AI Integration inom Programutveckling:

Nuvarande trender och framtida perspektiv

Kandidatuppsats i Informatik

Sebastian Olsson
Muhamedraed Nasser Hameed

VT 2023:KANI23

HÖGSKOLAN
I BORÅS

Svensk titel: AI Integration inom Programutveckling: Nuvarande trender och framtida perspektiv

Engelsk titel: AI Integration in Software Development: Current trends and future perspectives

Utgivningsår: 2023

Författare: *Sebastian Olsson*
Muhamedraed Nasser Hameed

Handledare: Anders Gidenstam och Håkan Sundell

Abstract
This thesis aims to discuss the impact of artificial intelligence (AI) on the software development process, both current and future. The focus is on examining how programmers and testers have come into contact with AI so far and to what extent they believe that AI can be utilized in future software development processes. This is particularly relevant to investigate because AI technology is developing rapidly and there are already tools that can potentially facilitate the work of programmers and testers. The thesis is written in swedish.

The research is of a qualitative nature and includes a literature review to examine previous research on the subject, as well as qualitative semi-structured interviews. The interviews were conducted with both testers and developers to get a more comprehensive picture of the software development field. A total of five semi-structured interviews were conducted where in-depth questions were asked to understand the individuals' thoughts on the subject and their motives.

The results of the study show that our respondents do not use AI to a large extent today. This despite the fact that both the literature and the respondents themselves point out that there are tools that could be used in both processes of programming and testing. One conclusion is that there are various ethical and legal issues that need to be sorted out before the tools can begin to be used. However, both respondents and previous research agree that in the future AI may change the way the software development process is carried out, as AI can be used as a tool to streamline and potentially improve the quality of testing and programming. This can lead to changes in the roles of testers and programmers.

Keywords: AI, ChatGPT, software development, programmers, testers, impact, future trends.

Sammanfattning
Denna uppsats syftar till att diskutera påverkan av artificiell intelligens (AI) på programutvecklingsprocessen, både nuvarande och framtida. Fokus ligger på att undersöka hur programmerare och testare har kommit i kontakt med AI hittills och i vilken utsträckning de tror att AI kan utnyttjas i framtida programutvecklingsprocesser. Detta är särskilt relevant att undersöka eftersom AI-tekniken utvecklas snabbt och det finns redan verktyg som potentiellt kan underlätta programmerares och testares arbete. Uppsatsen är skriven på svenska.

Forskningen är av kvalitativ karaktär och omfattar en litteraturgenomgång för att undersöka tidigare forskning om ämnet, samt kvalitativa semistrukturerade intervjuer. Intervjuerna genomfördes med både testare och utvecklare för att få en bättre täckande bild av området för programutveckling. Totalt genomfördes fem semistrukturerade intervjuer där djupgående frågor ställdes för att förstå individernas tankar kring ämnet och deras motiv.

Resultaten av studien visar att våra respondenter inte i stor utsträckning använder AI idag. Detta trots att både litteraturen och respondenterna själva påpekar att det finns verktyg som skulle kunna användas i båda processerna av programmering och testning. En slutsats är att det finns olika etiska och juridiska frågor som behöver redas ut innan verktygen kan börja utnyttjas.

Dock är både respondenterna och tidigare forskning överens om att AI i framtiden kan förändra sättet som programutvecklingsprocessen utförs på, eftersom AI kan användas som ett verktyg för att effektivisera och potentiellt förbättra kvaliteten på tester och programmering. Detta kan leda till förändringar i rollerna för testare och programmerare.

Nyckelord: AI, ChatGPT, programutveckling, programmerare, testare, effekt, framtida trender.

Innehållsförteckning

1 Inledning 6
1.1 Bakgrund 6
1.2 Forskningsöversikt 6
1.3 Problemdiskussion 7
1.4 Problemformulering 7
1.5 Forskningsfrågor och syfte 8
1.6 Avgränsning 8
1.7 Målgrupp för arbetet 9
2 Metod 9
2.1 Forskningsstrategi 9
2.2 Forskningsmetod 10
2.3 Urval av respondenter 11
2.4 Datainsamling 11
2.5 Dataanalys 13
2.6 Metodreflektion 14
2.7 Respondent reflektion 15
3 Litteratur 16
3.1 Mjukvaruutveckling 16
3.2 Roller inom programutveckling 18
3.3 Artificiell Intelligens 19
3.3.1 Språkmodeller 25
3.3.2 Chatbots 26
3.3.3 ChatGPT 27
3.4 AI:s roll i modern programmering och mjukvaruutveckling 28
3.5 AI Problematik 31
3.5 AI reglering 32
4 Resultat 33
4.1 Anonymitets tabell 33
4.2 Adam 33
4.2.1 Adams bakgrund och erfarenhet 33
4.2.2 Adams upplevelse av AI 33
4.2.3 Adams åsikter om framtid och AI 34
4.3 Peter 35
4.3.1 Peters bakgrund och erfarenhet 35
4.3.2 Peters upplevelse av AI 35
4.3.3 Peters åsikter om framtid och AI 35
4.4 David 36
4.4.1 Davids bakgrund och erfarenhet 36
4.4.2 Davids upplevelse av AI 36
4.4.3 Davids åsikter om framtid och AI 37
4.5 Liam 37
4.5.1 Liams bakgrund och erfarenhet 37
4.5.2 Liams upplevelse av AI 37

4.5.3 Liams åsikter om framtid och AI 38
4.6 Simon 39
4.6.1 Simons bakgrund och erfarenhet 39
4.6.2 Simons upplevelse av AI 39
4.6.3 Simons åsikter om framtid och AI 39
4.7 Hugo 40
4.7.1 Hugos bakgrund och erfarenhet 40
4.7.2 Hugos upplevelse av AI 40
4.7.3 Hugos åsikter om framtid och AI 40
4.8 Kodning 41
4.8.1 Bakgrund och erfarenhet 41
4.8.2 Upplevelse av AI 43
4.8.3 Åsikter om framtid och AI 45
5 Analys och Diskussion 47
5.1 AI inom programutveckling idag 47
5.1.1 Fördelar 50
5.1.2 Problematik 50
5.2 AI i framtiden 52
5.2.1 Hur kan rollerna komma att förändras 54
6 Slutsats 54
6.1 Vidare forskning 57
Referenser 58
Vetenskapliga referenser 58
Icke vetenskapliga referenser 60
Bilagor 60
Bilaga A - Intervjufrågor 60
Bilaga B - Intervjumallen 62

Tabellförteckning

Tabell 1. Begreppslista 3
Tabell 2. Sökord vid sökningar efter litteratur 12
Tabell 3. Appliceringsområden för AI inom programutveckling enligt Harman 23
Tabell 4. Appliceringsområden för AI inom SDLC enligt Shafiq et al. 24
Tabell 5. Grundläggande koncept inom chatbot-teknik enligt Adamopoulou och Moussiades 26
Tabell 6. Appliceringsområden för ML inom SDLC enligt Shafiq et al. 29
Tabell 7. Alias för respondenterna 33
Tabell 8. Respondenternas bakgrund och erfarenhet 41
Tabell 9. Respondenternas bakgrund och erfarenhet i kategorier 42
Tabell 10. Respondenternas upplevelse av AI 43
Tabell 11. Respondenternas upplevelse av AI i kategorier 44
Tabell 12. Respondenternas åsikter om framtid och AI 45
Tabell 13. Respondenternas åsikter om framtid och AI i kategorier 46

Tabell 1. Begreppslista

Begrepp	***Definition***
Artificiell intelligens (AI)	Teorin och utvecklingen av datorbaserade system som kan utföra uppgifter som normalt kräver mänsklig intelligens. (McCarthy, 2007)
Programmering och programmerare	*Programmering* är processen eller aktiviteten att skriva datorprogram eller applikationer. *Programmerare* är termen som används för personer som utför programmeringen. Det vill säga, en person som har en mer övergripande roll i utvecklingen av programvara, inklusive design, planering och testning, utöver att skriva kod. (Blackwell, 2002)
ChatGpt, ChatGPT-3 & Språkmodeller	*ChatGPT är en språkmodell med artificiell intelligens utvecklad av OpenAI. Den använder en djupinlärnings-arkitektur som kallas transformer för att generera mänskliga svar på textbaserade indata. ChatGPT har tränats på en enorm mängd textdata och kan förstå naturligt språk och generera sammanhängande svar i realtid.* *Medan ChatGPT hänvisar till en specifik implementering av GPT-3-modellen designad för interaktiva chatt baserade konversationer, är GPT-3 är det allmänna namnet för själva språkmodellens arkitektur.* *Språkmodeller, som ChatGPT, är AI-algoritmer designade för att bearbeta och förstå naturligt språk. De är utbildade i stora datamängder av text och använder statistiska metoder för att generera textbaserade utdata. Språkmodeller används i en mängd olika appliceringar, som chatbots och taligenkänning.* [1]
Machine Learning (ML)	*Maskininlärning är en utvecklande gren inom beräkningsalgoritmer som är utformade för att efterlikna mänsklig intelligens genom att lära sig från den omgivande miljön.* (El Naqa och Murphy, 2015)

[1] ChatGpt (2023-05-24)

Natural Language Processing (NLP)	*Datorers förmåga att känna igen och förstå mänskligt textspråk.* (Zhang och Lu, 2021)
Software Development Life Cycle (SDLC)	*Livscykeln för programvaruutveckling* som består generellt av faserna, mjukvaru- krav, arkitektur och design, implementering, analys och slutligen underhåll. (Shafiq, et al., 2021)

1 Inledning

1.1 Bakgrund

Inom området för programvaruutveckling inom informatik har intresset för användningen av artificiell intelligens ökat avsevärt. Teknologiska framsteg, särskilt genom språkmodeller som ChatGPT, har trängt fram diskussioner om AI:s roll inom detta område. Denna studie initierades för att adressera behovet av att förstå och utforska AI:s påverkan specifikt på programutveckling. Studien behövs för att förstå AI:s effekter på området, arbetsmarknadsfrågor, utmaningar och etiska aspekter som yrkesverksamma inom programutveckling bemöter, särskilt med uppkomsten av den nya AI-tekniken som ChatGPT. Med AI:s senaste stora framsteg behöver vi proaktivt förstå dess påverkan idag men även i framtiden. Detta för att kunna hålla jämna steg med teknologins utveckling inom området därefter.

Motivationen för studien ligger i den snabbt ökande tillgängligheten av AI och dess möjlighet att förändra hur programutveckling och dess processer genomförs. Behovet av denna studie kastar ljus över huruvida AI-teknologier, såsom ChatGPT, påverkar programutveckling och hur de kan forma framtidens arbetsprocesser inom området. Studien behövs för att bidra med en djupare insikt i hur AI kan integreras i programutveckling, belysa de nuvarande trender men även ge perspektiv och grund för ämnet i framtiden. Genom att förstå detta kommer vi att vara bättre rustade för de förändringar och möjligheter som AI kan medföra inom informatikområdet programutveckling.

1.2 Forskningsöversikt

För att sätta denna studie i rätt sammanhang och förstå dess relevans i den accelererande utvecklingen av artificiell intelligens, är en översikt över relevant forskning avgörande. Flera framstående forskare har bidragit till detta område och deras arbete ger en värdefull insikt i hur AI påverkar olika aspekter av programutveckling. Denna översikten är avsedd att ställa scenen för de kommande avsnitten i denna studie. Dess huvudsyfte är att skapa en sammanhängande sammanfattning av befintlig relevant forskning för att öka förståelsen, identifiera kunskapsluckor och lägga grunden för vår studie inom ämnesområdet som handlar om både de nuvarande och framtida aspekter av AI inom programutveckling.

Zhang och Lus (2021) studie riktar in sig på den pågående teknologiska revolutionen som påskyndas av den omfattande implementeringen av ny generation informations- och kommunikationsteknik som artificiell intelligens. Studien tar form av en litteraturgenomgång och erbjuder en kompakt samt mångfacetterad bild över AI med specifikt fokus på dess integration i olika branscher. Studien kunde uppnå detta genom att välja ut och utforska aktuella artiklar som är kopplade till AI, inklusive dess historik, pådrivande faktorer, teknologiska fundament och tillämpningsområden. Studiens resultat belyser AI:s aktuella tillstånd och potential, därav fungerar den som en referenspunkt för vår studie inom området. Adamopoulou och Moussiades (2020) ger en insikt i chatbot-teknologins tillämpning. Deras studie fokuserar på den snabba utvecklingen av chatbots inom olika områden. De ger en historisk översikt över intresset för chatbots, analyserar drivkrafterna bakom deras användning och betonar deras mångsidiga nytta. Studien inkluderar en diskussion om teknologiska begrepp och en klassificering av chatbots baserat på olika kriterier som kunskapsområde och ändamål. Dessutom presenteras den generella arkitekturen för moderna chatbots och de främsta plattformarna för deras skapande. Slutsatsen av studien är att författarna ser en positiv framtid för chatbots och önskar utforska ämnet mer ingående. Detta

gör denna studien ännu mer intressant eftersom den behandlar ChatGPT som fungerar i form av en chatbot. Shafiq et al. (2021) utforskar integrationen av maskininlärning i olika faser av mjukvaruutvecklingslivscykeln. Metoden som användes var att följa Goal, Question och Metric paradigmen som syftar till att styra studien genom att specificera dess mål för att möjliggöra en målinriktad dataextraktionsprocess. Den hjälper även att koppla mål till formulerade frågor, vilket leder till en bättre tolkning av data i linje med de uppsatta målen. Slutsatsen av studien belyser hur och i vilka faser maskininlärning är mest fördelaktigt inom mjukvaruutveckling.Genom att granska deras studie kan denna studien analysera hur maskininlärning har kommit att integreras i specifika aspekter av utvecklingspraktiken som är relevant till ämnet om AI inom programutveckling. Ernst och Bavotas (2021) arbete undersöker AI-drivna utveckling som integrerar konstgjord intelligens i befintliga programmeringsmiljöer och dess konsekvenser. Metoden som användes involverar litteraturgenomgång och även experimentering med språkmodeller och öppen källkod som automatiserar vanliga programmeringsuppgifter såsom AI-drivna utvecklingsmiljöer, särskilt exemplifierat genom Copilot, som genererar kod och ger förslag till utvecklare. Slutsatsen är att AI-drivna utvecklingsmiljöer har potential att drastiskt förändra programutvecklingsprocessen genom att automatisera tråkiga och repetitiva moment. Samtidigt tas komplexiteter upp kring fördomar i genererad kod, upphovsrättsliga aspekter och påverkan på inlärningsprocessen för programmering.Denna studie bidrar med att ge en förståelse av de förväntade trenderna och utmaningarna med att införliva AI i programutveckling. Genom att relatera till deras synpunkter kan denna studien skapa en koppling till framtidens arbetsprocesser och hur AI kommer att forma dessa.

Det är viktigt att notera att de nämnda studierna i denna översikt utgör endast en kortfattad del av den samlade litteraturen. Dessa studier anses vara närmast knutna till huvudfrågan i denna studie medan resten av litteraturen tas upp i senare avsnitt.

1.3 Problemdiskussion

Samhället genomgår under senaste tiden en betydlig förändring med den snabba tillväxten av artificiell intelligens. AI med sitt exponentiella framsteg, har introducerat nya sätt att lösa problem och utföra uppgifter som tidigare var förbehållna mänsklig kompetens. Ett uppenbart exempel på denna utveckling som också sätts i fokus under denna studie är användningen av språkmodeller som ChatGPT, som har blivit en integrerad del av daglig kommunikation och interaktion. Detta fenomen kan skapa en grundläggande omvandling i hur människor och maskiner samverkar, samtidigt som den har möjlighet att väcka frågor om hur denna förändring influerar och omdefinierar arbetsprocesser inom olika områden, särskilt programutveckling som är fokusområdet för denna studie.

I området för programutveckling är AI inte längre bara en avlägsen möjlighet, utan det har etablerat sig som en påtaglig verklighet. Forskningsinsatser av Zhang och Lu (2021) tydliggör att AI inte längre är begränsat till laboratorier och forskningsprojekt, utan snarare blir det en aktiv del av arbetslivet. Utvecklingen av AI-teknologier, såsom ChatGPT, öppnar nya perspektiv för hur programutveckling som innefattar programmering och testning kan utföras. Samtidigt som dessa framsteg genererar nya möjligheter, ställs vi inför utmaningar och osäkerheter rörande rollerna för yrkesverksamma inom programutveckling.

Den accelererande utvecklingen inom programutveckling och AI för med sig komplexa dilemman. Enligt författare som Ernst och Bavota (2021) menar de att med AI:s ökande förmåga att automatisera rutinmässiga och repetitiva uppgifter stiger frågor om vilken plats det kommer att finnas för mänsklig kreativitet och analys i utvecklingsprocessen. Vidare

pekar Shafiq et al. (2021) på att maskininlärning och AI redan har infiltrerat olika faser av mjukvaruutvecklingslivscykeln. Denna utveckling ger upphov till både fascination och oro över hur yrkesverksamma inom området ska hantera denna nya dynamik och förbli relevanta i en teknikdriven framtid.

1.4 Problemformulering

I samma sammanhang som forskningsöversikten och problemdiskussionen målar upp, söker denna studie att bidra med ny kunskap kring användningen av AI inom programutveckling i den svenska verksamhetsmiljön, med en särskild inriktning på AI-teknologier som ChatGPT. Genom att granska befintlig forskning, såsom Zhang och Lus (2021), Adamopoulou och Moussiades (2020), Shafiq et al. (2021) och Ernst och Bavota (2021), strävar studien efter att fördjupa förståelsen för hur AI integreras i programutvecklingsprocessen och vilken inverkan detta har på arbetsflödet.

Utvecklingen av ny AI-teknologi såsom ChatGPT visar tydligt en ökad kapacitet hos teknikens förmåga att kunna utföra en rad arbetsuppgifter som traditionellt utförs av människor. Detta genererar ett ökat intresse bland samtliga enheter med nära koppling till området. Det är av mer vikt än någonsin att förstå hur AI-teknik för nuvarande används inom programutveckling men även hur den kan i framtiden komma att integrera sig. Denna kunskap är avgörande för att kunna agera proaktivt gentemot förändringar istället för att agera reaktivt. För att möta dessa frågor krävs en djupgående förståelse för användning och integration av AI inom programutvecklingsområdet både idag och i framtiden. Studien tar sikte både nuvarande och framtida aspekter av AI-användning inom programutveckling. Genom att undersöka i vilken utsträckning AI används i dagens arbetsliv, med programmerare och testare som utgångspunkt, kan vi bättre förstå hur teknologin redan har infiltrerat utvecklingsmiljön. Vidare strävar studien efter att belysa hur användningen och integrationen av AI inom programutveckling kan utvecklas i framtiden enligt yrkesverksamma. Detta med perspektiv från tidigare forskning kommer att bidra till att fördjupa diskussionen om nuvarande trender och förväntade möjligheter samt utmaningar med AI inom programutveckling.

Genom att sammanföra och reflektera över den breda samling av forskning och perspektiv som kommer att presenteras, ämnar studien inte endast bidra med ny kunskap utan även öka medvetenheten kring AI:s potentiella integrationer och inverkan i dagens och morgondagens programutveckling i den svenska verksamhetsmiljön. Fokuset ligger både på de nuvarande trender och på framtida perspektiv. Detta för att vi ska kunna anpassa oss och främja en proaktiv inställning till den snabbt utvecklande AI-teknologin och dra nytta av dess potential inom programutveckling.

1.5 Forskningsfrågor och syfte

Denna studie har som avsikt att besvara två centrala frågeställningar. För det första undersöks användningen av AI med särskilt fokus på språkmodeller som ChatGPT, bland programmerare och testare i dagens verksamhetsmiljö i Sverige. För det andra undersöker studien hur användningen och integrationen av AI inom programutvecklingsområdet kan komma att utvecklas i framtiden enligt yrkesverksamma personer.

Baserat på detta har vi valt att utforma följande frågeställningar:

I vilken utsträckning används AI inom programutveckling idag?

Hur förväntas användningen av AI se ut inom programutveckling i framtiden?

