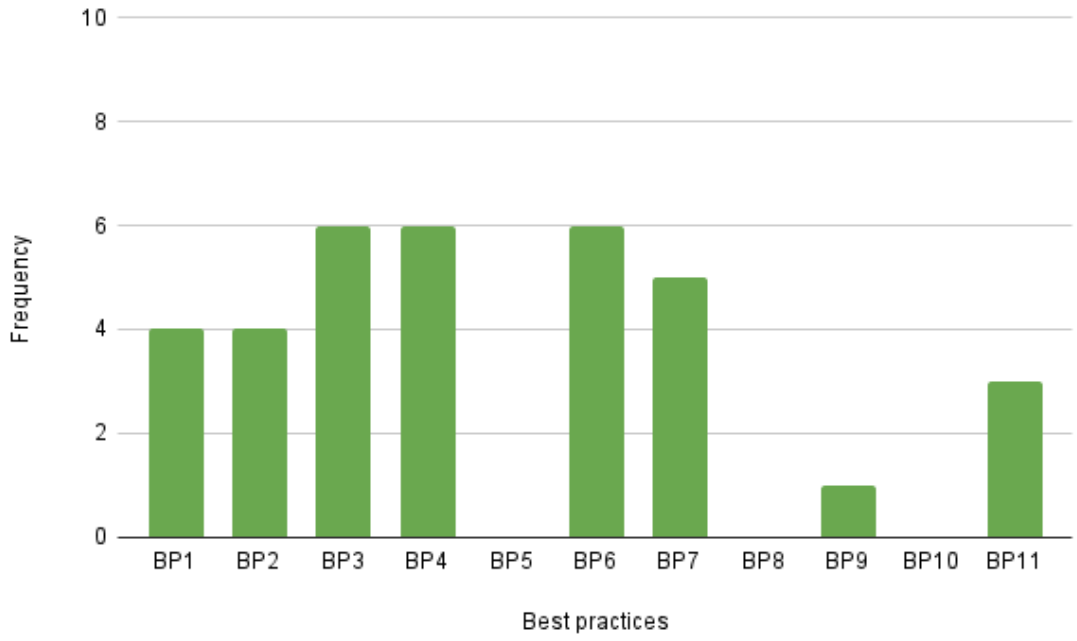

Figure 4.7: Best practices applicable to AI-assisted software development

Question five asked the participants if they could think of any best practices that were applicable to AI-assisted software development that were not included in the provided list. Similar to question two, the result of this question was varied in responses. Four of the total 10 participants could not think of any other best practices. Three participants suggested best practices that were already on the list using different words. The final three participants all suggested code reviews as a best practice to ensure that the code which was produced with AI assistance was of expected quality. Two of these three participants had differing views on how such a code review should be conducted. One participant suggested that a separate AI from the one that was used when producing the code could be utilised when reviewing said code. The other participant said that AI should not at all be used for the code review to "make results more human".

4.2.3 Summary and analysis

To summarise the findings and to analyse results further. Table 4.8 shows the challenges that 50% or more of the participants believed to be applicable to AI-assisted software development.

Table 4.8: Challenges participants believed to be most applicable

Label	Challenge	Frequency
C12	Company secrets shared unknowingly	9 (90%)
C5	Hard to ensure code is of good quality	8 (80%)
C13	Cannot always incorporate highly specialised libraries and APIs	8 (80%)
C14	Can be considered distracting for some developers	5 (50%)

Table 4.9 shows the challenges that 50% or more of the participants believed to be mitigated by using a local AI-assisted tool instead of one from a third party.

Table 4.9: Challenges participants believed to be mitigated when using local AI-assisted tools

Label	Challenge	Frequency
C8	Hard to ensure code is not reused by another party	7 (70%)
C12	Company secrets shared unknowingly	7 (70%)
C11	IP theft	5 (50%)

Table 4.10 shows the best practices that 50% or more of the participants believed to be applicable to AI-assisted software development.

Table 4.10: Best practices participants believed to be most applicable

Label	Best practice	Frequency
BP3	Confidentiality agreements	6 (60%)
BP4	Do not use it for highly sensitive code	6 (60%)
BP6	Keep core technologies and competencies close to home	6 (60%)
BP7	Log and/or watermark your products	5 (50%)

The above tables show a summary of the three to four most frequently picked options for interview question one, three and four amongst all participants. Dividing participants based on their answer on their personal use of AI in work, presented in Figure 4.3, resulted in new sets of graphs showing a pattern between the personal use of AI and answers on interview question one (Figure 4.8), interview question three (Figure 4.9) and interview question four (Figure 4.10). Four participants who answered either two or three on the previously mentioned control question were put into the group with a lower level of experience. Six participants who answered either four or five was put into the group with higher level of experience. Figure 4.8 shows how each group responded to interview questions one, three and four.

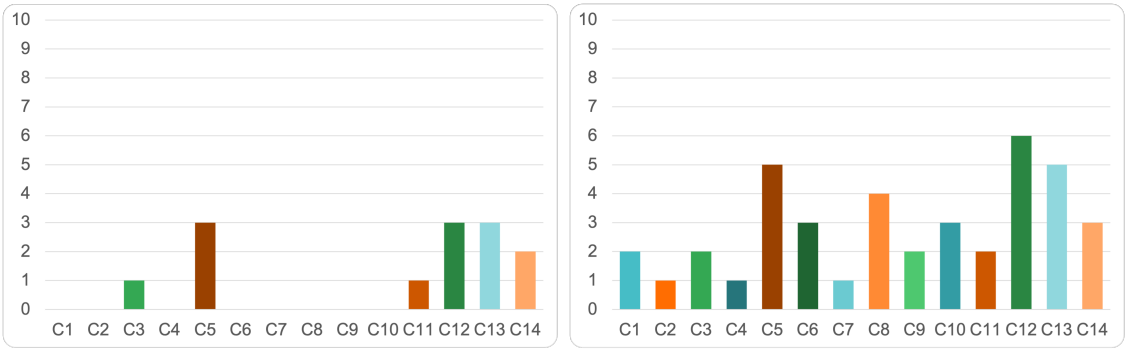

Figure 4.8: Number of times each challenge applicable to AI-assisted software development were picked by interview participants, grouped by level of experience with software development using AI-assisted tools. The left graph represents participants with a lower level of experience. The right graph represents participants with a higher level of experience

Table 4.11 presents the challenges 50% or more of the higher experienced participants believed to be applicable to AI-assisted software development. It displays slightly different challenges from Table 4.8 as well as a higher frequency with 100% of higher experienced participants stating that *Company secrets shared unknowingly* is a challenge for AI-assisted software development.

Table 4.11: Challenges high level experience participants believed to be most applicable

Label	Challenge	Frequency
C12	Company secrets shared unknowingly	6 (100%)
C5	Hard to ensure code is of good quality	5 (83%)
C13	Cannot always incorporate highly specialised libraries and APIs	5 (83%)
C8	Hard to ensure code is not reused by another party	4 (67%)

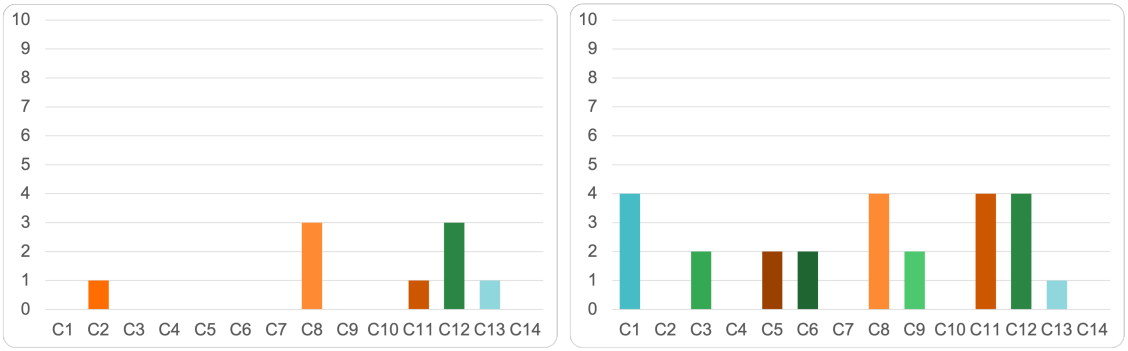

Figure 4.9: Number of times each challenge mitigated by using a local AI were picked by interview participants, grouped by level of experience with software development using AI-assisted tools. The left graph represents participants with a lower level of experience. The right graph represents participants with a higher level of experience