Efter att studien framgångsrikt har besvarat de ställda forskningsfrågorna ska den ha bildat en djupare förståelse och insikt om användningen av AI inom programutvecklingsprocessen. Studien ska även ha bildat en djupare förståelse kring AI:s potentiella inverkan på både dagens och framtida programutveckling i den svenska verksamhetsmiljön. Genom en ingående analys av nuvarande tillämpningar och erfarenheter inom programutvecklingsområdet, ämnar studien att belysa omfattningen av integration och utnyttjande av AI-teknologier såsom ChatGPT i relevanta arbetsprocesser. Samtidigt, genom utforskning av åsikter och perspektiv från yrkesverksamma inom branschen, strävar studien efter att analysera de förväntade riktningarna, utmaningarna och möjligheterna som rör implementeringen av AI och dess påverkan på programutvecklingsområdet, med särskilt fokus på programmering och testning.

Efter avslutad studie är avsikten att kasta ljus över de nämnda aspekterna inom ämnet och därmed fylla en kunskapslucka och öka medvetenheten. Utöver detta kan studien fungera som underlag för framtida studier gällande ämnet, samtidigt som den bidrar till en mer omfattande diskussion om AI:s roll inom området för programutveckling.

1.6 Avgränsning

Denna uppsats ämnar undersöka användningen av AI inom programutveckling i dagens verksamhetsmiljö i Sverige, särskilt i relation till språkmodeller som ChatGPT, samt perspektiv för framtida användning av AI inom programutveckling. I sammanhanget för denna uppsats innefattar begreppet programutveckling, både programmering och testning. Även om andra former av AI kan framkomma i respondenternas svar, är det inte huvudfokus för vår undersökning. Vi begränsar oss till de identifierade formerna som är språkmodeller som ChatGPT, men om något relevant uppkommer under undersökningen kan det inkluderas i uppsatsen. Urvalet av respondenter kommer att vara programmerare och testare med erfarenhet av arbete inom yrket i Sverige. Vissa respondenter kan även ha erfarenhet från andra delar av världen, men endast om deras huvudsakliga arbetslivserfarenhet och utbildning som är relevant för denna uppsats har ägt rum i Sverige.

Valet att studien ska begränsas endast inom Sverige beror på vår bas i landet och den begränsade kapaciteten att utforska internationellt. Det är också viktigt att notera att studien, även om den inte kan representera hela Sverige, fortfarande ligger inom ramen för den svenska verksamhetsmiljön med hänsyn till vårt geografiska läge och resursbegränsningar.

1.7 Målgrupp för arbetet

Målgruppen för denna studie omfattar olika grupper av människor. Det inkluderar nuvarande programmerare och testare som är osäkra eller nyfikna på hur AI-utvecklingen kan påverka deras yrke. Det inkluderar också de som är intresserade av att arbeta inom området i framtiden, eftersom de kan finna studien relevant av samma anledning. Företag som är intresserade av att utforska potentialen för att utnyttja AI i sina systemutvecklingsprocesser kan också finna studien intressant. Slutligen tror vi att även lärare och lärosäten kan dra nytta av studien eftersom AI kan påverka hur utvecklare och testare bör utbildas. Denna breda målgrupp utgör främsta anledningen till varför studien är relevant och informativ för olika intressenter inom området.

2 Metod

I detta avsnitt kommer vi att beskriva genomförandet av vår studie. Vi kommer att diskutera forskningsstrategin vi har valt, den metod vi har använt, urvalet av respondenter, insamlingen av data och vår analytiska tillvägagångssätt för att utvärdera resultaten.

2.1 Forskningsstrategi

För att genomföra vår studie har vi valt att tillämpa en induktiv metod. Denna metod innebär att vi utgår från verkligheten och försöker utveckla en teori baserad på observationer och erfarenheter. Genom att använda denna tillvägagångssätt säkerställer vi att den teori vi utvecklar i studien är baserad på verkligheten (Jacobsen, 2017). Valet av forskningsstrategi har ingående övervägts med studiens syfte och forskningsfrågor som grund. En noggrann avvägning av vilka resultat vi vill uppnå med studien har lett oss till att välja en detaljerad och holistisk undersökning av ämnet. Vi valde den induktiva metoden eftersom vi ansåg att den lämpligast möjliggör tolkning och förståelse av ämnet utifrån erfarenheter och perspektiv för att senare konstatera en teori, vilket är av stor relevans för vår studie som ämnar att undersöka verkligheten. Under hela processen har vi även ständigt förstärkt vårt val av metod genom att söka expert åsikter från vår professor och handledare vid vårt institut. Därmed kunde vi bättre säkerställa att vår valda forskningsstrategi är förankrad i forskningens syfte och uppfyller kraven för en noggrant utförd akademisk studie. Studien antar en kvalitativ forskningsansats, vilket innebär att vi fokuserar på att förstå ämnet på en djupare nivå istället för att samla kvantitativa data eller kombinera olika metoder.

Valet av en kvalitativ forskningsansats motiveras av vårt önskemål att fördjupa förståelsen för ämnet genom en detaljerad och holistisk undersökning. En kvalitativ forskningsansats är lämplig när man vill tolka och förstå ämnet utifrån erfarenheter och perspektiv. Denna metod passar även bra för att utforska frågor relaterade till människors uppfattningar, attityder och beteenden. Därför väljer vi en kvalitativ forskningsstrategi för denna studie, för att uppnå en djupare insikt i det ämne vi avser att utforska. Genom att använda en kvalitativ forskningsstrategi kan vi genomföra en mer utforskande studie av ämnet, vilket innebär att vår studie har ett mer detaljerat och personligt perspektiv (Jacobsen, 2017).

För denna studie är den kvalitativa forskningsansatsen idealisk för att ge insikter i användningen av AI bland programmerare och testare i den svenska verksamhetsmiljön, samt integreringen av AI-teknologier i programutvecklingsprocessen och framtida utveckling, utmaningar och möjligheter inom området. Med den kunskap som har genererats genom studien kan vi dra slutsatser som besvarar våra forskningsfrågor på ett tillfredsställande sätt.

2.2 Forskningsmetod

Studien genomfördes med en kvalitativ ansats, där semistrukturerade intervjuer användes som primär metod för datainsamling. Valet att omfamna en kvalitativ ansats var väl genomtänkt och påverkades av det begränsade antalet tidigare forskningsinsatser inom ämnet. Denna begränsning skapade specifika utmaningar under utförandet av studien. Ett exempel på en utmaning när det gällde valet av semistrukturerade intervjuer var att utforma lämpliga frågor för exempelvis en kvantitativ studie. En metod som tillät studien att dyka in djupt i ämnet var behövligt, en kvalitativ ansats var ett rimligare val baserat på detta. Utmaningarna antydde inte särskilt att studien uteslutande kunde endast omfamna en kvalitativ metod, dock med detta förstått, beslutet grundades på det faktum att den nya AI-tekniken som undersöktes i denna studie, till exempel ChatGPT, är relativt nyligen introducerad och ämnet har inte ännu

utforskats i djupet av många forskare. Det är därför vi valde att bidra till att fylla den kunskapslucka som existerar genom vår studie. I valet av metod var det avgörande att noga överväga metodens kapacitet att ge en grundlig förståelse av ämnet samtidigt som studiens omfattning och pålitlighet bevarades. Förutom de semistrukturerade intervjuerna genomfördes också en litteraturgenomgång för att samla in data. Genom att kombinera dessa metoder kunde vi jämföra våra upptäckter och slutsatser med tidigare forskning, vilket ökade validiteten och pålitligheten i vår analys och slutsats.

Andra alternativa metoder som övervägdes men till sist valdes bort inkluderade kvantitativa enkäter, en mixad metod samt utvecklingsarbete. I början övervägdes en mixad metod där de kvalitativa intervjuerna skulle kombineras med kvantitativa enkäter för att testa om de resultat som framkom under intervjuerna representerade en allmän uppfattning eller om de endast var enskilda personers åsikter. Dock valdes denna metod bort på grund av tidsbegränsningar. Vid valet av en passande kvalitativ och kvantitativ metod för vår studie kändes valet enkelt. Eftersom det fanns begränsad tidigare forskning inom ämnet och avsaknaden av tillräckligt med tid för att genomföra båda metoderna skulle ha påverkat studien negativt, valdes en kvalitativ metod som studiens huvudsakliga tillvägagångssätt.

En sista metod som övervägdes var utvecklingsarbete med en designmetod. Denna metod valdes dock bort på grund av den omfattande arbetsinsatsen som krävs i förhållande till uppgiftens omfattning. Även om utvecklingsarbetet skulle ha varit intressant för att vidareutveckla vår studie, ansåg vi att den extra tiden som krävts skulle ha påverkat uppsatsens kvalitet negativt.

Det bör noteras att valet av kvalitativ datainsamling i vår studie har vissa svagheter som måste beaktas. Enligt Jacobsen (2017) är kvalitativ datainsamling resurskrävande, eftersom intervjuerna kan ta lång tid att genomföra på det sätt som vi planerade. Dessutom kan resultaten vara svåra att generalisera eftersom endast ett fåtal personers åsikter samlades in. Det finns också en risk att öppna intervjuer genererar komplexa svar, vilket kan göra tolkningen och bearbetningen av data svår. Slutligen kan ny information upptäckas under studiens gång, vilket kan göra att studien aldrig känns fullständig.

2.3 Urval av respondenter

Vi genomförde intervjuer med ett urval av relevanta personer för att samla in data. Urvalet bestod av personer med erfarenhet eller expertis inom programmering och testning. För att göra detta använde vi en strategisk urvalsmetod där vi valde ut personer med den specifika kunskapen som krävdes (Jacobsen & Andersson, 2017). Vi kontaktade personer som hade erfarenhet av att utveckla eller testa programvara i sitt yrkesliv, eller personer som var insatta inom ämnet och hade värdefulla insikter att erbjuda. Totalt genomförde vi sex intervjuer som varierade i längd, men alla över 1 timme och 20 minuter vardera. Respondenterna kom från fem olika företag och var geografiskt spridda över landet.

Varje respondent intervjuades en gång, och om vi vid transkriberingen upptäckte att vi hade missat något viktigt för studien, kontaktade vi respondenten igen för att klargöra det. Vi hade inga specifika krav på ålder eller kön vid urvalet av personer. Istället strävade vi efter en bra blandning av erfarenhet inom området samt en balans mellan personer som arbetade med testning, utveckling eller både och. Vi arbetade också aktivt med att ha alternativ tillgängliga om någon av de kontaktade respondenterna inte kunde genomföra intervjun. Detta innebar att vi ständigt sökte efter fler personer att intervjua, även efter att vi hade nått det ursprungliga

målet på sex personer. Syftet var att undvika att hamna i tidsnöd om någon respondent skulle falla bort.

Vi använde tre olika metoder för att rekrytera respondenter. För det första publicerade vi inlägg på LinkedIn där vi kortfattat beskrev vår studie och vilka typer av respondenter vi sökte. Vi erbjöd yrkesverksamma möjligheten att kontakta oss om de var intresserade av att delta. Den andra metoden var att skicka e-post till företag inom mjukvaruutveckling och erbjuda intresserade personer att kontakta oss. Den sista metoden var att använda våra egna kontakter för att försöka hitta personer som skulle vara villiga att delta i vår studie. Vi kontaktade direkt testare och utvecklare.

2.4 Datainsamling

Under studien genomförde vi två typer av datainsamling. Den första metoden var en litteraturgenomgång där vi granskade relevant litteratur, akademiska studier och forskningsartiklar om AI och programutveckling. Vi utförde denna datainsamling för att stödja vårt litteratur kapitel. Vi valde olika källor som belyser ämnet i vår studie. Källorna inkluderar samtida forskning, etablerade teorier och praktiska exempel för att ge en övergripande förståelse. Till exempel var vetenskapliga artiklar och konferensbidrag om den senaste AI-utvecklingen avgörande för att fånga framsteg och trender. Teoretiska källor bidrog till vår förståelse av hur AI integreras i programutveckling, medan äldre källor etablerade grunden för exempelvis programutveckling, vilket skapade stabilitet och gav relevanta termer. Bland våra källor återfinns både nya och etablerade för att säkerställa en pålitlig diskussion. Valet av äldre källor gjordes för att bygga på en etablerad grund, på grund av bristen på ny litteratur. Detta beslut grundade sig i att vissa termer redan har definierats och behöver inte ny forskning för att fastställas igen, till exempel "programmering" och "mjukvaruutveckling". Sammantaget strävade vårt källurval efter att ge en djup och mångsidig förståelse av AI inom programutveckling, vilket stöder vår studie och ger insikter inom detta område.

För att samla in befintlig information använde vi sökmotorer som Primo, Google eller Google Scholar. Primo och Google Scholar var våra primära databaser eftersom de genererar resultat av akademisk karaktär i högre grad. Om vi inte kunde hitta den sökta informationen i dessa databaser, använde vi tillförlitliga webbplatser som behandlar ämnen av relevans för att komplettera våra kunskaper. Vid sökningarna i de olika databaserna använde vi olika fraser och sökord relaterade till de ämnen vi ville hitta litteratur om. Sökningarna kunde vara på svenska men även på engelska och inkluderade, men var inte begränsade till:

Tabell 2. Sökord vid sökningar efter litteratur

Ämne	***Sökningar***
AI	Vad är AI, Historiken av AI, AI appliceringar, Språkmodeller, ChatGPT, GitHub Copilot, Chatbots
Programutveckling	Mjukvaruutveckling
AI inom programutveckling	AI inom mjukvarutestning, AI inom mjukvaruutveckling, ChatGPT inom mjukvaruutveckling

Övrigt	Etiska dilemman med AI, etiska dilemman med användningen av AI inom mjukvaruutveckling

Efter att vi genomfört sökningar gick vi igenom ett antal träffar för att hitta relevanta artiklar för vår studie. Vi strävade alltid efter att hitta flera artiklar relaterade till varje ämne för att få en mångfald av perspektiv. Det var dock viktigt att vi noggrant valde trovärdiga och relevanta artiklar, vilket gjorde urvalet av artiklar något tidskrävande.

Vi genomförde även kvalitativa intervjuer med nuvarande programmerare och testare inom branschen. Intervjuerna genomfördes som semistrukturerade intervjuer för att samla in respondenternas tankar och erfarenheter kring ämnet. Valet att använda semistrukturerade intervjuer, istället för strukturerade intervjuer, gjordes för att undvika att begränsa kvaliteten och mängden av insamlad data. Genom att tillåta oväntade men intressanta upptäckter kring ämnet skulle vi ha en mer omfattande och högkvalitativ datamängd. En semistrukturerad intervju klassas som en kvalitativ forskningsteknik som faller mellan strukturerade och ostrukturerade intervjuer. I våra semistrukturerade intervjuer följde vi en förutbestämd uppsättning frågor samtidigt som vi tillät flexibilitet att utforska djupare svar och följa oväntade diskussionsgrenar. Detta tillvägagångssätt kombinerade fördelarna med standardiserade frågor med anpassningsförmågan att ta itu med individuella perspektiv, vilket skulle leda till rikare insikter och en mer omfattande förståelse av intervjupersonens erfarenheter, åsikter och tankar. För att säkerställa att de semistrukturerade intervjuerna uppfyllde sitt syfte förberedde vi öppna frågor som fokuserade på ämnet. Dessutom använde vi en reaktiv metod genom att ställa följdfrågor baserat på respondenternas svar.

Intervjuerna baserades på en förberedd intervjumall som utformats för att säkerställa att insamlad information skulle vara relevant för vår studie. Mallen var strukturerad på det sättet att varje fråga hade en huvudfråga och målformulering, vilket underlättade för intervjuaren att samla in önskad information. Om den inledande frågan inte genererade svar som uppfyllde alla målformuleringar ställdes följdfrågor för att säkerställa att alla mål uppnåddes.

Intervjumallen bestod av fem delar. Första delen var en introduktion där vi beskrev studien och gav bakgrundsinformation om vårt val av ämne. Detta syftade till att respondenten skulle kunna ge sitt informerade samtycke att delta i intervjun och förstå vad vi undersökte. Det hjälpte också till att säkerställa att respondenten gav svar som var relevanta för vår studie och inte avvek och diskuterade AI i ett helt orelaterat sammanhang. Den andra delen omfattade frågor om sekretess och samtycke, där respondenten bekräftade sitt medgivande att vara en del av studien och att vi spelade in intervjun. Dessutom gav det respondenten möjlighet att ställa eventuella frågor innan vi började samla in information från dem. I nästa del samlade vi in information om respondentens bakgrund, som inkluderade deras arbetslivserfarenhet och nuvarande arbete. Detta var viktigt för att bekräfta att respondenten var en relevant person att intervjua och för att ge oss en bakgrund inför kommande frågor. Den fjärde delen syftade till att få en förståelse för respondentens tidigare erfarenheter av AI i arbetslivet eller på annat sätt. Slutligen fokuserade den sista delen på respondentens åsikter om de tror och i det fallet hur de tror att AI kommer att användas inom programutveckling och dess processer i framtiden.

Det är viktigt att notera att eftersom det var semistrukturerade, öppna intervjuer, var de frågor som inkluderades i mallen endast en mindre del av det totala antalet frågor som ställdes under

intervjuerna. Följdfrågor baserades på respondenternas svar till huvudfrågorna från mallen för att ge en tydligare bild av situationen. För en detaljerad referens till intervjufrågorna och intervjumallen, se bilaga A och B.

Intervjuerna genomfördes digitalt via plattformar som Zoom för att säkerställa en geografisk spridning av respondenterna. För att skapa en mer mänsklig interaktion och främja delaktighet användes kameror under intervjuerna. Inspelning av intervjuerna gjordes med respondenternas tillstånd, vilket beviljades av samtliga intervjuade personer.

Avslutningsvis är det värt att notera, den data som samlas in från litteraturgenomgången och de kvalitativa semistrukturerade intervjuerna underkastas en analysprocess för att identifiera centrala teman och mönster. Genom denna analys kan vi dra slutsatser och presentera resultat som besvarar de huvudsakliga frågeställningarna i studien.

2.5 Dataanalys

Vi utförde transkribering av de insamlade intervjuerna genom att använda en transkriberingsverktyg som omvandlade ljudfilerna till text. För att säkerställa noggrannheten granskade vi transkriberingen samtidigt som vi lyssnade på ljudfilen. Detta innebar att vi kompletterade datan med manuell transkribering av intervjuerna för att inte missa någon relevant information. Efter transkriberingen genomförde vi en analys av intervjuerna för att säkerställa att tillräckligt med data hade samlats in för att kunna besvara studiens forskningsfrågor.

För analysen använde vi innehållsanalys, en metod beskriven av Jacobsen (2017) som innebär att man analyserar och kategoriserar textinnehållet. Genom att tillämpa innehållsanalys identifierade vi teman och ytterligare kategorier i den insamlade datan. Vi använde även öppen kodning, vilket innebär att vi aktivt sökte efter mönster och teman för att få en djupare förståelse av datan (Jacobsen, 2017). Genom att tillämpa dessa analysmetoder fick vi en mer omfattande bild av den data vi arbetade med.

2.6 Metodreflektion

Metoderna för denna studie är noga övervägda och baseras på studiens forskningsfrågor och syfte. Vi har valt att använda en kvalitativ forskningsansats snarare än en kvantitativ eller blandad metod. En kvalitativ forskningsmetod är lämplig för att förstå ett ämne och få en djupare förståelse för det. Enligt (Jacobsen 2017) är fördelarna med en kvalitativ forskningsmetod att det ger djupare förståelse av ämnen, möjliggör utforskning av nya aspekter som uppstår under forskningsprocessen, ger en mer personlig och detaljerad bild av deltagarnas perspektiv och är mer lämplig för att undersöka frågor relaterade till människors uppfattningar, attityder och beteenden. Detta är delar som är relevanta för den studie som vi valt att genomföra.

Enligt Jacobsen (2017) framkommer nackdelar med kvalitativ forskning, såsom att det kan vara resurskrävande. Det innebär att det krävs betydande tid, ansträngning och resurser för att genomföra sådan forskning. Detta beror på att kvalitativ forskning oftast involverar mer djupgående och detaljerade metoder för datainsamling, exempelvis intervjuer eller observationer, vilket kan ta mycket tid att genomföra och analysera. I detta studiefall användes intensiva litteraturgenomgångar och djupgående intervjuer som datainsamlingsmetoder, vilka båda krävde betydande tid och ansträngning.

En ytterligare nackdel med kvalitativ forskning enligt Jacobsen (2017) är svårigheten att generalisera resultaten till en större population. Eftersom kvalitativ forskning vanligtvis syftar till att förstå och utforska specifika fenomen och perspektiv, som i detta fall involverade ett begränsat antal intervjuade programmerare och testare, kan det vara svårt att generalisera resultaten till en bredare population eller att göra allmänna påståenden som kan tillämpas utanför den specifika kontexten som studeras. Jacobsen (2017) menar även att kvalitativ forskning kan också vara komplex och svår att tolka och analysera på grund av dess omfattande och detaljerade data, vilket kräver en mer krävande och komplex analysprocess. För att undvika felaktiga slutsatser eller missförstånd som kan leda till bristande tillförlitlighet och validitet i studien måste vi vara försiktiga vid tolkning och dra slutsatser från insamlade data.

En annan faktor enligt Jacobsen (2017) som kan påverka kvaliteten i kvalitativ forskning är närheten mellan forskaren och deltagarna. Forskaren kan bli för involverad eller påverkad av deltagarna, vilket kan leda till brist på objektivitet och påverka forskningsresultaten. En till liknande faktor som kan spela en roll i hur metoden kan brista är intervjuareffekten i kvalitativ forskning, där deltagarna kan komma att snedvrida svar för att motsvara forskarens förväntningar.

Slutligen, enligt Jacobsen (2017), kan flexibilitet inom kvalitativ forskning också vara en nackdel. Eftersom forskningsdesign och datainsamlingsmetoder ofta anpassas efter forskarens behov och forskningsfrågor, kan det vara svårt att jämföra och replikera studier eller dra generella slutsatser.

Med tanke på dessa möjliga nackdelar har vi planerat att genomföra flera steg för att bemöta dem och säkerställa kvaliteten i vår kvalitativa forskning. Först och främst har vi noggrant övervägt och valt en lämplig forskningsstrategi och metod för datainsamling och analys. Genom att vara medvetna om resurskraven kunde vi allokera tillräckligt med tid och insats för att genomföra en noggrann och tillförlitlig studie. Dessutom var vi realistiska när det gäller förväntningarna på tidsåtgång och såg till att vi hade tillräckligt med resurser tillgängliga för att genomföra studien på ett tillfredsställande sätt.

För att bemöta svårigheten att generalisera resultaten till en större population var vi tydliga med att vår studie fokuserar på specifika fenomen och perspektiv. Vi anger de specifika kontextuella begränsningarna i vår uppsats och är försiktiga med att dra allmänna slutsatser eller generalisera våra resultat utanför den specifika populationen eller kontexten. Dessutom strävade vi efter att inkludera en mångfald av respondenter och datakällor för att få en bredare förståelse för det undersökta fenomenet och öka trovärdigheten i våra slutsatser.

När det kommer till att hantera komplexiteten av en kvalitativ forskning och svårigheterna med att tolka och analysera omfattande, men även detaljerade data, kommer vi att använda oss av en systematisk analysprocess.

Utöver svårigheter ovan, för att hantera nära relationen mellan oss och respondenterna samt minimera risken för påverkan och brist på objektivitet var vi medvetna om våra egna förutfattade meningar och förväntningar. Dessutom följde etiska riktlinjer och säkerställde konfidentialitet och anonymitet för att skapa en trygg miljö för respondenterna att dela sina upplevelser och åsikter.

Slutligen rapporterade vi våra resultat på ett transparent och trovärdigt sätt genom att använda oss av detaljerade beskrivningar av vår studiedesign, metodologi, datainsamlings- och analysprocess. Vi inkluderade relevanta citat och exempel från vår data för att bygga våra slutsatser och göra det möjligt för läsarna att bedöma kvaliteten och validiteten i vår studie. Vi var öppna för feedback och granskning av vår studie för att främja replikering och vidareutveckling av kunskapen inom området. Genom att vidta dessa åtgärder strävade vi efter att maximera tillförlitligheten, validiteten och reproducerbarheten i vår kvalitativa forskning.

2.7 Respondent reflektion

De respondenter som har medverkat i intervjuerna utgör en diversifierad grupp inom området för programutveckling, vilket ger en bred insikt i utvecklares och testares erfarenheter och åsikter om AI både idag och i framtiden. Genom att inkludera utvecklare och testare med olika yrkeserfarenheter och expertisområden skapas en mångfald av perspektiv som berikar förståelse. Under intervjuerna har vi haft förmånen att intervjua fyra utvecklare, varav två, Peter och David, innehar en betydande erfarenhet på över 10 år inom yrket. Deras långa yrkesbana ger en djup förståelse för utvecklingsområdet och utmaningar inom det. De övriga två utvecklarna, Liam och Hugo, kompletterar gruppen med utbildning inom systemvetenskap. Denna variation i utbildningsbakgrund mellan yrkeshögskoleutbildningar och universitetsexamen bidrar till en bredare kunskapsbas och olika perspektiv på programutvecklingsprocessen och tillämpningar av AI inom den.

Genom att involvera utvecklare som arbetar på olika företag och med olika typer av system, har vi säkerställt en bredd av erfarenheter och arbetsmiljöer. Denna diversitet gör det möjligt att utforska både gemensamma utmaningar och specifika kontextuella faktorer inom programutveckling i relation till AI och dess appliceringsområden. Mångfalden i de olika teknologier, plattformar och appliceringsområden som används av våra respondenter ger en omfattande överblick över utvecklarnas perspektiv och ger oss möjlighet att identifiera gemensamma teman och mönster. Utöver utvecklarna har vi också inkluderat två testare i våra intervjuer för att komplettera diskussionen som berör programutvecklingsområdet. Simon, med sin examen inom systemvetenskap, tillför en teoretisk grund och förståelse för testmetoder och systemvalidering. Å andra sidan bidrar Adam med sin erfarenhet från en utbildning inom datateknik, vilket ger en aktuell insikt i aktuella testpraktiker och utmaningar inom programutveckling i relation till AI.

Sammanfattningsvis har vi genomfört intervjuer med en grupp deltagare som uttrycker ett starkt intresse för teknik och programutveckling. Respondenterna har varierande yrkeserfarenhet inom området, vilket ger oss en god bredd av deltagar-typer. Vi har inkluderat både utvecklare och testare i vår grupp, vilket är särskilt gynnsamt eftersom vårt syfte är att förstå både programmering och testning i relation till artificiell intelligens, både nu och i framtiden. Detta är av särskild vikt eftersom Fishpool, B. & Fishpool, M. (2020) konstaterar att utvecklare ibland kan ha ansvar för testning som en del av deras arbetsuppgifter. Emellertid framkom det från våra respondenter, David och Liam, att de arbetar i agila teams där det finns specialiserade testare. Respondenterna berättade att dessa testare har ansvar för att genomföra slutgiltiga tester av produkterna åt dem. Mot bakgrund av detta var det av betydelse att inkludera testare i våra intervjuer för att säkerställa en omfattande täckning av test-aspekten. De tester som vi intervjuade berättade att de samarbetar nära utvecklarna och att de besitter programmeringskunskaper. I Simons fall är hans framtida målsättning till och med att arbeta med utveckling. I Hugos fall utför han vissa tester själv som utvecklare innan koden skickas vidare till dedikerade testare. Detta innebär att deras åsikter kan tas i beaktning

även när det kommer till hur de tror att AI kan användas inom programutveckling i framtiden, baserat på att de besitter insikter från båda perspektiven.

Genom att inkludera deltagare med olika erfarenheter, utbildningsbakgrunder och arbetsmiljöer har vi erhållit en bred kunskapsbas och en mångfald av perspektiv att analysera. Denna mångfald av perspektiv stärker validiteten och tillförlitligheten hos våra studieresultat och ger en omfattande förståelse för programmerares och testares erfarenheter och åsikter om AI både i dagens sammanhang och med blick mot framtiden.

3 Litteratur

Litteraturkapitlet kommer att noggrant undersöka ämnen som är relevanta för vår studie. Syftet är att skapa en djupare förståelse för ämnet och samtidigt stödja den diskussion och analys som genomförs i nästa kapitel. Genom att fördjupa oss i relevant litteratur kan vi bygga en pålitlig grund för vår undersökning och stärka våra argument och slutsatser i studien.

3.1 Mjukvaruutveckling

Mjukvaruutveckling är den svenska översättningen av det engelska ordet *Software engineering*. Humphrey (1989) definierar mjukvaruutveckling som helheten av den process som genomförs för att gå från användarkrav till en färdig mjukvara. De beskriver hur detta kan inkludera processer som kravinsamling, design, implementation, verifikation, installation, operativt stöd samt dokumentation. Alla dessa är dock endast exempel på vad som kan ingå och det kan skilja sig mellan projekt. Vidare beskriver författaren att förbättringar och vidareutveckling är ytterligare två delar som kan ingå i mjukvaruutvecklingsprocessen, i det fall bara över en kortare tid efter det att man implementerat mjukvaran.

Bourque, Dupuis, Abran, Moore och Tripp (1999) har skapat en modell som de kallar för *Guide to the software engineering body of knowledge*. Modellen ämnar bland annat att karaktärisera de olika delarna som ingår i *Software engineering processen*. Modellen har tio stycken huvudområden som tillsammans ska inkapsla hela processen. De tio områdena som författarna har beskrivit är: software configuration management, software construction, software design, software engineering infrastructure, software engineering management, software engineering process, software evolution and maintenance, software quality analysis, software requirements analysis, och software testing.

Författarna beskriver *Software configuration management* som processen av att identifiera och ändra på konfigurationen av en produkts hårdvara eller mjukvara. Under denna process ingår även dokumenteringen av de förändringar som görs. Detta för att man ska ha kontroll över produkten under hela dess livscykel.

Software construction beskrivs som den delen av processen där en programmerare kodar för att skapa något. Dock är det inte endast kodningen som ingår i denna del, här ingår även validering av det man skapar samt enhetstestning.

Software design beskrivs som den del där man tar de insamlade kraven man har för den produkt som ska skapas och omvandlar dem till en potentiell lösning. I denna del beskriver man hur systemet är uppbyggt samt hur olika delar samverkar med varandra. Man beskriver allt på en sådan nivå att det sedan är möjligt att påbörja utvecklandet av en artefakt.

Software engineering infrastructure överlappar något med andra delar eftersom den beskriver tre delar tillsammans. Dessa tre är utvecklingsmetoder, mjukvaruverktyg och komponent integration. Utvecklingsmetoder är metoder för hur mjukvaruutvecklingen samt underhållet ska gå till. Målet är att ge ökad struktur och på det sättet åstadkomma bättre resultat. Mjukvaruverktyg är de datorbaserade verktyg som används för att underlätta processen att skapa en artefakt. Dessa verktyg kan användas i olika utsträckning under olika delar av en process beroende på arbetet som utförs. Komponent integration är den del som beskriver hur man får flera komponenter att samarbeta för att skapa en helhet. Den beskriver även hur

individuella komponenter ska behandlas samt hur man ska arbeta för att kunna återanvända komponenter (Bourque et al., 1999).

Författarna beskriver *Software engineering process* som en del som hanterar två delar relaterade till varandra. Dessa två är processhantering samt resultatmätning. Processhantering hanterar stora frågor som vilken typ av projekt man ska ta sig an, vilka standarder man ska följa samt hur man ska utveckla de relevanta team som arbetar med projekten. Mätningen handlar om att personen ska kunna få en förståelse för kvalitén av det man genomför. För att göra detta fokuserar man på vad som ska mätas, varför det mäts, hur man samlar in data för mätningarna samt hur man använder den information man får in för att förbättras.

Software engineering process handlar enligt författarna om definitionen, implementationen, mätningar, hantering, förändring samt förbättring av mjukvaruprocesser.

Software evolution and maintenance förklarar författarna med att delen *Software maintenance* definieras av IEEE Standard 1219-1998, IEEE Standard for Software Maintenance. Denna beskriver hur man ska hantera modifiering samt förbättringar av en produkt efter att den satts i bruk. Författarna menar dock att ett system aldrig blir komplett och alltid behöver utvecklas för att hålla sig relevant, därför inkluderas även *Software Evolution.* De beskriver hur denna delen har ett stort ansvar för att hålla kostnaderna till en rimlig nivå samtidigt som man säkerställer kvaliteten av systemet.

Software quality analysis är enligt författarna processen av att säkerställa att ett system är av hög kvalitet. För att genomföra detta finns det systematiska åtgärder att vidta. Två delar som beskrivs som grundläggande är verifiering och validering eftersom dessa ger en objektiv syn på kvaliteten. De beskriver även hur mjukvarukvalitetsanalys kan delas upp i fyra områden. Dessa är definition av kvalitetsanalys, processplaner, aktiviteter och tekniker för kvalitetsanalys och mätning av mjukvaru kvalitetsanalys.

Software requirements analysis kan delas upp i fem områden. Dessa är *requirements engineering process, requirements elicitation, requirements analysis, requirements validation* och *requirements management*. Det första området, *requirements engineering process*, introducerar processen och orienterar de fyra andra områdena. Här diskuteras även frågor om kontrakt och projektorganisation. Det andra området, *requirements elicitation,* handlar om att samla in och upptäcka krav. Det är en mänsklig aktivitet som även identifierar intressenter och etablerar relationer mellan utvecklingsteam och kunder som kan vara användbara under processen. Det tredje området, *requirements analysis*, fokuserar på att analysera kraven för att upptäcka och lösa konflikter däremellan. Det ska även upptäcka gränserna för systemet och hur det måste interagera med sin miljö samt utveckla användarkraven till mjukvarukrav (Bourque et al., 1999)

Requirements validation är det fjärde området, vilket ämnar att undersöka att rätt krav upptäckts, att man har tillräckligt många krav samt att de är bra beskrivna. Det sista området är *requirements management*, vilket ansvarar för att se till att kraven uppdateras under hela livscykeln för att se till att de ställda kraven återspeglas i programvaran. *Software testing* handlar om att kontrollera att systemet fungerar som förväntat, vilket utförs genom en rad olika testfall. Bourque et al. (1999) beskriver hur det finns en mängd olika testtekniker som kan hjälpa till att säkerställa olika nivåer av kvalitet på programvaran.

3.2 Roller inom programutveckling

Fishpool, B. & Fishpool, M. (2020) beskriver rollen *Software developer*, vilket på svenska kallas mjukvaruutvecklare eller programmerare, som en person med ansvar för att skriva säker kod. Detta säkerställs genom att följa ett testdrivet tillvägagångssätt samt att koden ska vara öppen och enkel för andra att återanvända. Författarna listar en mängd olika typer av *Software developers*. Dessa är web-developers, mobile app developers, game-developers, software developers och application developers.

De beskriver att trots skillnaderna mellan alla 5 är deras grundläggande arbetsuppgifter fortfarande väldigt lika. Huvuduppgiften för alla 5 är att bygga och testa enkel kod av hög kvalitet. För att göra detta behöver personerna som utför de fem olika rollerna väldigt lika kunskaper i form av analysering, programmering och testning. De olika typerna görs dock ofta i olika språk vilket skiljer dem från varandra. Författarna tar även upp att det finns olika subkategorier av *Software Developers* . Rollerna skiljer sig på det sätt att vissa kräver att man har en högre nivå av kunskap än vad andra kräver. Detta gäller både på en teknisk nivå men även på en ledarskapsnivå. Det förväntas därmed att de mer seniora utvecklare rollerna både ska kunna skriva kod på ett bättre sätt utifrån den erfarenhet och kunskap de besitter samt kunna tillämpa en högre standard vid utbildning. Rollerna som författarna listar är, apprentice developer, junior developer, developer, senior developer, lead developer och principal developer.

Enligt samma författare kan arbetsuppgifterna för en *Software developer* vara svårdefinierade. Detta beror på att det finns en mängd olika faktorer som avgör vilket ansvar och vilka arbetsuppgifter man har. De beskriver även hur samma roll kan skilja sig mellan olika organisationer vilket gör att även om man har exakt samma roll måste inte det innebära att man genomför samma uppgifter. De har dock tagit fram en lista med en mängd olika aktiviteter som kan förekomma för olika personer med någon form av *Software developer* roll. Fishpool, B. & Fishpool, M. (2020) skriver att framförallt i mindre företag brukar en utvecklare få vara en del av de flesta aktiviteterna. Listan innehåller aktiviteter som att vara med på samt hålla i möten med kunder, analysera kunder och slutanvändares krav, ta fram tekniska specifikationer för utvecklingen, skriva nykod, i olika programmeringsspråk, som håller organisationens standard, modifiera och förbättra existerande programkod för att förbättra dess effektivitet samt prestanda eller för att addera ny funktionalitet, utföra systemintegrationer genom datautbyte eller limtekniker för företag, inklusive för äldre system, skriva teknisk dokumentation för system, testa kod funktionalitet för att upptäcka och sedan fixa eventuella fel, arbeta i ett team bestående av många olika roller, till exempel UX-designers (User experience), grafiker, andra utvecklare, systemtestare och projektledare, undersöka nya tekniker och teknologier, installera, uppdatera och konfigurera nya databaser, tjänster, system och hårdvaror, utveckla dig själv genom att lära dig nya tekniker och uppdatera din nuvarande kunskap och följa och hålla koll på principerna av (GDPR).

3.3 Artificiell Intelligens

Vi har tidigare i uppsatsen tagit upp olika perspektiv på definitionen av artificiell intelligens. Schanks kritiska perspektiv från 1987 betonade behovet av AI-forskning för att gå bortom symbolisk bearbetning och utveckla mer kognitiva tillvägagångssätt. Zhang och Lus definition (2021) återspeglade den breda samsynen inom området AI, vilken omfattar maskininlärning, naturlig språkbehandling, robotik och datorseende. Schuetts argument för en juridisk definition av AI baserad på autonomi, förmåga att lära sig och förmåga att agera betonade behovet av att ta itu med juridiska och etiska problem. Denna överblick har gett en omfattande översikt över olika synpunkter på definitionen av AI.

För att säkerställa en rigorös studie är det viktigt att tydligt definiera vad artificiell intelligens (AI) är. Utöver de tidigare nämnda definitionerna inkluderar vi även beskrivningen av AI som *vetenskapen och tekniken för att skapa intelligenta maskiner, särskilt intelligenta datorprogram* enligt McCarthy (2007). Det är relevant att inkludera denna definition på grund av författarens betydelse i fältet. John McCarthy anses vara en av de mest kända amerikanska datavetarna och betraktas som en av grundarna av AI. Vid Dartmouth-konferensen 1956, som ofta anses vara starten på AI som ett vetenskapligt ämne, myntades uttrycket *artificiell intelligens* och McCarthy är en av de krediterade för detta. Utöver det är hans arbete inom AI och betydande bidrag till dess utveckling och andra områden inom datavetenskap välkända.

McCarthy (2007) hävdar att efter andra världskriget började ett antal individer separat arbeta på intelligenta maskiner, vilket var början på AI-forskning. Den första var möjligen den engelske matematikern Alan Turing när han presenterade ett föredrag om det 1947. Han kan även ha varit den första att inse att programmering av datorer snarare än att skapa maskiner var det perfekta sättet att studera AI. Utöver Alan fanns flera AI-forskare i slutet av 1950-talet och majoriteten av dem baserade sin forskning på datorprogrammering.

En djupare insikt på AI:s historiska utveckling ges av Michael Haenlein och Andreas Kaplan (2019). Enligt de, rötterna till AI kan spåras tillbaka till 1940-talet, med betydande bidrag från Isaac Asimovs science fiction-historia *Runaround* som introducerade Robotics Three Laws och Alan Turings arbete med The Bombe, en kod-brytande maskin som användes under andra världskriget. Turings publicering av *Computing Machinery and Intelligence* år 1950, där han föreslog Turing-testet som ett riktmärke för att testa maskinintelligens, ökade intresset för AI ytterligare. Termen *Artificial Intelligence* myntades officiellt 1956 under Dartmouth Summer Research Project on Artificial Intelligence (DSRPAI), som samlade framstående forskare, inklusive Marvin Minsky och John McCarthy och markerade början på AI-fältet som vi känner det idag. DSRPAI syftade till att skapa ett nytt forskningsområde fokuserat på att bygga maskiner som kan simulera mänsklig intelligens.