Table 4.12 presents the challenges 50% or more of the higher experienced participants believed to be mitigated for AI-assisted software development if the AI was maintained locally.

Table 4.12: Challenges high level experience participants believed to be mitigated if the AI was maintained locally

Label	Challenge	Frequency
C1	Ownership of code	4 (67%)
C8	Hard to ensure code is not reused by another party	4 (67%)
C11	IP theft	4 (67%)
C12	Company secrets shared unknowingly	4 (67%)

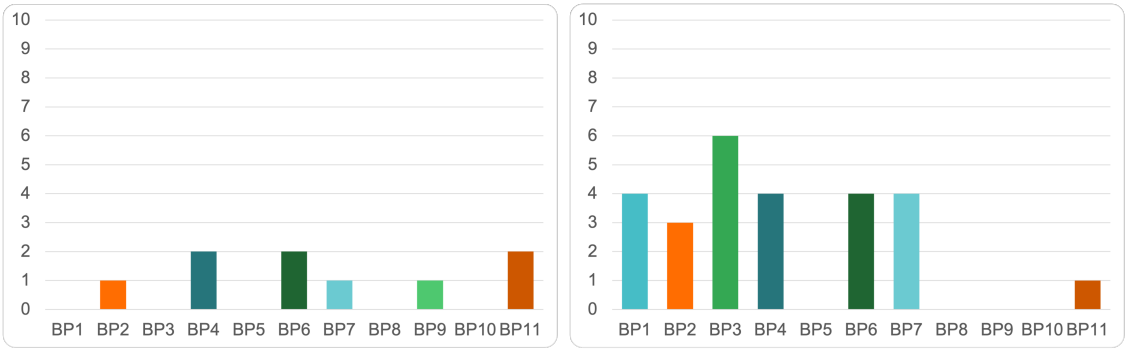

Figure 4.10: Number of times each best practice applicable to AI-assisted software development were picked by interview participants, grouped by level of experience with software development using AI-assisted tools. The left graph represents participants with a lower level of experience. The right graph represents participants with a higher level of experience

Table 4.13 presents the best practices 50% or more of the higher experienced participants believed to be applicable to AI-assisted software development. BP3 is applicable to AI-assisted software development according to all higher level experienced participants. BP1 is applicable according to 67% of this group while no lower level experienced participants claimed BP1 to be applicable.

Table 4.13: Best practices high level experienced participants believed to be most applicable

Label	Best practice	Frequency
BP3	Confidentiality agreements	6 (100%)
BP1	Get written agreements (who owns the code)	4 (67%)
BP4	Do not use it for highly sensitive code	4 (67%)
BP6	Keep core technologies and competencies close to home	4 (67%)
BP7	Log and/or watermark your products	4 (67%)

The result of this division shows that participants with a higher level of experience, on average, had a tendency to choose a higher number of challenges and best practices which they believed to be applicable to AI-assisted software development or mitigated by local tools. The participants with a lower level experience, on average, chose fewer options for all three questions. Some notable differences between the two groups include C8 being picked exclusively by the experienced group in question one and BP3 being picked by every participant in the higher level experience group in question three as opposed to none of the participants in the lower experience group.

5
Discussion

This section discusses the findings in relation to the research questions, present threats to validity and future work.

5.1 Answer to research questions

The aim of this thesis was to answer all three research questions stated in the introduction. In this section an answer to all three of the research questions is presented and discussed.