Med grundandet av AI vid Dartmouth-konferensen 1956 påstår Haenlein och Kaplan (2019) att AI-området hade en period på nästan två decennier präglad av anmärkningsvärda prestationer. Exempel inkluderar programmet General Problem Solver skapat av Herbert Simon, Cliff Shaw och Allen Newell, som automatiskt skulle kunna lösa några enkla problem. Utöver det datorprogrammet ELIZA som utvecklades av Joseph Weizenbaum vid MIT, som kunde simulera konversationer med människor och försöka passera Turingtestet. Denna prestation resulterade i betydande investeringar för AI-forskning och övertygelsen om att en dator med intelligens på mänsklig nivå kan skapas på kort tid.

Medan AI gjorde betydande framsteg framkom det under denna tid tvivel kring AI:s förmåga. Den amerikanska kongressen började starkt kritisera de höga utgifterna för AI-forskning år 1973. I samma år publicerade den brittiske matematikern James Lighthill en rapport godkänd av det brittiska forskningsrådet för vetenskap där han ifrågasatte optimistisk utsikt från AI-forskare. Lighthill uppgav att maskiner skulle endast nå nivån för en *erfaren amatör* i spel som schack och att sunt förnuft-resonemang alltid skulle vara bortom deras förmåga. Detta ledde till att finansieringen för AI-forskning minskades markant, vilket i sin tur ledde till en period av *AI vinter* där framsteg var obetydliga (Haenlein och Kaplan, 2019).

Haenlein och Kaplan (2019) tar även upp hur den unika metoden, som tidiga system som ELIZA och General Problem Solver, syftade till att efterlikna mänsklig intelligens, var en

orsak till den initiala bristen på utveckling inom AI-området och det faktum att verkligheten dramatiskt gick tillbaka i jämförelse med förutsägelser. På grund av gränserna för de tidigare nämnda expertsystem, som var beroende av regelbaserade metoder och saknade kapacitet att förstå externa data, lära sig och anpassa sig flexibelt, stoppades framsteg inom området AI. Skapandet av artificiella neurala nätverk och hebbian lärande av den kanadensiske psykologen Donald Hebb, som imiterade neuronernas funktion i den mänskliga hjärnan, gav dock ramarna för vidare utveckling inom AI på 1940-talet.

2015 skedde det dock ett återupplivande på området med Googles AlphaGo, ett Deep Learning-baserat program som slog världsmästaren i det utmanande spelet Go. Trots bristen på framsteg i fältet år 1969 när datorer ansågs inte klara av bearbetningskraven av artificiella neurala nätverk. I den nuvarande eran av AI, när artificiella neurala nätverk och tekniker för djupinlärning används flitigt i appliceringar som bildidentifiering, taligenkänning och självkörande bilar, representerade detta ett enormt framsteg för AI (Haenlein och Kaplan, 2019). Detta stämmer med McCarthy (2007) om olika potentiella användningsområden för AI, som han menade skulle inkludera spel, taligenkänning, förståelse av naturligt språk, datorseende, expertsystem och heuristisk kategorisering.

När man betraktar utvecklingen av artificiell intelligens brukar begreppet vanligtvis diskuteras i termer av generationer. Enligt Zhang, Zhu och Su (2023) inkluderar den *första generationen* av AI, även kallad *symbolisk AI*, de tidigare tillvägagångssätten inom AI-forskning som starkt förlitade sig på formella regler samt logiska system för att representera och manipulera kunskap. Dessa system utformades för att efterlikna mänskligt tänkande genom att bryta ner komplexa problem i mindre, mer hanterbara delar och sedan använda logiska slutsatser för att härleda lösningar från dessa delar. Fokus låg på att skapa explicita kunskapsrepresentationer och bygga system som kunde manipulera dessa för att lösa problem. Detta tillvägagångssätt hade dock begränsningar, såsom svårigheter att hantera osäkerhet och sammanhang. En ersättning av nyare tillvägagångssätt inom AI förekom gradvis efter, som neurala nätverk och djupinlärning.

Zhang, Zhu och Su (2023) tar även upp den *andra generationen* av AI. Denna generation, även känd som djupinlärning, bygger på idén om *connectionism*, vilket hävdar att sensoriska responser inte lagras i minnet utan skapar stimulus-respons-kanaler i neurala nätverk som leder till intelligenta beteenden. I grunden är detta en modell för *artificiella neurala nätverk* (ANN), där den ursprungliga perceptronen av Rosenblatt från 1958 var en prototyp för ANN, baserad på idén om connectionism. Kritiken av perceptronen var att den bara kunde lösa linjärt separabla problem, vilket ledde till att connectionism föll till en lågpunkt under mer än ett decennium. Linjärt separabla problem innebär att de två typerna av data i problemet lätt kan särskiljas från varandra med hjälp av en rak linje. Om man inte kan göra det kallas det ett icke-linjärt problem, därav kan man behöva mer komplexa tekniker för att lösa det. Efter att connectionism föll till en lågpunkt, trots kontinuerliga ansträngningar från forskare hade stora framsteg gjorts inom både neurala nätverksmodeller och inlärningsalgoritmer under de senaste 30 åren enligt författarna, vilket gradvis formade mogna teorier och tekniker inom djupinlärning.

Andra viktiga framsteg under denna period inkluderade bättre förlustfunktioner, algoritmutveckling, nya former av nätverk och djupa tro-nätverk. Dessa metoder tillsammans öppnade en ny era av andra generationens AI baserad på djupinlärning (Zhang, Zhu och Su, 2023).

Enligt Zhang, Zhu och Su (2023) producerade den första generationen av AI intelligenta maskiner med hjälp av kunskap, algoritm och datorkraft. Den andra generationen imiterade mänskligt beteende med hjälp av data, algoritmer och datorkraft. Varje generation hade dock sina egna begränsningar och representerade inte korrekt mänskligt beteende. För att överkomma dessa begränsningar skapas för närvarande en tredje generation av AI som kombinerar de bästa egenskaperna från den första och andra generationen. Huvudtanken är att integrera kunskapsdrivna och datadrivna metoder för att skapa en mer kraftfull AI som samtidigt använder kunskap, data, algoritmer och datorkraft. Det slutliga målet med tredje generationens AI enligt författarna är att bygga robust, säker, pålitlig och utvecklingsbar AI-teknik som kan återspegla intelligenta mänskliga beteenden.

Studien av Zhang och Lu (2021) ger en omfattande översikt över drivfaktorer och tekniker som möjliggör utvecklingen av AI. Författarna diskuterar ett antal viktiga element, såsom förbättringar av datorkraft, *big data* eller tillgängligheten till enorma mängder data, tekniska framsteg inom algoritmer och metoder samt naturlig språkbehandling (NLP).

De menar att en stor mängd data, eller *big data*, är nödvändig för att AI ska kunna upptäcka och effektivt analysera data. Den ökande användningen av Internet of Things har bidragit till spridningen av big data och med en betydande årlig tillväxttakt växer mängden data som produceras snabbt. Dessutom blir den data som skapas mer sofistikerad och flerdimensionell, vilket utökar dess räckvidd och kvalificerar den för att bidra till utvecklingen av AI. Big data är därför en förutsättning för utvecklingen av AI och är avgörande för att höja effektiviteten och precisionen i AI-systemen (Zhang och Lu, 2021).

Författarna tar även upp framstegen inom algoritmer. De menar att traditionella mönsterigenkänningsmetoder förlitade sig på att sammanfatta lagar och metoder, som ofta var begränsade i noggrannhet. Forskare har inspirerats av hur spädbarn lär sig att känna igen föremål utan explicit instruktion och har utvecklat metoder för maskininlärning som gör att datorer kan lära sig av stora mängder data. Till exempel, genom att mata in många foton av hundar i en dator, kan datorn lära sig hundarnas egenskaper och exakt känna igen dem i andra foton med hjälp av neurala nätverk och andra maskininlärningsmodeller.
Även maskininlärningsalgoritmer gör det möjligt för datorer att automatiskt analysera stordata, fatta beslut och förutsäga händelser i verkliga världen. AI-algoritmer har framgångsrikt tillämpats inom olika områden, inklusive taligenkänning, sökmotorer, semantisk analys och rekommendationssystem, med anmärkningsvärda framsteg inom vart och ett av dessa områden på grund av kraften från AI-algoritmer (Zhang och Lu, 2021).

I deras studie, Zhang och Lu (2021) diskuterar de även vikten av naturlig språkbehandling (NLP) som ett nyckelelement i utvecklingen av AI. Även i relevans till vår studie borde vi ta upp NLP, som beskrivs enligt författarna som datorers förmågan att upptäcka och förstå mänskligt skriftspråk. Eftersom det yttersta målet är att göra det möjligt för datorer att förstå naturligt språk, är NLP ett tvärvetenskapligt område som kombinerar datavetenskap med mänsklig språkvetenskap. Grammatisk och semantisk analys, text extrahering och utvinning, informationssökning och hämtning, maskinöversättning, frågebesvarande systemet och dialogsystemet är de sju underliggande områdena till NLP. Varför NLP är ett viktigt område inom AI beror på att den har potential att förändra interaktionen mellan människa och maskin. NLP har många viktiga appliceringar inom olika branscher och områden, några exempel på appliceringsområden är följande;

Virtuella assistenter som Siri, Alexa och Google Assistant där NLP används för att förstå och svara på röstkommandon (Zhang och Lu, 2021).

Sentimentanalys, när man vill analysera textdata och bestämma sentimentet eller känslorna bakom texten. Detta kan användas till exempel i övervakning av sociala medier eller analys av kundfeedback (Zhang och Lu, 2021).

Maskinöversättning, som innebär att automatiskt översätta text från ett språk till ett annat, används ofta i applikationer som översättning av webbinnehåll (Zhang och Lu, 2021).

Chatbots, där NLP används för att simulera mänskliga samtal och ge kundservice, support och information (Zhang och Lu, 2021).

Taligenkänning när man vill konvertera talat språk till text. Denna teknik är kopplad till den första nämnda punkten där tekniken används i applikationer som röst-assistenter och vägbeskrivningsprogram som Google Maps (Zhang och Lu, 2021).

Utöver appliceringsområdena för NLP finns flera appliceringsområden för AI som helhet i den moderna världen idag. Författarna i samma studie tar upp olika tillämpningsscenarier för AI i olika branscher, inklusive fordonsindustrin, finansmarknaderna, hälsoindustrin, detaljhandeln, medieindustrin och smarta betalningssystem. Inom dessa områden används AI för (Zhang och Lu, 2021):

- att utveckla teknik för autonom körning,
- riskkontroll, marknadsprognoser och kreditvärdering,
- att till exempel upptäcka cancer och utveckla nya läkemedel,
- obemannad detaljhandel, produktrekommendationer och lagerhantering,
- att generera innehåll, opinionsanalys och varumärkesreklam,
- röst- och ansiktsigenkänning vid betalningar som kräver det.

Om man tittar på Harman (2012) lyfter han appliceringsområdena för AI inom mjukvaruutveckling. Han menar att AI har potential att revolutionera mjukvaruutveckling genom att automatisera många av de involverade uppgifterna och göra processen snabbare, effektivare och mer exakt. Han nämner olika tillämpningsscenarier såsom automatiserad mjukvarutestning, mjukvaru debugging, intelligent kodkomplettering, förutsägande underhåll och även mjukvaruoptimering.

Tabell 3. Appliceringsområden för AI inom programutveckling enligt Harman

Appliceringsområden	*Hur AI kan bidra*
Automatiserad mjukvarutestning	AI kan prioritera tester, producera testfall automatiskt och hitta mjukvarubrister
Mjukvaru debugging	AI kan analysera kod med hjälp av algoritmer som kan hitta möjliga brister och rekommendera åtgärder
Intelligent kodkomplettering	Applikationer med programmeringsmiljöer (IDE) kan ge förslag på kodavsnitt och funktioner beroende på sammanhanget för den kod som för närvarande skrivs

Predictive underhåll	AI kan undersöka kod och identifiera möjliga fel innan de eskalerar till större problem
Mjukvaruoptimering	AI-algoritmer kan analysera kod och ge optimeringsförslag för att öka prestanda eller spara resurser

Harman (2012) påstår till slut att det finns svårigheter med att utvärdera och förklara resultaten av AI-algoritmer, såväl som nödvändigheten av data av hög kvalitet, när man använder AI för att skapa programvara. Trots dessa svårigheter menar han dock att det finns betydande potentiella fördelar med att använda AI i mjukvaruutveckling med hänsyn till att detta område aktivt forskas om och utvecklas.

Shafiq, et al. (2021) diskuterar appliceringen av AI inom mjukvaruutveckling. Deras analys visar att AI har förmågan att dramatiskt förbättra mjukvaruutvecklingsprocessen genom att automatisera tidskrävande och repetitiva operationer, minska sannolikheten för misstag och höja nivån på mjukvaruutvecklingens allmänna kvalitet.

Tabell 4. Appliceringsområden för AI inom SDLC enligt Shafiq et al.

Appliceringsområde	*Hur AI kan bidra*
Kravanalys	AI kan översätta naturliga språkbehov till formella specifikationer
Design	AI kan generera designalternativ för mjukvarusystem baserade på funktionella och ickefunktionella krav
Kodning	AI kan automatisera kodgenerering baserat på mjukvarukrav och design
Testning	AI kan automatisera mjukvarutestning, inklusive generering och exekvering av testfall
Underhåll	AI kan analysera loggar och användarfeedback för att identifiera och diagnostisera programvaruproblem

Författarna (Shafiq, et al., 2021) påpekar dock att det fortfarande finns en hel del frågor som måste lösas, såsom kravet på data av hög kvalitet, AI-modellers tolkningsbarhet samt etiska frågor.

En vanligt förekommande praxis inom programmering och mjukvaruutveckling idag där man utnyttjar AI är kodgenerering med hjälp av olika AI-verktyg, det är ett utmärkt exempel på tillämpningen av AI inom dessa områden. Ett framstående exempel på ett sådant verktyg är *GitHub Copilot*. Enligt GitHub (2023) fungerar GitHub Copilot som en virtuell par programmerare genom att ge förslag i form av autokomplettering medan användaren kodar. Användaren kan trigga Copilots förslag genom att skriva den kod som behövs eller genom att skriva en kommentar i naturligt språk som beskriver den funktionalitet som eftersträvas. Genom att analysera sammanhanget för filen som redigeras, tillsammans med relaterade filer, genererar GitHub Copilot relevanta förslag direkt i textredigeraren. Verktyget drivs av

OpenAI Codex, ett AI-system som utvecklats av OpenAI och har mycket stor betydelse idag när det kommer till framgången av AI. Verktyget har även stor relevans till ämnet som undersöks i denna studie.

Med träningsdata från alla språk som finns i offentliga arkiv, kan kvaliteten på GitHub Copilots förslag variera beroende på mängden och mångfalden av data för varje språk. Till exempel, med tanke på dess omfattande representation i offentliga arkiv, är *JavaScript* bland de bäst stödda språken på GitHub Copilot. Samtidigt kan språk med färre exempel eller mindre robust träningsdata ge färre eller mindre precisa förslag (GitHub, 2023).

Framtiden av AI. Enligt Zhang och Lu (2021) finns det tre hörnstenar som utvecklingen av AI grundar sig på. Dessa centrala synpunkter är *symbolism*, *Connectionism* och *behaviorism*. De fungerar som den teoretiska grunden för AI utveckling.

Symbolism betonar användningen av symboler och logik för att representera och resonera kring kunskap. *Connectionism* fokuserar på att imitera strukturen och arbetssättet hos den mänskliga hjärnan för att simulera mänsklig intelligens. *Behaviorism* betonar vikten av perception och handling i intelligent beteende och förespråkar en gradvis utveckling av AI. Dessa tre synpunkter har haft men även att fortsätta ha en betydande inverkan på utvecklingen av artificiell intelligens. Författarna föreslår att nästa genombrott inom AI inte bara kan ge datorer mer logiska resonemangsförmåga utan också känslomässiga kapaciteter. Detta menar de kan potentiellt leda till att intelligensen hos maskiner överträffar människors (Zhang och Lu, 2021).

Ett annat perspektiv på framtiden av AI är enligt Haenlein och Kaplan (2019) där de diskuterar behovet av reglering i framtiden med avseende på AI-system och deras påverkan på samhället ur tre olika perspektiv. Mikroperspektivet belyser den potentiella snedvridningen i AI-system, på grund av indata som används för att träna dem. Författarna föreslår utveckling av allmänt accepterade krav för träning och testning av AI-algoritmer för att undvika sådana fel. Meso-perspektivet fokuserar på AI:s inverkan på sysselsättningen, där ökningen av AI kan leda till att tjänstemannajobb går förlorade och föreslår att reglering kan vara ett sätt att undvika detta. Makroperspektivet diskuterar det potentiella missbruket av AI av stater, såsom det sociala kreditsystemet i Kina och betonar behovet av reglering för att säkerställa att AI används på ett demokratiskt och fredligt sätt.

3.3.1 Språkmodeller

Språkmodeller är statistiska modeller som tillämpas på naturlig språkbehandling för att beräkna sannolikheten för en ordsekvens. De är en avgörande komponent i många textbehandlingsprogram, såsom maskinöversättning, textkategorisering och informationssökning. En översikt över språkmodeller, deras bakgrund och deras många användningsområden ges av Hiemstra (2009).

I kapitlet *language models* förklarar Hiemstra (2009) den grundläggande idén med språkmodellering, som är att beräkna sannolikheten för en uppsättning ord givet ett visst sammanhang. Detta åstadkoms vanligtvis med n-gram-modeller, som fungerar under antagandet att ett ords sannolikhet helt enkelt beror på dess närhet till de föregående orden i sekvensen. N-Gram-modeller är en vanlig typ av språkmodell som använder sig av ordpar eller ordtripler för att uppskatta sannolikheter baserat på hur ofta de kombinationer av ord förekom i träningsdatan.

Språkmodeller sträcker sig tillbaka till början av 1900-talet när Andrei Markov använde sig av modeller för att beskriva bokstavsföljder i rysk litteratur. Begreppet *språkmodeller* myntades på 1980-talet i samband med utvecklingen av automatiserade system för taligenkänning. Sedan dess har språkmodeller använts flitigt inom flera fält, såsom maskininlärning, informationsåtervinning och behandling av naturligt språk. I slutet av 1990-talet blev de särskilt populära inom informationsåtervinning. En forskargrupp publicerade år 2003 en rapport där de poängterade det ömsesidiga beroendet mellan informationsåtervinning och språkmodellering, vilket visar på fortsatt betydelse av språkmodeller inom dessa områden (Hiemstra, 2009).

Språkmodeller kan även användas för en mängd olika NLP uppgifter, såsom tidigare nämnt, textklassificering, informationsåtervinning och maskinöversättning. En språkmodell kan till exempel användas för att klassificera dokument i fördefinierade kategorier som skräppost eller inte skräppost. I informationsåtervinning används språkmodeller för att rangordna dokument baserat på deras relevans för en viss fråga. Inom maskinöversättning används språkmodeller för att generera översättningar från ett språk till ett annat. I relation till NLP och AI spelar språkmodeller en viktig roll, vilket möjliggör för maskiner att förstå och generera mänskligt språk. Med de fortsatta framstegen inom AI och djupinlärning förväntas utvecklingen av mer avancerade språkmodeller att öka, vilket leder till ännu kraftfullare och mer precisa system för natural language processing (Hiemstra, 2009).

3.3.2 Chatbots

Enligt Adamopoulou och Moussiades (2020) är chatbot ett typiskt exempel på ett AI-system. Ett av de mest grundläggande och utbredda exemplen på intelligent människa-datorinteraktion. En chatbot är ett datorprogram som svarar som en smart enhet när man konverserar med den genom text eller röst och förstår ett eller flera mänskliga språk genom naturligt språkbehandling (NLP).

Historien om chatbots började 1950 när Alan Turing föreslog Turingtestet. Den första kända chatbot var Eliza, som utvecklades 1966 och användes som psykoterapeut. År 1972 utvecklades PARRY, som var en förbättring jämfört med Eliza. 1995 utvecklades ALICE, som vann Loebner-priset för mest mänskliga dator i år 2000, 2001 och 2004. Efter år 2000 utvecklades flera chatbots och personliga virtuella assistenter som Apple Siri, Microsoft Cortana, Amazon Alexa, Google Assistant och IBM Watson. Sedan 2016 har intresset för chatbots ökat kraftigt. Många chatbots har utvecklats för industriella lösningar medan andra är relevanta för forskning och dess tillämpningar (Adamopoulou och Moussiades, 2020).

Författarna (Adamopoulou och Moussiades, 2020) listar några grundläggande koncept i kapitlet om chatbot-teknik, dessa koncept är följande nedan.

Tabell 5. Grundläggande koncept inom chatbot-teknik enligt Adamopoulou och Moussiades

Koncept	***Förklaring***
Pattern Matching (Mönsterigenkänning)	En teknik där en chatbot använder fördefinierade svar baserat på input den får.
Artificial Intelligence Markup Language (AIML)	Ett markup språk som tillåter chatbots att svara på användarinput. Det består av grundläggande dialog enheter som kallas kategorier, som inkluderar mönster av användarinmatning och

	motsvarande chatbot svar.
Latent Semantic Analysis (LSA)	En teknik som används i samband med AIML för att chatbots ska kunna svara på ett bredare urval av frågor. LSA används för att identifiera likheter mellan ord som vektorrepresentationer.
Chatscript	Ett expertsystem som inkluderar ett skriptspråk med öppen källkod och motorn som kör det. Systemet använder regler kopplade till ämnen för att avgöra det bästa svaret på en användares fråga.
RiveScript	Ett linje-baserat skriptspråk med vanlig text för utveckling av chatbots. Den är öppen källkod och kompatibel med flera programmeringsspråk.
Natural Language Processing (NLP)	En gren av artificiell intelligens som fokuserar på att göra det möjligt för datorer att förstå och manipulera mänskligt språk.
Natural Language Understanding (NLU)	En teknik som används för att extrahera sammanhang och mening från användarens naturliga språk. Den hjälper chatbots att identifiera användarens avsikt.
Entity	Ett verktyg som används för att extrahera parametervärden från naturligt språk inmatningar.
Contexts	En sträng som lagrar sammanhanget för objektet som en användare refererar till. Dessa sammanhang hjälper chatbots att ge lämpliga svar baserat på föregående konversation.

Författarna (Adamopoulou och Moussiades, 2020) diskuterar också frågan om att designa och utveckla chatbots. De tror att denna process kräver en mängd olika strategier. De hävdar att för att skapa en chatbot är det viktigt att förstå de tjänster den ska tillhandahålla och kategorierna som den ska tillhöra. Detta hjälper konsumenter att förstå vad de kan förvänta sig från chatboten såväl som utvecklare att välja rätt plattformar och algoritmer att bygga på.

Exakt kunskapsrepresentation, en metod för att generera svar och ett bibliotek med neutrala svar som kan användas om användarens fråga inte förstås är alla nödvändiga komponenter i en chatbots arkitektur. Användarens begäran, komponenten för språkförståelse som tolkar användarens information och avsikt, informationsextraktion, åtgärds exekvering och svar genererings komponenten som genererar lämpligt svar baserat på användarens sammanhang och avsikt utgör den typiska chatbot-arkitekturen. Med AI och maskininlärning kan chatbots bli sofistikerade till den nivån att användarna inte längre kan skilja dem från verkliga agenter (Adamopoulou och Moussiades, 2020).

3.3.3 ChatGPT

ChatGPT skapades av OpenAI, ett forskningslaboratorium för artificiell intelligens. Enligt OpenAI (2023) är utvecklingen av deras omfattande språkmodeller, inklusive de som driver ChatGPT, bygger på tre huvudsakliga informationskällor som är allmänt tillgänglig information på internet, licensierad information som erhållits från tredjepartskällor och information från användare eller mänskliga tränare.

Enligt OpenAI (2023) är ChatGPT en AI-baserad tjänst som nås online. Tjänsten skapades för att förstå användarfrågor och instruktioner genom att läsa mycket material och ta reda på hur ord används i olika situationer. Som svar på en användarfråga använder den vad den har lärt sig för att förutse följande ord och den gör detta för varje ord efter det. Enligt ChatGPT:s skapare är detta jämförbart med de automatiska kompletteringsfunktionerna som erbjuds av sökmotorer och mobila enheter. ChatGpt som en tjänst och dess möjligheter sträcker sig till ett brett utbud av applikationer, från innehållsskapande och hantering av sociala medier till kundservice och språköversättning. Skaparen menar alltså att i huvudsak kan ChatGPT användas i alla sammanhang där en användare behöver hjälp med naturliga språk uppgifter.

För att lära sig detta, enligt OpenAI (2023) förbrukar modellen mycket litteratur samtidigt som den tränas för att lära sig detta. Till exempel, efter att ha fått en mening eller fras, kan modellen lära sig vilka ord som passar bäst med den. Programmet kan mer korrekt identifiera följande ord efter att ha läst och lärt sig från flera jämförbara fraser. Modellen kan svara på en användarfråga på en mängd olika sätt eftersom det finns flera fraser som kan vara lämpliga i en given omständighet. Detta beror på att modellens reaktion är tänkt att vara något oförutsägbar.

OpenAI (2023) förklarar att maskininlärningsmodeller är uppbyggda av många siffror och koden som förstår och använder dessa siffror. Modeller ändrar sina siffror för att återspegla vad de har lärt sig snarare än att lagra kopior av data de använder för att lära sig av. Till exempel ändras modellens interna siffror något när den lär sig att förutsäga rätt ord för att avsluta en fras, vilket förbättrar dess förmåga att göra det i framtiden.

3.4 AI:s roll i modern programmering och mjukvaruutveckling

Utifrån de tidigare ämnen som har behandlats hittills kan man med säkerhet påstå att AI och mjukvaruutveckling korsar varandra på ett eller annat sätt. I publikationen av Harman (2012) undersökte författaren olika tillämpningar av AI inom mjukvaruutveckling. Han påstår att AI har potential att revolutionera mjukvaruutvecklingen genom att automatisera många av de involverade uppgifterna, vilket kan göra processen snabbare, effektivare och mer exakt. Han nämner flera exempel på användningsområden såsom automatiserad mjukvarutestning, mjukvaru debugging, intelligent kodkomplettering, förutsägande underhåll och mjukvaruoptimering. Dessa användningsområden har vi redan tagit upp tidigare, som visas i tabell 4.

Harman drar slutsatsen att det ökande intresset för *Search Based Software Engineering* visar att AI-tekniker är av stort intresse. Mjukvaruutveckling går från att vara hantverksmässigt till att bli en ingenjörsvetenskap med fokus på storskaliga, komplexa och interaktiva system. Utmaningarna i dagens mjukvaruutveckling, med föränderliga appliceringsområden, kräver ingenjörsmässig optimering. Trots svårigheter med att utvärdera och förklara AI resultat samt behovet av högkvalitativ data, betonar Harman (2012) de betydande fördelarna med att

använda AI inom mjukvaruutveckling med tanke på det pågående forsknings- och utvecklingsarbetet inom området.

Shafiq, et al. (2021) undersöker sambandet mellan faserna i livscykeln för systemutveckling (SDLC) och olika typer, verktyg och tekniker av maskininlärning. Tidigare i tabell 5 nämndes översiktligt appliceringsområden för ML inom systemutveckling.

Författarna förklarar att maskininlärning har genomgått betydande utveckling de senaste åren och används nu globalt. ML som en del av artificiell intelligens betraktas som avgörande vid utveckling av programvarusystem inom områden som tal- eller bildigenkänning eller fordonsindustrin. ML-tekniker har även tillämpats för att hantera olika problem och aktiviteter inom mjukvaruutveckling. Vanligtvis används ML främst inom *defect prediction*, *effort estimation* samt identifiering av mönster och likheter i källkoden.

ML-teknikerna är avsedda att lösa problem som ibland kan vara svåra att analysera för människor och därmed orsaka felaktig information. Dessa problem kan kategoriseras generellt in i övervakad, oövervakad och förstärkande inlärning. De flesta tillämpningar av ML handlar om problem som kan betraktas som övervakad inlärning, vilket innebär att man lär sig från funktioner tillsammans med deras kända gruppbeteckning och sedan förutsäger gruppbeteckning för nya, osedda funktioner. För att hantera olika typer av problem används traditionella ML-tekniker och neurala nätverksbaserade tekniker såsom djupinlärning.

Följande tabell visar i viss omfattning mer detaljerat hur ML appliceras i SDLC-faserna enligt Shafiq, et al. (2021)

Tabell 6. Appliceringsområden för ML inom SDLC enligt Shafiq et al.

SDLC faser	***Hur ML kan bidra***
Mjukvarukrav	ML används inom denna fasen genom modellering, analys, urval, prioritering, klassificering och spårbarhet. ML-tekniker har tillämpats här för att ta itu med frågor som oklarhet, ofullständighet, korrekthet och länkningskrav till kod/artefakter. **Användningsexempel:** Generering av prioriteringsgrader, klassificering av krav med hjälp av neurala nätverk och utnyttjande av djupinlärning för spårbarhet.
Mjukvaruarkitektur och Design	ML har tillämpats inom denna fasen genom förslag modeller, identifiering av design svagheter, modifiering av objektorienterade strukturer, språkmodellering av programvara, igenkänning av designmönster och *effort estimation*. **Användningsexempel:** Språkmodellering av mjukvaror baserat på djupinlärning. Djup UI för halvautomatiska designuppgifter och artificiella neurala nätverk-baserade *effort* estimationer.
Mjukvaru Implementering	ML utnyttjas inom denna fasen genom att den

	hjälper till med uppgifter som att hitta redundant kod, lokalisering av kod, omstrukturera kod, upptäcka kod och design svagheter, automatisera kod granskningar och identifiera kod/program likheter. **Användningsexempel:** Förutsäga kodblock i tutorials, adressering av obalanserade datauppsättningar, analysera och jämföra program för att identifiera instanser av piratkopiering av programvara.
Mjukvarukvalitet och Analys	ML bidrar inom denna fasen genom att göra förutsägelser av fel och buggar, generera, göra urval av, prioritera och klassificera testfall, upptäcka, analysera och utvärdera sårbarheter och avvikelser med mjukvaror och tekniker samt verifiera, validera och uppskatta test-insatser.
Mjukvaru Underhåll	Inom denna fasen är ML minst utforskad. Studier fokuserar på kostnadsuppskattning, förutsägelse av modularitet av programvaror, upptäcka programvaruåldring och för att uppskatta underhåll.

Författarna menar dock till slut att, även om idén om full automatisering med hjälp av maskininlärning är tilltalande, kan det leda till felaktiga system. Därför rekommenderas det att människor involveras i avgörande beslut. Det är också viktigt att ha kostnads-nyttoanalyser av ML-tekniker för att de ska vara praktiska för industrianvändning.

Om vi tar en titt närmare på AI som ChatGpt inom mjukvaruutveckling. Aljanabi, et al. (2023) menar att ChatGPT-3 har potential att revolutionera interaktionen mellan människa och teknik. ChatGPT-3 med sin omfattande utbildning på enorma data påstår författarna kan generera anmärkningsvärt korrekt människoliknande text. Dessa applikationer inkluderar förbättrad bearbetning och förståelse av naturligt språk, driva chatbots, virtuella assistenter och andra gränssnitt för konversationer. De menar även att när interaktioner via röst och text blir vanligare spelar ChatGPT-3 en avgörande roll för att förändra dessa interaktioner.

Aljanabi et al. (2023) diskuterar olika scenarier som relaterar till möjligheterna med AI:n ChatGpt. Vår specifika fokus ligger dock inom mjukvaruutveckling i samband med den. I detta avseende erkänner författarna flera fördelar med att använda ChatGPT-3 för kodning. De uppskattar dess förmåga att förstå inmatningar av naturligt språk, vilket gör att utvecklare kan interagera med den som om de ställer frågor till en person. Detta intuitiva och användarvänliga tillvägagångssätt enligt de gör kodningsprocessen mer tillgänglig. Dessutom menar de att ChatGPT-3:s förmåga att tillhandahålla kontextuellt relevant information är tydligt belyst. Istället för att bara presentera en lista med länkar eller dokumentation, kan den förstå avsikten bakom frågor och erbjuda information direkt relaterad till kodavsnittet eller kommandot som används. Baserat på det menar författarna att detta sparar tid för utvecklare och ökar effektiviteten. En annan styrka som nämns är ChatGPT-3:s möjligheter av att

generera kod, som är värdefulla för att generera *boilerplate*-kod och automatisera repetitiva uppgifter i stora projekt.

Författarna erkänner dock också begränsningarna med ChatGpt. En primär begränsning påstår de är kostnaden och tillgängligheten för ChatGPT-3. Under tidpunkten av skrivandet menar författarna att ChatGpt var begränsad till en utvald grupp av utvecklare och forskare, vilket begränsar dess potentiella användarbas. En annan nackdel som de nämner är ChatGPT-3:s oförmåga att helt förstå nyanserna i programmeringsspråk, vilket leder till potentiella felaktigheter eller begränsad användbarhet i informationen som den tillhandahåller. Dessutom lider den av att hantera vissa typer av frågor som felsökning eller prestandaoptimering (Aljanabi et al. 2023).

I forskningsartikeln av Biswas (2023) diskuterar författaren möjligheterna och potentiella tillämpningar av ChatGPT inom datorprogrammering. Forskningsartikeln tar upp datorprogrammering som ett komplext område och belyser de utmaningar som utvecklare står inför när det gäller att hänga med i den senaste tekniken och lösa tekniska problem. Den föreslår ChatGPT som en språkmodell som kan hjälpa utvecklare i olika programmeringsuppgifter. Författaren påstår att ChatGpt utmärker sig i uppgifter som kodkomplettering, korrigering, generering av dokument, utveckling av chatbotar, text-till-kod konvertering och svar av tekniska frågor. Biswas menar att genom integreringen av ChatGPT i programmeringsmiljöer kan utvecklare förbättra sin produktivitet, påskynda kodningsprocessen och få hjälp med ett brett utbud av programmeringsuppgifter.

När det gäller kodkomplettering och korrigering bevisar ChatGPT enligt författaren sitt värde genom att förutsäga och föreslå kodavsnitt, automatiskt fixa syntaxfel och vanliga misstag, ge optimeringsförslag och generera saknad kod baserat på projekt kontext. Syftet med att utnyttja ChatGPT:s möjligheter säger författaren är för att utvecklare ska kunna spara tid, förbättra kod noggrannheten och fokusera på kritiska aspekter av sina projekt istället. Dessutom spelar ChatGPT en viktig roll för att förutsäga och föreslå kodbitar. Den kan analysera kodkontext, införliva användarens kodnings stil, bygga på sin kunskapsbas om programmeringsspråk och bibliotek, samt beakta projektberoenden och begränsningar för att ge korrekta och sammanhangs medvetna förslag. Detta menar författaren underlättar snabbare kodning, minimerar fel och förbättrar kod kvaliteten (Biswas, 2023).

Biswas (2023) säger även att ChatGPT:s möjligheter sträcker sig till att ge förslag på kodoptimering och omstrukturering. Den kan analysera kod för prestanda svagheter, identifiera redundant kod, rekommendera andra strukturer för förbättrad läsbarhet, underhållbarhet och föreslå designmönster samt bästa praxis.

Slutsatsen som författaren (Biswas, 2023) till slut konstaterar är att ChatGPT:s integration i programmeringsmiljöer förbättrar utvecklarnas produktivitet och effektivitet avsevärt. Dess möjligheter i de nämnda uppgifterna tidigare gör den till ett nödvändigt verktyg för programmerare som söker hjälp i olika aspekter av sitt arbete. Genom att utnyttja ChatGPT:s kraft menar författaren att utvecklare kan effektivisera sina kodningsprocesser, förbättra kod kvaliteten och fokusera på kärn elementen i sina projekt.

3.5 AI Problematik

Coeckelbergh (2019) har tagit upp etiska problem som kan uppstå runt användningen av AI. I sin artikel tar Coeckelbergh upp sex stora etiska problem som man bör ta itu med för att säkra att vi använder AI på rätt sätt. Det första etiska dilemmat som Coeckelbergh tar upp är om

man respekterar användarnas privatliv. Detta är framförallt kopplat till AI som utnyttjar maskininlärning och samlar in information från de personer som använder den. Problematiken kring detta är att man inte kan säkerställa om användarna vet om att deras data samlas in. Datan samlas dessutom in i ett sammanhang och används sedan av AI:n i en massa olika sammanhang utan att personen som "lämnade in den" vet om vad den används till.

Det andra problemet som tas upp är att AI-systemen samlar in en stor del information från en stor mängd olika personer. Denna information riskerar att förloras och användas i ett dåligt syfte om någon tar sig in i dem på ett olagligt sätt, det vill säga att de blir hackade (Coeckelbergh, 2019).

Det tredje dilemmat som Coeckelbergh (2019) tar upp handlar om vem som har ansvar för det som kommer ut från AI:n. Eftersom AI:n inte kan hållas ansvarig för sina svar som den producerar bör någon annan göra det. Men att veta vem som ska hållas ansvarig är generellt väldigt svårt eftersom det är många olika människor och många maskiner utöver själva AI som har en hand i vad det är för svar som produceras.

Det fjärde dilemmat kommer i form av att nästan ingen vet exakt vad som händer inuti AI:n. Detta är framförallt ett problem eftersom man ofta inte kan få fram anledningen till varför AI:n har kommit fram till det svar som den har kommit fram till. Detta är något som en person som blir påverkad av svaret borde ha rätt att veta (Coeckelbergh, 2019).

Det femte dilemmat handlar om att det finns en risk att AI:n har fördomar eller är partisk. Coeckelbergh (2019) beskriver hur detta alltid har varit ett problem i vårt samhälle men att det är något som kan bli ännu värre om en AI som tar beslut kring saker skulle vara det.

Det sjätte och sista dilemmat handlar om att det finns en risk att AI börjar automatisera mycket till den nivån att arbetslöshet blir ett stort samhällsproblem. Det kan gå lika långt som att vi hade behövt förändra hur ekonomin och liknande fungerar för att detta inte ska bli ett enormt problem (Coeckelbergh, 2019).

3.5 AI reglering

EU-kommissionen presenterade den 21 april 2021 ett förslag till en förordning som kan komma att påverka AI framöver. Förordningen har som mål att på ett tydligt sätt reglera saker runt AI för att hela EU ska arbeta med samma regler när det kommer till området. Reglerna ska ha en utgångspunkt i den etiska problematik som finns runt AI och hoppas lösa mycket av den problematiken. Förordningen tar upp en mängd olika delar som de anser behöva regleras runt AI. Där inkluderas bland annat vilka typer av AI-system som är tillåtna, vilka som inte är tillåtna, reglering kring transparens mot användaren, regler som ska underlätta innovation och regler kring hur insamlad information ska skyddas. Förordningen är dock endast ett förslag och håller fortfarande på att förhandlas kring i EU. Om man skulle komma fram till en förordning som sätts i verket som beskriver (källan) att den kan komma att kompletteras med svenska lagar också (2020/21:FPM 109).

4 Resultat

Här kommer resultatet av de intervjuer som vi genomfört presenteras.

4.1 Anonymitets tabell

En tabell har skapats som inkluderar unika ID:n för de respondenter som vi har samlat in information från. Dessa ID:n är framtagna för att säkerställa att personerna kan hållas anonyma samtidigt som vi kan föra den dialog som kommer krävas runt den insamlade informationen på ett smidigt och lätt antagligt sätt för läsaren.

Tabell 7. Alias för respondenterna

ID	***Roll***	***Intervjulängd* (t,m)**
Adam	Testare	1,30
Peter	Utvecklare	1
David	Utvecklare	1,30
Liam	Utvecklare	1
Simon	Testare	1
Hugo	Utvecklare	1,30

4.2 Adam

4.2.1 Adams bakgrund och erfarenhet

Adam, med bakgrund inom el-automation från gymnasiet och erfarenhet som elmontör inom industriell ventilation, valde att studera datateknik på högskolan. Efter att ha känt sig uttråkad av studierna under andra året, anslöt han sig som testare till ett svenskt företags utvecklingsteam som förvaltar och utvecklar ett bokföringsprogram. Med över två års erfarenhet trivs Adam i den praktiska arbetsmiljön och har en fri roll som testare, aktiv från planering till testning. Hans goda relation med utvecklarna och förmåga att programmera i Java har stärkt hans bidrag. Adam uppskattar att identifiera svåra fel och ge värdefull feedback till teamet, vilket ger honom uppfyllelse i hans roll som testare.