5.1.1 Research Questions 1 and Research Question 2

Research question one "*What are the similarities between AI-assisted software development practices and offshore software development practices?*" and research question two "*What are the shared challenges between offshore software development and AI-assisted software development?*". After conducting a literature review for literature on both offshore software development and AI-assisted software development, we have presented answers to both of the research questions in Section 4.1. The answer presented is that offshore software development is similar to AI-assisted software development in three main areas, security, privacy and tool incompatibility. These three areas also contain the shared challenges between offshore software development and AI-assisted software development.

To further validate RQ2, the results from the interviews was used. *Company secrets shared unknowingly* (C12) and *Hard to ensure code is not reused by another party* (C8) were presented as *IP theft* and *Code Privacy* in Table 4.4 respectively. They were determined by 83% (C12) and 67% (C8) highly experienced participants to be significant challenges within AI-assisted software development. *Cannot always incorporate highly specialised libraries and APIs* (C13) were presented as *Tool incompatibility* in Table 4.4 and was determined to be a challenge by 83% of high level experience participants. *Hard to ensure code is of good quality* (C5) were not listed as an offshore software development challenge in the literature review and were not determined to be similar to any of the listed offshore software development challenges. Therefore, C5 is not a shared challenge.

5.1.2 Research Question 3

Research question three "*What offshore software development best practices are applicable to AI-assisted software development?*". In order to answer this research question, we first compiled a list of best practices in offshore software development. Then interviews with industry professionals were held to see if any of the compiled practices would be applicable to AI-assisted software development. According to the interviews, four of the compiled best practices are applicable to AI-assisted software development:

- BP3: Confidentiality agreements
- BP1: Get written agreements (who owns the code)
- BP4: Do not use it for highly sensitive code
- BP6: Keep core technologies and competencies close to home
- BP7: Log and/or watermark your products

All of the above listed best practices had 50% or more of the interview participants say that they were applicable. With 100% of the high level experience participants stating that BP3 is applicable to AI-assisted software development, BP3 can be determined as a best practice for AI-assisted software development. The rest of the best practices listed above were applicable to AI-assistant software development according to 67% of the high level experience participants. Those best practices can therefore require more research before determining them to be applicable to AI-assistant software development.

Participants stated four challenges could be mitigated if the AI was maintained locally, in Table 4.12. However, this also requires further research to determine if a locally maintained AI-assistant can help mitigate challenges and therefore be considered a best practice. As this was not something we focused on, it was not further researched in this thesis.

The results of RQ3 have an impact on state of the art, as to the best of our knowledge, there have yet to be any papers proposing best practices for working with AI-assisted software development. This is also the first study to compare the challenges in AI-assisted software development to those in offshore software development, as no other literature on the topic could be found.

5.2 Threats to validity

A number of threats to validity have been identified throughout work on this thesis.

5.2.1 Internal validity

One identified threat is the potential biases we as authors could have had on the result and conclusions drawn. To minimise these, we have made sure to always try to remain as objective as possible when conducting interviews and analysing

results.

Another potential threat is the lack of diversity in the interview participants chosen. All of the interview participants are practitioners within the the area of software development with experience in either AI-assisted software development or offshore software development. No participants working in academia were able to be interviewed. We tried minimising this threat by reaching out for participants from academia. In the end no professionals within academia were able to be recruited as all that responded to our invitation were unable to participate until after the end of our thesis work.

Another identified threat was a potential bias among interview participants towards or against the use of AI-assisted tools in software development. To minimise this threat control questions were included in the interviews to clarify how the potential implementation of such tools in the future would affect each participant.

5.2.2 External validity

A potential external threat is that the low number of interview participants could cause the results to not be able to be generalised across the software development industry as a whole. To minimise this, participants from a wide array of roles were chosen in order to make the results as general as possible with the low number of participants. All participants were from Europe, which is something that was not able to be minimised. Attempts to include both more participants and participants from other continents were made but proved difficult due to participant availability and time zone differences.

5.2.3 Construct validity

One identified threat is the risk that any potential challenge or best practice from both offshore software development and AI-assisted software development could have been missed in the literature review and therefore not covered in the study. To minimise this threat we made sure to be as thorough as possible when conducting all parts of the literature review.

Another potential threat to construct validity is the participants understanding of the concepts explored in this thesis. There is no guarantee that each participant have the same interpretation of the concepts "challenges", "best practices" and "AI-assisted software development" which could cause inconsistent data if not handled properly. To minimise this we gave each participant an explanation of the concepts to ensure a similar baseline of knowledge and interpretation among each participant.

5.2.4 Conclusion Validity

An identified threat to conclusion validity was that interview questions could have been interpreted differently by participants. To minimise this we performed a pilot interview to validate the interview format and that the questions were easily understandable.

5.3 Future work

The results of this thesis have proposed some best practices that are potentially applicable to AI-assisted software development gathered through looking at a similar software development technique. BP1, BP4, BP6 and BP7 were applicable according to a majority of the high level experience participants and can benefit from further research to determine their applicability to AI-assisted software development. One way to determine this can be to put the proposed best practices from this thesis to the test in a real world setting. Another way is to conduct a similar study over a longer period of time with more participants to see if the same best practices are repeated as applicable by the majority of the participants. When it comes to contribution, we believe that this thesis have showcased the potential of how to find best practices applicable to AI-assisted software development. There could exist other areas of software development which could be investigated in similar ways to find further applicable best practices.

6
Conclusion

This thesis aimed to answer the three research questions that were introduced in the introduction. Through the use of a literature review, literature on both offshore software development and AI-assisted software development was found and analysed. This analysis resulted in similarities found in 10 out of the 14 challenges. Three challenges were then determined to be shared between offshore software development through both the literature review and interviews. These challenges were listed as IP theft, code privacy and tool incompatibility. The literature review also resulted in a list of identified best practices for offshore software development which were used to design the interviews.

The interviews were structurally conducted with industry professionals, totaling 10 participants. The structured interviews contained a mixture of both close-ended questions and open-ended questions. These were designed to establish the reliability of the participants, identify any potential bias among participants and gather results about which of the identified best practices are applicable to AI-assisted software development. The results of the interviews was used further validate the answer of research question two, as well as answering the third and final research question about which practices from offshore software development are applicable to AI-assisted software development.

The findings demonstrate notable similarities between the challenges of offshore software development and AI-assisted software development. It also shows that at least one best practice for offshore software development is applicable to AI-assisted software development. With further research, more best practices may be applicable, as determined in this thesis.

Bibliography

[1] M. Neelam, "Aspects of Artificial Intelligence," in J. Karthikeyan, S.-H. Ting, and Y.-J. Ng (eds), "Learning Outcomes of Classroom Research," L'Ordine Nuovo Publication, India, 2022, pp. 250-256, ISBN: 978-93-92995-15-6.

[2] M. Harman, "The role of Artificial Intelligence in Software Engineering," 2012 First International Workshop on Realizing AI Synergies in Software Engineering (RAISE), Zurich, Switzerland, 2012, pp. 1-6.

[3] GitHub Copilot · Your AI pair programmer, https://github.com/features/copilot.

[4] Amazon CodeWhisperer · Your AI-powered productivity tool for the IDE and command line, https://aws.amazon.com/codewhisperer/

[5] D. Sobania, M. Briesch, F. Rothlauf, "Choose your programming copilot: a comparison of the program synthesis performance of github copilot and genetic programming," Computation Conference (GECCO '22), Boston, USA, 2022, pp. 1019-1027.

[6] B. Zhang, P. Liang, X. Zhou, A. Ahmad , M. Waseem, "Practices and Challenges of Using GitHub Copilot: An Empirical Study" in 35th International Conference on Software Engineering and Knowledge Engineering (SEKE), 2023.

[7] V. Garousi, K. Petersen, B. Ozkan, "Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review," Information and Software Technology 79, 2016, pp. 106-127.

[8] A. Fan, B. Gokkaya, M. Harman, M. Lyubarskiy, S. Sengupta, S. Yoo, J. M. Zhang, "Large language models for software engineering: Survey and open problems," arXiv preprint arXiv:2310.03533, 2023.