4.2.2 Adams upplevelse av AI

Adam har på sin arbetsplats stött på användning av AI under deras utvecklingsprocess. Detta har förekommit i form av ChatGPT som de använder för att anpassa de olika texter som man kan stöta på i deras produkt. Adam beskriver hur det förekommer att de använder ChatGPT för att kolla efter stav och grammatikfel som kan förekomma i texten. Den kan även användas för att ge förslag på hur man kan få texten att framställas i olika typer av tonlägen. Dessa två delar beskriver Adam att han tycker att ChatGPT är ett väldigt bra verktyg för. I övrigt har han även testat att använda ChatGPT för att hjälpa honom bli en bättre testare. Detta har gjorts genom att bara chatta med den och genom att ställa frågor gällande hur han kan göra vissa delar eller för att be om råd gällande hur man kan vara en bättre testare. Dock har han aldrig använt den för att den ska skriva någon form av kod eller testfall eftersom han inte känt ett behov av det. Adam beskriver även att han inte har stött på någon annan form av AI som stödjer utvecklingsprocessen under tiden som han har jobbat som testare utan ChatGPT är

hans första erfarenhet av det. Dock vet han om att de har en del AI lösningar i sina system som utför vissa uppgifter. Men det är inget som han personligen varit med och utvecklat eller själv har någon djupare kunskap gällande. Adam beskriver att han inte kan redogöra för om det finns några planer för att använda AI i framtiden redan nu men menar att de självklart funderat över det, något han tror att alla inom tech branschen gjort i någon mån.

Generellt sett tycker Adam att det är kul att AI har blivit så pass stort som det har sedan dess att ChatGPT kom ut. Dock säger han att han självklart funderat över vilka konsekvenserna av utvecklingen kan bli och det har skapat en form av skräckblandad förtjusning. Dock menar Adam att han välkomnar utvecklingen med öppna armar och känner att det är viktigt för honom. Han beskriver att ingen kan helt säkert veta exakt vad det kommer ha för konsekvenser för IT branschen men att det med största sannolikhet kommer att ha någon form av påverkan på densamma. Av just den anledningen känner Adam att det är viktigt för honom att han försöker förstå sig på AI:n och testar att använda den.

4.2.3 Adams åsikter om framtid och AI

Adam tror att man inom en ganska snar framtid bör kunna utnyttja AI likt den typen som ChatGPT är inom programutveckling. Han tror att det finns goda förutsättningar för att man ska kunna använda den för att sätta upp en “mok” eller för att genomföra enklare unit tester. Adam tror att dessa saker som han beskriver som något enklare är något som han tror att en AI kommer kunna lösa åt en programmerare, något han tror skulle kunna underlätta programmerarens jobb. Adam tror att längre in i framtiden kan det bli att man kan använda AI i ännu större utsträckning, för att skriva ännu mer av koden. Han tror dock inte att yrket kommer försvinna helt utan han tror istället att en programmerares roll kommer att ändras något. Adam tror att en programmerare kan komma att vara en person som har en bredare förståelse än att bara kunna koda, en förståelse som kan komma att inkludera att skriva prompts till AI, ha en bredare förståelse för helheten av ett system och för utvecklingsprocessen. Det är något som Adam menar skulle skilja sig en hel del från hur dagens utvecklare fungerar eftersom han menar att de ofta har en tendens att ha ett tunnelseende och endast kunna fokusera på uppgiften som är precis framför dem. Adam beskriver hur han tror att en bredare kunskap för en programmerare skulle leda till att rollen kommer närmare andra roller i programutvecklingen som till exempel produktägaren. Han tror att man kanske inte kommer ha samma behov av alla personer som finns i programutvecklingsprocessen idag. Istället tror han att det kan bli på det sättet att man har en person som kan mer och som vet hur man får AI att genomföra olika former av uppgifter. Dock tror han aldrig att rollerna helt kommer kunna ersättas eftersom han inte tror att en AI kommer kunna endast utgå från ett "Business case” och via det ta fram ett helt system.
Han menar att det är för många olika steg där emellan som han tror AI:n, åtminstone inom den närmaste tiden, behöver hjälp av en människa.

Adam tar dock även upp att det kan finnas frågetecken gällande rättigheter till sånt som AI genererar. Han tar som exempel att man när man använder någon form av AI till något som sedan har ett kommersiellt värde måste man vara försiktig för att man inte ska bryta några former av lagar eller rättigheter. Detta är en av de saker som Adam menar att företag måste anpassa sig till för att kunna utnyttja denna typ av teknologi. Han menar även att företag kan komma att behöva ligga i framkant för att verkligen dra nytta av det. Detta kan till exempel innebära att man utbildar och informerar sina anställda gällande teknologin för att på det sätt se till att man drar maximal nytta av den. För att lyckas med detta tror Adam att man måste välkomna den tekniska utvecklingen som företag och vara öppen för att utnyttja den. Dock

måste de vara vaksamma mot de risker som finns. Detta gör att man enligt Adam måste vara redo att öka användningen men utan att ta för stora risker.

4.3 Peter

4.3.1 Peters bakgrund och erfarenhet

Peter är utbildad systemvetare och har arbetat som programmerare sedan år 2000. Efter att ha studerat systemvetarutbildningen började Peter programmera i C++ inom barnomsorg och skoladministrativa branschen. Peter fortsatte sedan med konsultarbete där jobbet ofta var mot Sony. Peter har framför allt arbetat med Java, backend server programmering och Linux, men även med säkerhetslösningar och kompilatorteknik. Peter har dock aldrig jobbat med användargränssnitt eller konsumentnära utveckling och har till exempel aldrig heller kodat i Javascript.

4.3.2 Peters upplevelse av AI

Peter förklarar att fram till ett halvår sedan hade de inte stött på AI. Tidigare än dess har AI för Peter handlat om att simulera och bygga beslutsträd. Peter har dock testat DALL-E och liknande AI, men är varken skrämd eller imponerad av någon av lösningarna hittills. Peter har hört talas om ChatGPT, men verkar inte ha någon praktisk erfarenhet av det än. Peter verkar ha begränsad erfarenhet av AI, men är medveten om dess existens. När det gäller användningen av AI i sitt arbete, nämner Peter verktyget GitHub Copilot. Peter menar att Copilot kan hjälpa till att snabba upp processen med enkla kommandon men att det inte är okej att använda det i koden som produceras till uppdragsgivaren på grund av problem med immateriella rättigheter. Peter tror att Copilot kan hjälpa till att skriva kod snabbare men kommer aldrig att ersätta det *hantverksmässiga* arbetet som kräver mänsklig kompetens eller den mänskliga förståelsen av vad som händer.

Testningen tas även upp av Peter eftersom det är en fas som ingår i ett utvecklingsarbete. Peter menar att AI definitivt skulle kunna hjälpa till med testning såsom enhetstester, men att det fortfarande kommer att krävas en människa för att penetrera och hitta svagheter. Peter säger att de inte har stött på någon form av AI i produkter de har utvecklat tidigare.

4.3.3 Peters åsikter om framtid och AI

Peter uttryckte sin inställning till AI som spännande och hoppas att AI-tekniken kommer att tillåta de ägna sig åt mer kreativa uppgifter i framtiden. Peter tror inte att de kommer att bli ersatta av AI, men det är inte omöjligt. De legala och etiska frågorna som kommer med användningen av AI är en komplex fråga som inte är lätt att lösa, betonar Peter också. De tror att AI kommer att begränsas av juridiken, att beslutsfattarna i EU och Sverige inte är tillräckligt insatta inom IT, vilket kan fördröja lösningen av dessa frågor.

När det gäller användningen av AI inom programmeringsyrket tror Peter att det kommer att bli ett verktyg för att öka produktiviteten och snabbheten, men att AI inte det enda verktyget. Peter menar att hantverksmässig kompetens krävs fortfarande för att kunna göra jobbet på ett effektivt sätt. Även om AI kan göra en stor del av det tråkiga arbetet, är det fortfarande människor som behöver kontrollera och justera koden för att säkerställa dess kvalitet. Ett exempel på enklare uppgifter enligt Peter som AI kan komma att klara av är enhetstester, som han menar är något som definitivt skulle kunna skrivas av AI när tekniken blir lite bättre. Han fortsätter med att säga därefter att enkla tester kommer AI säkert kunna hjälpa till med, men aldrig att någon form av AI kommer kunna ersätta en riktigt bra testare som verkligen försöker testa hans kod i hans ögon.

Peter är medveten om riskerna med att implementera och använda AI, inklusive att förlora kvaliteten om tekniken används utan att förstå helheten. Därför säger Peter att det är viktigt att ha en bra förståelse av AI-teknik, dess möjligheter och begränsningar för att kunna använda det på ett säkert och effektivt sätt. Avslutningvis kan man sammanfatta att Peter verkar ha en positiv syn på AI som en hjälpande hand, men tror inte att det kommer att ersätta människor inom sitt yrke.

4.4 David

4.4.1 Davids bakgrund och erfarenhet

David, med mer än 11 års erfarenhet som erfaren fullstackutvecklare, inledde sin karriär som programmerare på en yrkeshögskola, lockad av sin passion för teknik, tekniska lösningar och kreativt tänkande. Han har bekvämt hanterat flera programmeringsspråk, inklusive HTML, CSS, Javascript, Node.js och .Net. Han har huvudsakligen verkat inom finans- och e-handelssektorerna, och har skapat både webbplatser och system. Idag innehar David rollen som lead developer på ett företag som specialiserar sig på IT-lösningar för externa kunder. I denna position står han inför ökad ansvar, särskilt inom backendutveckling. Trots sin kompetens som fullstackutvecklare föredrar han att fokusera på backend. Han är en aktiv deltagare i korta och intensiva utvecklingsperioder, kallade "sprints," där ett team på 8 personer strävar efter att skapa värdefulla artefakter. Även om dessa perioder ibland kan vara stressiga när saker inte går enligt plan, finner David generellt nöje i utmaningarna och uppskattar arbetets roliga aspekter.

4.4.2 Davids upplevelse av AI

David har stött på AI under sitt arbetsliv framförallt som delar av de IT lösningar som han varit med och tagit fram. Inom e-handel har han arbetat på system som använder AI för att förutspå hur en kund ska handla och baserat på det erbjuder de olika erbjudanden. David beskriver att detta inte är något som liknar AI som vi idag fått lära känna i form av ChatGPT men den använder några liknande principer i form av maskininlärning. David har även stött på AI i form av en chatbot som man kan prata med när man handlar på en hemsida. David menar dock att de inte var i närheten av lika utvecklade som Chat GPT, utan de kunde bara besvara väldigt enkla frågor. David säger dock att han personligen inte arbetat på att utveckla någon av lösningarna utan att det endast är delar som varit integrerade i större projekt som han har varit en del av. När det kommer till att använda AI för att hjälpa till i utvecklingsarbetet säger David att han aldrig varit med om något som de använt. Han känner dock till att det finns lösningar som han tror hade kunnat hjälpa till att effektivisera processen av att programmera. Medan han känner till att ChatGPT kan programmera tänker han framförallt på verktyg som “Copilot” som han beskriver är direkt integrerat och därför lättare att dra nytta av. David har generellt sett tyckt att det varit kul att se den utvecklingen som hänt runt AI det senaste halvåret. Som teknikintresserad tycker han att denna typ av utveckling är rolig och tycker det är intressant att följa hur andra liknande lösningar växer fram. Han ser utvecklingen som en enorm möjlighet för samhället och tycker det är viktigt att vi använder den på rätt sätt och för att det ska vara hållbart i längden.

4.4.3 Davids åsikter om framtid och AI

I framtiden tror David att AI kommer att få en större roll i utvecklingsarbetet. Han tror att man kommer kunna använda AI som liknar den som ChatGPT är i flera olika delar av processen. Han tror att det kommer att gå att använda för att skapa olika former av testfall och för att skriva unit tester under testprocessen. Han tror att den kommer kunna skriva enklare

delar kod som användaren sedan kan komplettera under programmeringsprocessen och han tror även att den kommer kunna genomföra delar av den dokumentation som behövs under arbetet. David tror att användning av denna typ av AI skulle kunna hjälpa till att effektivisera processen samtidigt som det skulle kunna höja kvaliteten av koden som skrivs och minska antalet fel. Dock tror han fortfarande att AI har sina begränsningar och att den aldrig kommer lyckas ersätta människan helt utan de kommer fungera som ett verktyg som någon form av programmerare utnyttjar under utvecklingsprocessen. David är inte orolig för att AI skulle kunna ta hans jobb eftersom han menar att innan det är möjligt tror han att det fortfarande behövs en hel del utveckling. Istället ser han fram emot att få vara med om en utveckling inom yrket som han tror kommer börja ställa nya frågor och utmaningar till en programmerare. Han tror att de i framtiden kanske kommer bli lite mindre personer i ett agilt team om AI:n nu lyckas få det inflytande som han tror. Detta han tror att om effektiviteten ökar kan ett fåtal personer få lite fler uppgifter och fortfarande plocka fram ett program. Han tror även att en programmerare kan komma att behöva utbilda sig och bli bra på att instruera en AI att göra något åt den eftersom yrket kan komma att innehålla en del av den varan.

I framtiden tror David att verksamheter kommer behöva anpassa sina processer att man utnyttjar AI för att fortsätta vara konkurrenskraftig. Detta tror han att man måste göra både i form av hur man får sina arbetare att arbeta med AI genom att se till att de har lärt sig och kan göra det på bästa möjliga sätt. Men David ser även att man måste se till att ha koll på lagar som omger användningen av AI. Detta tror David kommer att vara det största hindret som man kommer behöva ta sig förbi för att kunna dra maximal nytta av AI eftersom han tror att många företag kommer att vara rädda för att göra fel i en ganska oprövad rättslig svär.

4.5 Liam

4.5.1 Liams bakgrund och erfarenhet

Liam har läst programmering i två år till fullstackutvecklare på en yrkeshögskola. Efter utbildningen fick de praktik på ett företag som fokuserar på bokföringsprogram för mindre företag. Liam har nu arbetat där i 2-3 år som fullstackutvecklare. Det framgår att de har erfarenhet av både frontend och backend-utveckling, med betoning på Javascript, Typescript och Java. De har också genomfört egna hobbyprojekt och har erfarenhet av imperativ programmering. Liam nämner även att de har arbetat med arduinos och liknande enheter, såsom ISP-kretskort, vilket tyder på att de har kunskaper och lite erfarenhet inom inbyggda system och elektronik. För cirka sex månader sedan blev de även *Tech lead*, vilket innebär att de ansvarar för systemförvaltning och att deras team undviker teknisk skuld. Detta inkluderar till exempel att ta bort död kod, optimera systemet och se till att det är välskött. I sitt arbete förvaltar Liam och utvecklar nya funktioner för befintliga program, inklusive insticksmoduler för att förbättra och underhålla de befintliga systemen.

4.5.2 Liams upplevelse av AI

Liam framhäver att de inte har upplevt AI i stor utsträckning, men de har stött på integrerad AI-teknik som hjälper till att få insikter och extrahera data från programmen de arbetar med. Liam är bekant med *ChatGpt* och ser det som ett verktyg för att formulera text eller meddelanden på bästa sätt. De betonar dock att utvecklingsarbetet utförs av ett team av utvecklare och att AI-verktyg som *ChatGpt* inte används i utvecklingsprocessen.

Liam uttrycker att de ser AI som ett bra verktyg, men de betonar att det inte kan göra hela jobbet åt en. De nämner att det finns en viss oro och osäkerhet kring AI i relation till GDPR och behandlingen av personuppgifter. De anser att om det finns en lösning som kan integrera

AI med respekt för GDPR, till exempel genom anonymisering av data, kan det vara acceptabelt. Liam är däremot mer intresserad av att se hur AI kan användas inom viktiga områden som medicin, men de tror att det finns för många lagar som begränsar användningen av AI inom deras yrke men även över huvud taget inom jobbsammanhang.

När det gäller språkmodeller är Liam medveten om att de utgör grunden för chatbotar som *ChatGpt* och används för att tolka mänskligt språk. De anser att språkmodeller och generativ AI, som *ChatGpt* och *Midjourney*, har potentialen att användas på viktigare sätt än vad de används till idag, vilket de betraktar som mestadels löjliga saker. De uttrycker en viss skepticism och osäkerhet kring AI-teknikens funktion och hur långt den kan tas.

Liam väljer aktivt att inte använda AI i sina arbetsuppgifter som utvecklare. De tycker att skrivandet av kod är det som de tycker är mest intressant och roligt. De ser inte AI-verktyg som en ersättning för den problemlösningsspekten som programmering innebär. De är medvetna om att det finns andra AI-verktyg som kan hjälpa utvecklare med kodning, som *GitHub Copilot*, men de ser ingen nytta med att använda dem från sitt perspektiv som utvecklare, eftersom de tycker att kodning i sig själv är kul. De kan dock tänka sig att använda *ChatGpt* i sina hobbyprojekt, men inte på jobbet.

4.5.3 Liams åsikter om framtid och AI

Liam tror att AI kommer att påverka deras yrkesroll positivt genom att förbättra kod kvaliteten och minska misstag. De anser dock att integreringen av AI är ett komplext problem, eftersom AI fortfarande är väldigt nytt, men däremot menar de att användningen av AI till exempel inte bör regleras inom verksamheter, förutsatt att det inte finns konkreta problem eller lagar emot det. Liam betonar att AI behöver anpassas både efter marknaden och landspecifika lagar. När det gäller deras yrkesroll anser de att AI skulle kunna fungera bra inom frontend-utveckling, men inte lika bra inom backend-utveckling där det är mer komplext och anpassat till specifika verksamheter. De tror att AI-tekniken kan lösa komplexa problem, men de menar att den visar brister när det gäller företagsspecifika problem eller marknadsanpassade uppgifter där människor är bättre lämpade.

Liam tror att anpassning till AI-utvecklingen kan vara nödvändig för alla. De ser fram emot denna utveckling men känner också en konstig känsla kring den. De föreslår att utbildning av anställda inom området är viktigt och att användningen av AI bör vara för att effektivisera arbetet. Samtidigt, Liam uttrycker sin rädsla för att AI kan användas till dåliga syften och ser risker i att AI kan agera oönskat, eller att dess utveckling kan leda till problematiska situationer.

Avslutningsvis tror Liam att utvecklare generellt har ett starkt intresse av att lära sig om AI och att de flesta skulle kunna anpassa sig om AI blir mer relevant inom deras bransch. De ser AI främst som ett verktyg, till exempel *ChatGpt*, och tror inte att AI kommer att kunna ersätta utvecklare på kort sikt. De ser dock till exempel möjligheter för AI att påverka transportindustrin i första hand, däremot ser de svårigheter med AI i att ersätta utvecklare på grund av huvudsakligen komplexiteten i att utveckla system som är specifika på sina olika sätt.

4.6 Simon

4.6.1 Simons bakgrund och erfarenhet

Simon berättar att han har en utbildning som systemvetare, men han har alltid känt att utveckling och programmering är hans starka sidor. Efter examen hade Simon svårt att hitta ett jobb som programmerare och tog istället en roll som testare, vilket har fungerat väl för honom under de senaste 1,5 åren. Simon arbetar i en mindre grupp som använder agil metodik för att underhålla och utveckla nya delar av företagets IT-system. Simon berättar att teamet bland annat består av en testare, ett par utvecklaren, en produkt ägare. Trots att hans stora intresse ligger inom programmering, berättar Simon att han lärt sig mycket som testare och har blivit bättre på kodning genom att vara nära processen och se mycket kod vilket han hoppas kunna ta sig med in i framtiden. Han trivs bra i sin nuvarande roll som testare och har som mål att hitta en roll som programmerare i framtiden, men är nöjd med sin nuvarande roll.

4.6.2 Simons upplevelse av AI

Simon berättar att han aldrig har använt AI för att direkt underlätta en arbetsuppgift. Däremot har han använt chattfunktioner för att fråga om saker som gäller kod eller hur saker bör fungera, men han har aldrig låtit AI skriva några testfall eller använt det för kodning för olika tester. Han är också osäker på om någon i hans team använder AI för detta ändamål. Trots detta har han lekt runt med AI på fritiden och testat vad det kan göra, och har några hobbyprojekt där han testar att be AI om kodsnuttar. Men han tycker inte att chattfunktionen är helt optimal för den här typen av arbete, eftersom han har svårt att få fram precis vad han vill. Om det är något han redan kan, är det oftast lättare att bara skriva det själv eller använda verktyg som Copilot som är direkt inbyggt. Simon har inte arbetat med någon produkt som har AI integrerat i sig och har därför ingen erfarenhet av att använda AI i ett professionellt sammanhang.