[9] J. Cusick, A. Prasad, "A Practical Management and Engineering Approach to Offshore Collaboration," IEEE Software 23, (5), 2006, pp. 20-29.

[10] S. Sahay, B. Nicholson, S. Krishna, "Global IT Outsourcing," Cambridge University Press, 2003.

[11] BairesDev, "Onshore, Nearshore, and Offshore Software Development: What's the Difference?," https://www.bairesdev.com/blog/onshore-nearshore-offshore-software-development/. Accessed: 1 June 2024.

[12] A. Delmonte, R. McCarthy, "Offshore Software Development: Is the Benefit Worth the Risk?", 2003.

[13] J. H. Fetzer, "What is Artificial Intelligence?," in *Artificial Intelligence: Its Scope and Limits*, Dordrecht: Springer Netherlands, 1990, pp. 3-27.

[14] M. Wermelinger, "Using GitHub Copilot to Solve Simple Programming Problems," presented at the 54th ACM Technical Symposium on Computer Science Education., 2023, pp. 172-178.

[15] C. Ebert and P. Louridas, “Generative AI for Software Practitioners,” in IEEE Software 40, (4) 2023, pp. 30-38.
[16] S. Khan et al., “Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review,” Information and Software Technology 53, 2011, pp. 693-706.
[17] D. Damian, L. Izquierdo, J. Singer, I. Kwan, “Awareness in the wild: why communication breakdowns occur,” in International Conference on Global Software Engineering, 2007, pp. 81–90.
[18] H. Holmstrom, E.Ó. Conchúir, P. Ågerfalk, B. Fitzgerald, “Global software development challenges: a case study on temporal, geographical and sociocultural distance,” in International Conference on Global Software Engineering, 2006, pp. 3–11.
[19] D. A. Vogel, J.E. Connelly, “Best practices for dealing with offshore software development,” Handbook of Business Strategy 6, (1), 2005, pp. 281-286.
[20] R.H. Kulkarni, P. Padmanabham, “Integration of artificial intelligence activities in software development processes and measuring effectiveness of integration,” IET Software 11, 2017, pp. 18-26.
[21] A. M. Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, Z. M. Jiang, "GitHub Copilot AI pair programmer: Asset or Liability?," Journal of Systems and Software 203, 2023.
[22] S. Kvale, "Den kvalitativa forskningsintervjun," Studentlitteratur, Lund, 1997.
[23] C. Wohlin, M. Kalinowski, K. Romero Felizardo, E. Mendes, "Successful combination of database search and snowballing for identification of primary studies in systematic literature studies," Information and Software Technology 147, 2022.
[24] X. Zhou, Y. Jin, H. Zhang, S. Li and X. Huang, "A Map of Threats to Validity of Systematic Literature Reviews in Software Engineering," 2016 23rd Asia-Pacific Software Engineering Conference (APSEC), Hamilton, New Zealand, 2016, pp. 153-160.
[25] L. J. Goldsmith, "Using Framework Analysis in Applied Qualitative Research," Qualitative Report, vol. 26, no. 6, pp. 2061-2076, 2021. DOI: 10.46743/2160-3715/2021.5011.
[26] R. J. G. Opdenakker, "Advantages and disadvantages of four interview techniques in qualitative research," Forum Qualitative Sozialforschung / Forum: Qualitative Social Research, vol. 7, no. 4, 2006. Institut fur Klinische Psychologie und Gemeindesychologie.
[27] Microsoft Teams, https://www.microsoft.com/sv-se/microsoft-teams/group-chat-software
[28] LinkedIn, https://www.linkedin.com/home
[29] J. McDermott, M. Schemitsch, and E. Simoncelli, "Summary statistics in auditory perception," Nat. Neurosci., vol. 16, pp. 493-498, 2013. DOI: 10.1038/nn.3347.
[30] V. Braun and V. Clarke, "Thematic analysis," in H. Cooper, P. M. Camic, D. L. Long, A. T. Panter, D. Rindskopf, and K. J. Sher (Eds.), APA Handbook of Research Methods in Psychology, Vol. 2. Research Designs: Quantitative, Qualitative, Neuropsychological, and Biological, American Psychological Asso-

ciation, 2012, pp. 57-71. DOI: 10.1037/13620-004.
[31] J. Lutabingwa and C. J. Auriacombe, "Data analysis in quantitative research," Journal of Public Administration, vol. 42, no. 6, pp. 528-548, 2007.
[32] C. Farrell, "Software development: an international perspective," Journal of Systems and Software 9, 1989, pp. 305-309.
[33] D. A. Vogel, J. E. Connelly, "Best practices for dealing with offshore software development," Handbook of Business Strategy 6, 2005, pp. 281-286.
[34] M. F. Wong, S. Guo, C. N. Hang, S. Ho, and C. W. Tan, "Natural Language Generation and Understanding of Big Code for AI-Assisted Programming: A Review," Entropy, vol. 25, 2023.
[35] M. Jiménez, M. Piattini, and A. Vizcaíno, "Challenges and Improvements in Distributed Software Development: A Systematic Review," Advances in Software Engineering, vol. 2009, Article ID 710971, pp. 1-14, 2009. [Online]. Available: https://doi.org/10.1155/2009/710971
[36] B. Zhang, P. Liang, X. Zhou, A. Ahmad, and M. Waseem, "Practices and challenges of using GitHub Copilot: An empirical study," arXiv preprint arXiv:2303.08733, 2023.
[37] S. Sikand, K. K. Phokela, V. S. Sharma, K. Singi, V. Kaulgud, T. Tung, et al., "How much SPACE do metrics have in GenAI assisted software development?" in Proceedings of the 17th Innovations in Software Engineering Conference (ISEC '24), Association for Computing Machinery, New York, NY, USA, 2024, pp. 1–5. [Online]. Available: https://doi.org/10.1145/3641399.3641419
[38] S. ul Haq and M. Tariq, "The Context of Global Software Development: Challenges, Best Practices and Benefits", vol. 3, no. 4, pp. 193-197, Oct. 2011. [Online]. Available: https://doi.org/10.22610/imbr.v3i4.933
[39] S. Cherry and P. N. Robillard, "Communication problems in global software development: Spotlight on a new field of investigation," in Proceedings of the International Workshop on Global Software Development, International Conference on Software Engineering, Edinburgh, Scotland, May 2004, pp. 48-52.
[40] M. D. Kahya and C. Şeneler, "A Literature Review on Challenges in Distributed Software Development," vol. 9, pp. 159-170, 2018.
[41] D. Boland and B. Fitzgerald, "Transitioning from a co-located to a globally-distributed software development team: A case study at Analog Devices Inc.," in 3rd Workshop on Global Software Development, IET, May 2004.
[42] S. Komi-Sirviö and M. Tihinen, "Lessons learned by participants of distributed software development," Knowledge and Process Management, vol. 12, no. 2, pp. 108-122, 2005. [Online]. Available: https://doi.org/10.1002/kpm.225
[43] B. Zhang, P. Liang, X. Zhou, A. Ahmad, and M. Waseem, "Demystifying Practices, Challenges and Expected Features of Using GitHub Copilot," arXiv preprint arXiv:2309.05687, 2023.
[44] V. Murali, C. Maddila, I. Ahmad, M. Bolin, D. Cheng, N. Ghorbani, et al., "CodeCompose: A large-scale industrial deployment of AI-assisted code authoring," arXiv preprint arXiv:2305.12050, 2023.
[45] B. Idrisov and T. Schlippe, "Program Code Generation with Generative AIs," Algorithms, vol. 17, no. 2, p. 62, 2024. [Online]. Available: https://doi.org/10.3390/a17020062

[46] Y. Fu, P. Liang, A. Tahir, Z. Li, M. Shahin, J. Yu and J. Chen, "Security Weaknesses of Copilot Generated Code in GitHub," arXiv preprint arXiv:2310.02059, 2023.