4.6.3 Simons åsikter om framtid och AI

Simon tror att man kommer kunna dra nytta av AI i framtiden. Även om det kanske inte kommer att vara chatt, tror Simon att verktyg som Copilot, när den har lärt sig mer, kommer att vara väldigt användbara för både programmerare och testare. Simon tror också att det kommer att dyka upp många liknande verktyg i framtiden, eftersom det verkar som om något nytt släpps varje dag. Det finns dock oro kring regler och lagar som kan komma att skapa problem när man ska använda AI i kommersiellt syfte, eftersom "... man inte vet vem som äger det som botten skapat". GDPR och andra lagar kan skapa problem när det gäller att avgöra vem som äger vad när det kommer till material som har skapats av en AI. Simon tror att detta är något som måste lösas. Simon tror också att AI kan förändra programutvecklingen genom att minska antalet personer som behövs för att skriva och testa kod. Om en bot kan skriva en stor del av koden och dessutom testa den, kan det finnas utrymme för att ha några färre personer med en bredare kunskap som kan hantera allt och få allt att flyta på. Det kan också innebära att yrkena testare och programmerare kanske kommer att slås ihop till ett mer effektivt yrke. Dock tror Simon inte att AI kommer att ta över jobben inom programmering och testning helt och hållet, eftersom det fortfarande behövs tekniskt kunniga personer som kan hjälpa till när nya problem uppstår och när AI behöver lära sig nya saker. Simon tror också att det är programmerare som utvecklar och förbättrar AI, vilket är ett tecken på att människor fortfarande behövs för att AI ska kunna fungera på ett optimalt sätt.

4.7 Hugo

4.7.1 Hugos bakgrund och erfarenhet

Hugo inledde sitt programmeringsintresse med extra kurser parallellt med gymnasiet. Vidare studerade han datavetenskap på LTH. Under sin högskoletid etablerade han ett eget konsultföretag och fokuserade på små och medelstora företags projekt. Han agerade som konsult för ett företag och samarbetade med IKEA för utveckling av en applikation för virtuell husplanering. Vidare arbetade han på en bank där han skapade en betalnings-widget för banksystemet. Hugos nuvarande anställning involverar att stödja medicinföretag inom klinisk forskning, genom att utveckla en plattform för deltagarurval och statistikhantering. Hans omfattande programmeringserfarenhet omfattar flera branscher och inkluderar expertis inom testning, inklusive enhetstester och integrationstester. Hugo betonar att utvecklare bör ansvara för vissa tester innan dedikerade testare tar över, med tanke på att testning är integrerad i utvecklingsprocessen.

4.7.2 Hugos upplevelse av AI

AI har korsat Hugos väg både direkt och indirekt. Indirekt stötte Hugo på AI under sin tid när han jobbade på banken genom användning av maskininlärning inom *fraud detection*.
När Hugo pratar om AI menar han främst *narrow* AI snarare än generell AI. Hugo säger att AI har visat sig vara framgångsrikt på flera olika sätt. Han menar att med AI blir det möjligt att identifiera komplexa mönster som människor själva kan ha svårt för att upptäcka, särskilt i stor skala. Hugo har även direkt stött på AI inom ett snabbt växande område som riktar sig specifikt till utvecklare. Det finns flera produkter tillgängliga för användning, säger han, som till exempel GitHub Copilot. Detta verktyg enligt honom visar sig vara mycket värdefullt och fungerar i princip som en *peer*-programmerare. Genom att använda en programmerares mänskliga språk kan man dra nytta av verktyget för att underlätta olika uppgifter, såsom implementering och generering av tester och testdata, allt i linje med önskade mål.

Hugo fortsätter med att säga, att det utöver dessa specifika exempel på AI-verktyg pågår för närvarande en växande diskussion om hur man effektivt kan utnyttja AI, med tanke på de ständigt framväxande nya verktygen inom branschen som driver sådana diskussioner. Utifrån Hugos observationer hittills hjälper dessa verktyg främst med mindre kod relaterade aspekter, eftersom verktygen fortfarande befinner sig i ett tidigt skede i sin utveckling. Personligen säger Hugo och inom branschen som helhet håller vi på att upptäcka de mest effektiva sätten att använda dessa utvecklande verktyg. Hugo menar att han skulle fortfarande klassificera dem som relativt nya. Han menar att med tanke på den snabba utvecklingstakten skulle det inte vara överraskande om hans svar på detta ämne helt förändras om sex månader, med tanke på den nuvarande utvecklingens fart.

4.7.3 Hugos åsikter om framtid och AI

När det gäller åsikter om AI idag och i framtiden framhäver Hugo att verktyget GitHub Copilot är fantastiskt för att stödja programmeraren på en enhetsnivå, särskilt med enklare uppgifter och funktioner där människan har kontroll över både input och output på ett enkelt sätt. Det noteras även av Hugo att dessa verktyg ständigt blir smartare, vilket främjar deras användning även i andra projekt. Vid användning av liknande verktyg säger Hugo att det vanligtvis krävs några justeringar, men generellt sett kan upp till en hög nivå av genererad kod vara korrekt, beroende på kvaliteten av de angivna specifikationerna.

Hugo betonar att AI kan ta bort mycket av det *tråkiga* genom AI-teknologier som utvecklas allt mer. Detta innebär att en stor del av programmerings arbetet automatiseras, menar Hugo.

Hugo säger att det förutspås dock att detta kommer att utmana programmeringsyrket och ställa det inför en prövning. Han menar att många har redan argumenterat för att programmerare inte bara är kodare utan utvecklare som tar ett problem, formulerar det på ett konkret sätt och hittar en lösning utifrån den. Detta blir särskilt tydligt betonar han med framväxten av AI-verktyg. Verktygen är inte bra på att göra det som utvecklarna gör menar han, men beroende på de angivna parametrarna kan verktygen prestera väl eller dåligt. Hugo förväntar att det naturliga utvecklingssteget under de kommande åren blir att kodare och programmerare integrerar sina roller med AI, dock det som kommer att skilja utvecklarna är deras förmåga att formulera problem och förstå både människor och maskiner samt översätta mellan dem.

Hugo säger att framsteg inom AI förväntas leda till snabbare och effektivare utförande av arbetsuppgifter, vilket kan påverka hur företag betraktar sina anställdas roller ur ett resursperspektiv. Det kan leda antingen till minskade teamstorlekar eller att arbetsuppgifter och domäner blir mer omfattande. När det kommer till ChatGPT, Hugo har begränsad erfarenhet av ChatGPT, men känner till några kollegor som använder det tillsammans med andra AI-verktyg som Midjourney och DALL-E för att generera bilder. När det gäller problem och utmaningar med användningen av AI påpekar Hugo att dessa verktyg kan användas för både bra och dåliga ändamål. Han påstår att det förväntas uppstå lagliga och moraliska problem, som tidigare tekniska framsteg har också mött och löst. Han menar att AI kommer inte vara något annorlunda i det avseendet.

Hugo säger att han är något orolig för att AI kan ta över hans jobb, särskilt i den formen AI finner sig idag och hur snabbt tekniken utvecklas. Dock hävdar han att så länge diskussionen handlar om generell AI kommer det att finnas andra jobbmöjligheter för hans yrkesroll som utvecklare. Inom hans livstid menar han att de inte behöver oroa sig som kodare eller utvecklare, men att rollen kommer att genomgå betydande förändringar och innebära nya utmaningar.

4.8 Kodning

De identifierade teman som kommer att tas upp för den data som vi samlat in för våra intervjuer är

- Bakgrund och Erfarenhet
- Upplevelse av AI
- Åsikter om Framtid och AI

4.8.1 Bakgrund och erfarenhet

De personer som har intervjuats utgör en blandning av testare och utvecklare. Deras yrkeserfarenhet varierar från ett och ett halvt år till över tjugo års erfarenhet. Respondenterna har olika utbildningsbakgrund, där två av dem har utbildning från yrkeshögskolan, tre har utbildning inom systemvetenskap och den sista har erfarenhet från en oavslutad utbildning inom datateknik. En trend som framkom i fyra av de sex intervjuerna var att respondenten arbetade i utvecklingsteam där det fanns personer med olika roller.

Tabell 8. Respondenternas bakgrund och erfarenhet

Respondent	***Värdering***
Adam	Testare med strax över två års erfarenhet.

	Har en oavslutad utbildning som datateknik och jobbar i ett agilt team där testare och utvecklare jobbar nära varandra.
Peter	Har arbetat som utvecklare i mer än 20 år och har erfarenhet av att arbeta på projekt relaterade till en mängd olika sektorer. Har sedan tidigare en utbildning som systemvetare.
David	Har 11 års erfarenhet av att arbeta som utvecklare. Är i grunden utbildad på en yrkeshögskola. Idag arbetar respondenten i ett mindre agilt team som arbetar i sprinter.
Liam	2 års programmeringsutbildning till fullstackutvecklare. Jobbat som fullstackutvecklare i 2-3 år. Han arbetar för närvarande med systemförvaltning och utvecklar nya funktioner för befintliga program.
Simon	Har arbetat som testare i ett och ett halvt år och har sedan tidigare en utbildning som systemvetare. Har ett stort intresse för utveckling och har i framtiden ambitioner att arbeta som utvecklare.
Hugo	Han har datavetenskaplig utbildning och bred erfarenhet inom konsultarbete, inklusive samarbete med IKEA och arbete inom banksektorn. Nu arbetar han med utveckling mot medicinföretag inom klinisk forskning.

Tabell 9. Respondenternas bakgrund och erfarenhet i kategorier

Kategori	***Respondent***					
	Adam	Peter	David	Liam	Simon	Hugo
Yrkesroll Utvecklare		X	X	X		X
Yrkesroll Testare	X				X	
Junior (1-2 års erfarenhet)					X	
Mid (2-3 års erfarenhet)	X			X		

Senior (4+ års erfarenhet)	X	X	X

4.8.2 Upplevelse av AI

Under intervjuerna visade alla de personer vi intervjuade en positiv inställning till utvecklingen kring AI, och de ansåg att det var roligt att följa denna utveckling. Dock uppgav endast tre personer att de på något sätt hade utnyttjat AI i sitt arbete. Av dessa hade två personer inte använt det för att skriva kod eller utföra tester, utan de använde det för att förbättra text som skulle skrivas ut i systemet. En av dem hade dock utnyttjat exempelvis GitHub Copilot för mindre kodning. Det innebär att ingen av de sex respondenter vi intervjuade hade kommit i kontakt med ChatGPT under arbetstid. Endast en respondent uppgav att de använde AI-verktyg under arbetstid. Peter hade tidigare sett sina kollegor använda tekniken under en kort period, men arbetsgivaren hade sedan reglerat det för att de inte längre skulle få göra det. Alla respondenter berättade att de på något sätt hade använt ChatGPT eller andra språkmodell baserade AI-verktyg på sin fritid. Simon berättade hur han hade använt ChatGPT för att hjälpa honom lösa programmeringsproblem, men påstod att han hade fått blandade resultat. Simon föredrar för närvarande att använda andra AI-verktyg, som Copilot, för att lösa den typen av problem istället. Både Adam och Peter pratade om hur de hade använt AI-verktyg för att rita bilder eller liknande. Adam uttryckte att han var extremt imponerad av resultaten som man kan uppnå med verktyget Midjourney, medan Peter, som använde DALL-E, ansåg att han hade fått mer blandade resultat under sin användning. Hugo berättade om GitHub Copilot, men visade även kännedom för AI-verktyg som ChatGPT, Midjourney och DALL-E.

Tabell 10. Respondenternas upplevelse av AI

Respondent	***Värdering***
Adam	Adam har använt ChatGPT under utvecklingsprocessen och för att utveckla sig själv som testare. Dock har han aldrig använt AI som ett verktyg för att testa eller skriva kod i sitt yrke.Respondenten ser totalt sätt positivt på den senaste utvecklingen av AI.
Peter	Peter beskrev hur han tidigare sett kollegor utnyttja verktyget Copilot men att de nu fått information om att de inte ska undvika det. Utöver detta har Peter inte stött på användning av AI som ett verktyg i utvecklingsprocessen.
David	David tycker att utvecklingen av AI de senaste åren varit väldigt kul att följa och ser positivt på att man bör kunna utnyttja den i sitt yrke. Han har dock aldrig stött på det och tror att det kan bero på delvis hur nytt det är och delvis de frågetecken som finns runt lagar inom området.

Liam	Liam har en positiv inställning till den utveckling som har skett runt AI det senaste året. Han har endast använt AI i programutvecklingsprocessen för att förbättra till exempel text som ska vara med i programmet och men har inte stött på det som ett hjälpmedel för att koda.
Simon	Simon har inte stött på AI på sin arbetsplats som ett verktyg som han tror går att använda i programutvecklingsprocessen. Dock har Simon testat att utnyttja AI i den samma på sin fritid med vad den beskriver som mixade resultat. Totalt sett är den dock positiv i sin inställning mot den utveckling av AI som skett.
Hugo	Hugo har praktisk erfarenhet av AI genom både direkt och indirekt användning. Han anser att AI framgångsrikt kan identifiera komplexa mönster och underlätta specifika uppgifter för utvecklare. Han har själv använt AI-verktyg såsom *GitHub Copilot* inom sitt arbete och betraktar liknande verktyg som värdefulla. Han noterar att verktygen främst hjälper med mindre kod relaterade aspekter och att branschen fortfarande utforskar de mest effektiva sätten att använda dessa utvecklande verktyg.

Tabell 11. Respondenternas upplevelse av AI i kategorier

Kategori	***Respondent***					
	Adam	Peter	David	Liam	Simon	Hugo
Har använt AI som verktyg i utvecklingsprocessen	X		X			X
Har använt AI som verktyg till att utveckla eller testa kod						X
Positiv inställning till AI	X	X	X	X	X	X
Negativ inställning till AI						

4.8.3 Åsikter om framtid och AI

När det kommer till hur våra intervjupersoner ser på framtiden och AI finns det väldigt många trender. Alla våra respondenter har en optimistisk syn på att man kommer kunna utnyttja AI till tester och programmering i framtiden. Alla tror även att rollen av en testare eller en utvecklare kommer att få anpassas för att man ska kunna dra nytta av användningen. Dock är även problematik med implementeringen något som alla såg men lite i olika utsträckning. De allra flesta tror att det kommer finnas problem med lagar och etik som kan påverka hur snabbt man vågar börja använda AI inom företag. Det fanns även en tydlig tro kring att företag kommer att få anpassa sig som en konsekvens av AI utvecklingen.

Tabell 12. Respondenternas åsikter om framtid och AI

Respondent	***Värdering***
Adam	Adam tror att man ganska snart kommer använda AI i programutveckling. Till en början tror han det kommer vara för mindre och enklare uppgifter men längre fram för mer komplicerade. I längden tror han att det kommer leda till att en del olika roller inom programutvecklingen kommer att sättas ihop eftersom alla inte kommer att behövas.
Peter	Tycker det ska bli intressant att se hur AI kan användas i utvecklingsprocessen i framtiden. Tror inte att AI kommer kunna ersätta människor utan att det endast kommer att agera som ett verktyg. Ser dock en del problematik kring lagar och det etiska som måste redas ut innan man kan börja använda AI i kommersiellt syfte.
David	Ser fram emot att vara en del av den utvecklingen som kommer ske inom yrket till följd av AI likt Chatt. Tror att AI:n endast kommer att fungera som ett verktyg för användaren och att det fortfarande kommer att behövas utvecklare och testare bredvid.
Liam	Liam tror att AI kommer kunna vara ett effektivt AI verktyg som utvecklare och testare kommer kunna använda. Han tror att yrkesverksamma kommer ha lätt för att anpassa sig. Dock ser han lite problem i att det finns risker med att AI inte uppför sig som förväntat. Liam tror inte att AI kommer kunna ersätta yrkesverksamma på grund av komplexiteten av kod.
Simon	Simon tror att AI kommer kunna vara en del

	av programutvecklingsprocessen i framtiden men tror det kommer vara andra verktyg än just ChatGPT. Dock tror Simon att det finns en del legala delar som måste lösas innan företag kan sätta verktygen i arbete. Simon är dock tydlig med att de inte tror att yrken kommer att ersättas helt.
Hugo	Hugo ser potential i AI-verktyg som GitHub Copilot och tror att utvecklarrollen kommer att förändras. Han är orolig för att AI kan ta över vissa jobb men förväntar sig också lagliga och moraliska utmaningar med tekniken.

Tabell 13. Respondenternas åsikter om framtid och AI i kategorier

Kategori	***Respondent***					
	Adam	Peter	David	Liam	Simon	Hugo
Optimistisk syn på framtiden	X	X	X	X	X	X
Neutral syn på framtiden						
Pessimistisk syn på framtiden						
Tror att AI kommer kunna ersätta testare eller utvecklare helt						X
Tror att testare och utvecklares roller kommer få anpassas	X	X	X	X	X	X
Ser problem runt att implementera AI i processerna	X	X	X	X	X	X
Tror företag kommer få anpassa sig efter utvecklingen	X	X	X	X	X	X

5 Analys och Diskussion

5.1 AI inom programutveckling idag

I litteraturen diskuteras användningen av AI inom mjukvaruutveckling och hur den påverkar olika faser av SDLC. Författarna (Harman, 2012; Shafiq, et al., 2023; Biswas, 2023; Aljanabi, et al., 2023) nämner flera exempel på hur AI-tekniker används för att automatisera uppgifter och förbättra programutvecklingsprocessen. Detta inkluderar automatiserad mjukvarutestning, mjukvarudebugging, intelligent kodkomplettering, förutsägande underhåll samt mjukvaruoptimering. Enligt författarna har AI potentialen att revolutionera mjukvaruutvecklingen genom att göra processen snabbare, effektivare och mer exakt.

Litteraturen (Shafiq, et al., 2023) visar på att genom att tillämpa till exempel maskininlärning inom olika faser av SDLC kan utvecklare hantera problem som oklarhet i mjukvarukrav, identifiering av designsvagheter, automatiserad kodgranskning, förutsägelser av kodfel, samt upptäckt och analys av sårbarheter och avvikelser i mjukvaror. Liknande ML-tekniker och andra neurala nätverksbaserade ML-tekniker som djupinlärning används för att lösa problem av samma sort som ibland är svåra att analysera eller hantera för människor.

Biswas och Aljanabi (2023, 2023) tar upp användningen av ChatGPT inom mjukvaruutveckling. Enligt författarna har ChatGPT potentialen att förbättra interaktionen mellan människor och teknik. Den kan användas för att förbättra bearbetning och förståelse av naturligt språk, driva chatbots och virtuella assistenter samt underlätta konversationer och interaktioner via röst och text. Författarna hävdar att det finns flera fördelar med att använda ChatGPT inom mjukvaruutveckling, såsom förbättrad kodning genom kodkomplettering, korrigering och generering av dokument. Litteraturen säger även att ChatGPT kan förstå inmatningar av naturligt språk och ge kontextuellt relevant information baserat på kodavsnitt eller kommandon. Den kan också generera kod och automatisera repetitiva uppgifter. Utöver fördelarna noteras också begränsningarna med användningen av ChatGPT. Enligt Aljanabi et al. (2023) kan begränsningarna inkludera begränsad tillgänglighet, stora kostnader samt bristen på fullständig förståelse för nyanserna i programmeringsspråk.

Respondenterna har haft olika erfarenheter av AI inom sina respektive områden, vilket visar på både likheter och skillnader i hur AI har använts. Alla respondenter har kommit i kontakt med AI-teknik, även om graden av exponering har varierat. De är bekanta med AI-verktyg som ChatGPT och GitHub Copilot. Respondenterna nämner att de i någon form har använt AI-verktyg, men alla har inte praktisk erfarenhet av att använda dem i arbetssammanhang. När det gäller användning av AI berättar respondenterna att det sker på olika sätt.