[47] R. Pudari and N. A. Ernst, "From copilot to pilot: Towards AI supported software development," arXiv preprint arXiv:2303.04142, 2023.

[48] R. A. Poldrack, T. Lu, and G. Beguš, "AI-assisted coding: Experiments with GPT-4," ArXiv, abs/2304.13187, 2023.

[49] H. Mozannar, G. Bansal, A. Fourney, and E. Horvitz, "Reading Between the Lines: Modeling User Behavior and Costs in AI-Assisted Programming," ArXiv, abs/2210.14306, 2022.

[50] S. Imai, "Is GitHub Copilot a substitute for human pair-programming? An empirical study," in Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings (ICSE '22), Association for Computing Machinery, New York, NY, USA, 2022, pp. 319–321. [Online]. Available: https://doi.org/10.1145/3510454.3522684

[51] C. W. Tan, S. Guo, M. F. Wong, and C. N. Hang, "Copilot for Xcode: Exploring AI-Assisted Programming by Prompting Cloud-based Large Language Models," arXiv preprint arXiv:2307.14349, 2023.

[52] A. Story, "Intellectual Property and computer software," in Intellectual Property Rights and Sustainable Development, ed. by ICTSD-UNCTAD, Imprimerie Typhon, Chavanod, 2004.

A

Interview questions

Control Questions

1. What is your role in the company?

2. Is your current work using AI in the software development process?

3. To what extent?

1	Never used an AI based tool
2	Use an AI based tool as an alternative search engine
3	Use an AI based tool to sometimes generate code
4	Use an AI assistant for coding on a monthly basis
5	Use an AI assistant for coding on a weekly basis

4. To what extent are you using or have you used AI assistance in your work?

1	Never used an AI based tool
2	Use an AI based tool as an alternative search engine
3	Use an AI based tool to sometimes generate code
4	Use an AI assistant for coding on a monthly basis
5	Use an AI assistant for coding on a weekly basis

5. Please describe how your role would be affected if your company decided to adopt/integrate AI assistants in software development processes/activities?

Interview Questions

1. Which of the following challenges listed do you think applies to AI-assisted software development?
   (a) Ownership of code
   (b) Tool incompatibility (uses different versions)
   (c) Loss of informal communication
   (d) Cultural differences
   (e) Hard to ensure code is of good quality
   (f) Use stolen code

(g) Loss of productivity due to time zone differences
(h) Hard to ensure code is not reused by another party
(i) Problems with using certain IDEs
(j) Problematic when working with larger or multiple files
(k) IP theft
(l) Company secrets shared unknowingly
(m) Cannot always incorporate highly specialised libraries and APIs
(n) Can be considered distracting for some developers

2. Do you think there are any other challenges not shown here?

3. Which of the above mentioned challenges applicable to AI assisted software development would be mitigated if the AI assistant was developed and maintained locally by your company?
   (a) Ownership of code
   (b) Tool incompatibility (uses different versions)
   (c) Loss of informal communication
   (d) Cultural differences
   (e) Hard to ensure code is of good quality
   (f) Use stolen code
   (g) Loss of productivity due to time zone differences
   (h) Hard to ensure code is not reused by another party
   (i) Problems with using certain IDEs
   (j) Problematic when working with larger or multiple files
   (k) IP theft
   (l) Company secrets shared unknowingly
   (m) Cannot always incorporate highly specialised libraries and APIs
   (n) Can be considered distracting for some developers

4. Which of the following Best practices do you think can help mitigate the challenges of AI-assisted software development?
   (a) Get written agreements (who owns the code)
   (b) Ensure original code (no stolen code)
   (c) Confidentiality agreements
   (d) Do not use it for highly sensitive code
   (e) Follow the sun method (Someone is always working on the code)
   (f) Keep core technologies and competencies close to home
   (g) Log and/or watermark your products
   (h) Make use of global pairs
   (i) Using Agile Methods
   (j) Optimise travelling and creating friendships
   (k) Use good communication tools

5. Do you think there are any other best practices not shown here that you think can help mitigate the challenges?