Adam använder ChatGPT för att anpassa text, kontrollera grammatikfel och få förslag på tonlägen. Utöver det har han ingen erfarenhet av att använda AI som ett verktyg i sitt arbete, vilket innebär att han inte utnyttjar det för att utföra testning av kod. Peter nämner att GitHub Copilot är ett verktyg som kan påskynda kodningsprocessen, men han använder det inte själv på grund av immateriella rättigheter. Trots att Peter har lång erfarenhet inom programutveckling, påpekar han att han aldrig har använt eller stött på AI-verktyg i sitt arbete. Hugo använder GitHub Copilot för mindre kodsnuttar i sitt arbete och har inte haft några problem med det. Dock har han ingen erfarenhet av ChatGPT och andra AI-verktyg i sitt arbete. Han ser potentialen i dem, men eftersom de fortfarande är relativt nya är det oklart hur man bäst kan dra nytta av dem i sin yrkesroll. David har ingen erfarenhet av att utnyttja någon form av AI i sitt arbete. Liam har stött på integrerad AI-teknik för att extrahera insikter och data från programvara, men varken förlitar sig på direkt användning av AI-verktyg eller

tycker om att använda dem vid kodning. Detta eftersom han finner själva kodningsprocessen rolig och ser ingen anledning att använda AI i sitt arbete. Simon har experimenterat med AI på sin fritid men har ingen professionell erfarenhet av att använda det. Simon anser att han har haft blandad nytta av verktygen när han har arbetat med dem, eftersom vissa saker är lättare för honom att skriva själv än att försöka hitta rätt prompt för det han försöker skapa.

Det framgår tydligt att respondenterna använder AI-verktyg i programutvecklingsprocessen i en relativt begränsad utsträckning, trots att det finns en tydlig medvetenhet om dess existens. Vid användning av AI-verktyg finns det en tydlig trend bland våra respondenter, AI-verktyg används i högre grad för kodning än för testning. Detta kan förklaras av att vissa av de mer välkända AI-verktygen är främst utformade för att underlätta kodningsrelaterade uppgifter snarare än att stödja testprocessen.

Generellt är användningen av AI låg bland respondenterna, vilket kan bero på flera faktorer. En potentiell faktor kan vara de företagen som respondenterna arbetar för. Vissa företag kan vara mer innovativa och sträva efter att dra nytta av AI i ett tidigt skede, medan andra kan föredra att vara mer försiktiga och avvakta för att få ökad klarhet kring till exempel de rättsliga aspekterna inom området. En annan faktor kan vara att denna typ av AI, som till exempel ChatGPT är, fortfarande är relativt ny. Vissa företag kan ha tydliga avsikter att implementera AI som verktyg i sina verksamheter, men de föredrar att ta sin tid för att hitta det bästa sättet att genomföra detta på. Andra kan köra på en strategi där de testar sig fram för att hitta den bästa lösningen och implementerar därför verktygen tidigare. Detta kan också vara kopplat till hur olika roller är utformade inom företagen.

Enligt Fishpool och Fishpool (2020), som pratar om roller inom programutveckling, kan det vara en del av en utvecklares uppgift att hålla sig uppdaterad om de senaste IT-utvecklingarna och relevanta lagar som gäller för deras arbetsuppgifter. Vissa av våra respondenter kan ha dessa som sina uppgifter vilket gör det möjligt för dem att dra nytta av AI i sina utvecklingsprocesser. Några av de andra respondenterna kan ha andra personer inom företagen de jobbar på som tar hand om dessa uppgifter, vilket kan leda till en längre implementeringstid av AI-verktygen.

Både respondenterna och en del av litteraturen (Harman, 2012; Shafiq, et al., 2023; Biswas, 2023; Aljanabi, et al., 2023) är överens om att AI har potentialen att förbättra processen för programutveckling och erbjuda fördelar som automatisering, förbättrad kodning och ökad effektivitet. Dessutom uppfattar de att verktyg som ChatGPT kan vara användbara inom programutveckling, särskilt för att förbättra bearbetningen och förståelsen av naturligt språk. Detta kan över en längre period utveckla interaktionen mellan människor och teknik till det bättre. Respondenternas svar tillsammans med litteraturen kan leda till uppfattningen att detta inte endast gäller ChatGPT i sin nuvarande form utan även för liknande verktyg som har samma potential. ChatGPT är ett stort steg framåt mot detta mål, vilket går att återfinna i både litteraturen (Biswas, 2023; Aljanabi, et al., 2023) och respondenternas svar.

En korrelation mellan respondentsvaren och litteraturen (Harman, 2012; Shafiq, et al., 2023) är att AI redan idag används i någon form inom programutveckling, även om samtliga respondenter själva inte gör det. Både respondenterna och litteraturen lyfter fram farhågor och begränsningar med AI idag, som till exempel brist på fullständig förståelse för nyanserna i programmeringsspråk och den problematik som man försöker lösa med IT-systemen. En skillnad som märks är att respondenterna ger en mer individuell och specifik bild av sin erfarenhet av AI inom sina respektive arbetsområden. Respondenterna har generellt sätt liten

erfarenhet av att använda AI inom programutveckling, men bland deras erfarenheter skiljer sig användningsområdena. Å andra sidan ger litteraturen en bredare översikt över användningen av AI inom programutveckling baserat på forskning och flera författares perspektiv. Det är värt att notera igen att respondenterna har en mer praktisk och konkret erfarenhet av AI genom sin faktiska användning av verktyg som ChatGPT och GitHub Copilot, medan litteraturen ger en mer övergripande och teoretisk diskussion om AI-användning inom programutveckling. Respondenterna därmed visar att AI inte används i lika stor utsträckning som litteraturen menar på. De tar även upp flera frågor som man kan stöta på i verkligheten gällande legala eller juridiska hinder som kan förekomma. Hinder som begränsar AI:s användning och möjligheten att utnyttja deras fulla potential.

Sammanfattningsvis går det att se likheter mellan respondenterna och litteraturen, främst i att de båda ser fördelar och begränsningar med AI-verktyg inom programutveckling. Skillnaderna ligger främst i respondenternas individuella erfarenheter och litteraturens bredare perspektiv baserat på forskning och flera författares insikter.

5.1.1 Fördelar

När det kommer till vilka fördelar som skulle kunna uppstå genom att man utnyttjar AI i programutvecklingsprocessen finns det vissa som tagits upp av alla respondenter medan vissa endast nämnts av några enstaka. En tydlig trend var att man tror att AI ska kunna hjälpa programmerare och testare att skriva kod och göra tester mer effektivt än vad det går att göra manuellt. Detta genom att AI:n kan skriva koden eller testerna snabbare än vad en människa kan göra när det kommer till enklare tester eller skelettet till en kod. Detta är något som respondenterna visat att de både tycker är kul och synd. Peter beskrev hur han tycker det kan vara bra eftersom det gör det möjligt för honom att fokusera på de mer komplexa delarna av koden som respondenten tycker är roligare att arbeta med. Liam tar dock upp att en av anledningarna till att han inte använt någon AI för att skriva kod än är för att han tycker det är kul att skriva kod och tycker därför det är synd att låta en AI göra det åt honom.

När det kommer till om AI kommer kunna förbättra kvaliteten på kod, skiljer det sig i vad respondenterna tror. Peter tror inte att AI kommer kunna skriva högkvalitativ kod och att den därför inte kommer kunna skriva bättre kod än en bra kodare. Han menar att det finns en risk med att AI:n lär sig från en specifik kunskapsbas. Peter tar upp att denna typ av kunskapsbas måste kvalitet kontrolleras ifall man ska kunna få ut högkvalitativ kod från den. För att detta skulle höja kvaliteten på kod måste det alltså inte bara vara att AI kan förstå sig på vad den ska göra lika bra eller bättre än en människa. AI-verktyget måste även basera sin kunskap på en databas som innehåller bättre kod än det som en kodare själv skulle kunna skriva. Liam, Hugo och David tar båda upp att de tror att AI kan skriva bättre kvalitativ kod än vad mänskliga kodare kan. Detta eftersom den skulle kunna skriva kod som innehåller mindre fel än vad kod skriven av en utvecklare gör idag. Anledningen till att de olika respondenterna hade olika uppfattning om hur vidare kodnings kvaliteten kommer att höjas eller inte kan bero på att de har olika förväntningar på kvalitet. Respondenten som inte tror att kvaliteten av kod nödvändigtvis kommer att höjas har mer erfarenhet av kodning än vad de andra respondenterna har. Detta kan ligga till grund för att den har en högre förväntan på kvaliteten av koden och kanske även en större kunskap om vad som gör kod dålig eller bra.

5.1.2 Problematik

När det kommer till utmaningar kring att lyckas implementera AI i programutvecklingsprocessen var det framförallt två stycken som vi kunde se en trend av att

det kom upp. Dessa var lagar och etik. När Peter fick höra av sina arbetsgivare att de inte fick lov att utnyttja verktyg som Copilot berodde det på att de inte ville bryta mot några avtal eller lagar för hur de får lov att arbeta. Respondenten beskrev att eftersom de inte visste vilka licenser som Copilot skrev sin kod baserat på ansåg deras arbetsgivare att det var för stor risk att de bryter mot något avtal.

Att inte veta hur AI fungerar eller vad det är som leder fram till dess beslut är två saker som Coeckelbergh (2019) diskuterade som etiska dilemman. Coeckelbergh (2019) tog upp hur personerna som använder AI som ett verktyg bör ha rätten till att känna till just vad det är som leder till att AI tar sina beslut och vad det är som pågår inuti verktygen. Hade Peter och hans arbetskamrater kunnat få fram denna information, hade det blivit tydligare för de om vilka licenser det är som gäller vid användningen av olika AI-verktyg och i vilken utsträckning de kan utnyttja dessa verktyg. Detta hade gjort att de kunnat veta om de får lov eller inte att använda den istället för att bara avstå eftersom det finns en risk att bryta mot vissa licenser.

Även Simon tog upp att han tror att företag kommer behöva vara försiktiga om man ska använda AI-verktyg för att utveckla något som har ett kommersiellt värde eftersom "... man inte vet vem som äger det som botten skapat". Simon tar AI genererade bilder som exempel där han berättar att han läst en del om just rättigheterna runt dessa bilder och den osäkerhet som finns kring bilder som liknar andra verk. Respondenten berättar hur det blir en oklarhet i vem det egentligen är som äger rättigheterna kring det som AI:n har skapat, om det någon gång skulle tas upp i rätten. Även Adam är inne på ett liknande spår som Simon och menar att detta kan skapa en osäkerhet kring att använda just AI fram tills dess att det skapat en tydlighet i vad som gäller. Peter tror att det finns en risk för att processen av att skapa denna klarhet kan ta ett ganska bra tag eftersom han anser att det är den erfarenhet han har av tidigare, liknande problematik när det kommer till IT-världen. Hugo berättar att AI kommer att möta legala och moraliska frågor precis som all ny teknik mötte förut.

Förutom att det finns en osäkerhet kring lagar och rättigheter var även etiken något som respondenterna tog upp. Dock var detta ett område som respondenterna själva hade betydligt mindre att säga om. Peter gick till och med så pass långt som att säga att det är ett område som han självklart känner till men inget han är tillräckligt insatt i för att uttala sig om. De enda som tog upp exempel på etiska dilemman var David och Hugo som uttryckte att AI-verktyg är endast verktyg i slutändan och därför finns en risk att man använder denna typ av AI i fel syfte.

Coeckelbergh (2019) tog dock upp etik runt just AI i sin artikel. Ett av de etiska problem som han tar upp handlar om hur AI automatiserade processer, vilket är något som kan sluta i att folk som utfört denna typ av uppgifter tidigare kan bli arbetslösa. Detta etiska dilemma är extra intressant när det kommer till just testare eller utvecklare eftersom dessa är med och tar fram den AI som sedan kan komma att ersätta dem. Även om våra respondenter inte var rädda för att det skulle kunna hända är det ändå något som är värt att ha i åtanke för att ha en sorts plan för vad som händer ifall detta skulle ske. För om det slår in skulle man vara i ett läge där utvecklare har arbetat för att ersätta sig själva, något som för just utvecklare gör att dilemmat för användning av AI är lättare att ta på och koppla till sig själva.

Det finns även ett etiskt dilemma i att en AI sparar ner den information som en användare matar in. Detta gör att det finns risk att man matar in saker som kan vara känsliga för den kund som beställer det system som ska utvecklas. Detta kan leda till att informationen i slutändan hamnar i fel händer, vilket skulle leda till att kundens information handhas felaktigt, lagar bryts och liknande. Därför finns det

även en vikt vid att man antingen ser till att de som använder AI verktyg vet vad de får mata in för information till AI eller att man löser på det sätt att informationen inte kan hamna i fel händer.

Haenlein och Kaplan (2019) diskuterar behovet av reglering av AI-system i förhållande till deras påverkan på samhället ur tre perspektiv. Dessa perspektiv anser vi också omfatta AI inom programutveckling på grund av dess potentiella påverkan på minskning av partiskhet, automatisering av uppgifter och etiska överväganden vid algoritmdesign. Författarna tar upp mikroperspektivet som belyser en potentiell snedvridning i AI-system på grund av de data som används för deras träning. Författarna föreslår utvecklingen av allmänt accepterade krav för träning och testning av AI-algoritmer för att undvika sådana felaktigheter. Detta innebär att man måste vara medveten om de potentiella problem som kan uppstå när AI-system tränas på data som är partiska eller ofullständiga. Genom att etablera riktlinjer och normer för hur AI-algoritmer ska tränas och testas kan man försöka minimera dessa problem och säkerställa en mer rättvis och pålitlig användning av AI.

I meso-perspektivet riktas fokus mot AI:s påverkan på sysselsättningen. Med ökningen av AI-teknologier kan det finnas en risk att många tjänstemannajobb försvinner, inklusive programmerare eller testare. För att mildra dessa negativa konsekvenser föreslår författarna att reglering kan vara en lämplig åtgärd. De menar att genom att införa regler kan man försöka hantera och omställa arbetsmarknaden för att minimera de negativa effekterna av arbetslöshet till följd av automatisering och AI. I sista perspektivet som är makroperspektivet diskuteras det potentiella missbruket av AI av stater. Men det är ett perspektiv som är relevant för våran diskussion.

Denna typ av problematik är något som EU försöker att lösa genom den förordning som man la fram den 21 april 2021 (2020/21:FPM109). Dock är det fortfarande inte något som man har tagit beslut om mer än två år efter att förslaget först lades fram. Sedan dess har AI utvecklingen varit enorm och det finns idag ett större behov av att reglera den än vad det någonsin funnits tidigare. Detta gäller både för att folk inte ska kunna använda AI-verktygen på fel sätt. Men även för att man ska möjliggöra att vi ska kunna dra så stor nytta av verktygen som möjligt inom till exempel området för programutveckling.

5.2 AI i framtiden

Resultatet från vår undersökning tyder på att utvecklare och testare tror att man i en stor utsträckning kommer kunna använda AI för att genomföra programmerings-och testuppgifter i framtiden. Alla respondenter tror att AI kommer kunna hjälpa till med enklare uppgifter. Peter berättar i sin intervju att “Enhetstester definitivt, det tror jag att jag skulle kunna slippa skriva när AI blir lite bättre” medan han även tror att AI kommer kunna hjälpa till att lösa enklare uppgifter och snabba på programmeringsprocessen. Att AI i början endast kommer kunna lösa enklare uppgifter är något som Peter inte var ensam med om att tro utan det var en återkommande trend bland de respondenterna som vi intervjuade. Att de tror att det är just denna typen av uppgifter som AI:n kommer kunna lösa i början grundar sig en hel del i den erfarenhet de har av att använda AI sedan tidigare. Simon berättar hur han på sin fritid testat att utnyttja både ChatGPT och GitHub Copilot när han arbetat på hobbyprojekt. Han menar att hans erfarenhet kring detta varit något varierande eftersom han anser att resultaten från ChatGPT kunde variera en hel del och att det framförallt var till att “bygga grunden” för vissa kodsnuttar som det gick att använda. Även Peter och David hade någon form av erfarenhet där de sett Copilot användas och lösa just denna typ av kodningsuppgifter. Dock är den överväldigande tron bland personerna inte att AI någon gång snart kommer kunna lösa koduppgifter eller testa kod fullt ut. Istället tror de att det kommer behöva finnas en människa

som utnyttjar AI:n och fyller i de luckor som finns. Liam menar att AI har en del begränsningar som kommer göra det svårt för den att ta över en kodares uppgifter helt. Adam pratar om hur han tror det kommer behöva vara någon som kan ha en förståelse för helheten och Simon anser att det finns en komplexitet i delar av kod som han tror kan vara svårt för AI-teknik att hålla reda på och förhålla sig till utan att få hjälp. När det kommer till hur testning kommer att påverkas i framtiden nämner Peter att han tror att "...enkla tester kommer AI säkert kunna hjälpa till med men jag tror aldrig att en AI kommer kunna ersätta en riktigt bra testare, en testare som verkligen försöker förstöra min kod". Peter förklarar sitt uttalande med att han har svårt att se att AI ska fungera lika bra som en mänsklig testare när det kommer till kreativiteten som kan krävas för att testa något riktigt bra. Tron kring hur man kommer kunna använda AI inom programutveckling inom en snar framtid grundar sig nog till stor del av den erfarenhet man har runt AI fram till nu. Eftersom ingen förutom Hugo använt AI professionellt kan det skapa en lite mer "tveksam syn" gällande precis hur mycket av kodning och testning som man kommer kunna utnyttja AI verktygen till. Hugo som å andra sidan har jobbat med det redan kunde genom denna arbetserfarenhet ha skapat sig en bättre förståelse gällande exakt vilken kapacitet som AI verktygen besitter. Detta skulle kunna ligga till grund för varför Hugo tror att verktygen kommer kunna användas i större utsträckning än vissa andra respondenter i ett tidigare skede.

Längre fram i tiden tror dock vissa respondenter att AI kommer kunna få en större roll i utvecklingsarbetet. När vi frågar Peter om han tror att AI kommer kunna ta över en utvecklares roll helt svarar han att han inte tror det kan hända under den tiden som han själv fortfarande är yrkesverksam. Adam och Hugo berättar hur de tror att AI med tiden kommer kunna skriva allt större delar av koden men att det känns som att det ändå kommer behöva finnas någon form av person som överser allt.

När det gäller AI i framtiden generellt fokuserar litteraturen (Zhang och Lu, 2021; Haenlein och Kaplan, 2019) på två olika perspektiv. För det första diskuteras tre centrala synpunkter inom AI-utveckling: symbolism, connectionism och behaviorism. Symbolism hjälper datorer att tänka logiskt, connectionism hjälper dem att lära av exempel och behaviorism hjälper dem att anpassa sig och förbättra baserat på belöningar och straff. Enligt (Zhang och Lu, 2021), genom att sammanföra dessa metoder fortsätter AI att utvecklas, vilket gör datorer smartare och mer kapabla att förstå och interagera med världen omkring oss. Dessutom föreslår författarna (Zhang och Lu, 2021) att nästa genombrott inom AI kan leda till att maskiners intelligens överträffar människors. För det andra diskuterar (Haenlein och Kaplan, 2019) behovet av reglering av AI-system i framtiden från tre olika perspektiv: mikroperspektivet, meso-perspektivet och makroperspektivet. Dessa perspektiv betonar olika aspekter av reglering, dessa aspekter inkluderar undvikande av snedvridning i AI-system, hantering av arbetslöshet orsakad av ökad AI användning och förebyggande av AI missbruk från stater.

Respondenterna gentemot litteraturen (Zhang och Lu, 2021; Haenlein och Kaplan, 2019) om AI:s framtid har olika fokus. Respondenterna ger en mer inriktad synpunkt på utvecklarnas och testarnas syn på AI:s praktiska aspekter inom programmering och testning av kod, medan litteraturen om AI generellt tar upp bredare frågor som de teoretiska grunderna för AI-utveckling och behovet av reglering av AI-system.

Å andra sidan, enligt Zhang, Zhu och Su (2023), en tredje generation av AI håller på att utvecklas genom att kombinera de bästa egenskaperna från de två tidigare generationerna. Logisk problemlösning och kunskapsrepresentation från den första generationen samt lärande från data och mönsterigenkänning från den andra generationen. Målet med den tredje

generationen enligt författarna är att skapa en stark AI-teknik som kan återspegla intelligenta mänskliga beteenden. Man kan argumentera för idén som författarna menar att uppkomsten av den tredje generationen är vad som till en viss nivå upplevs redan idag med all den nya AI-tekniken som kommer upp och blir mer populär dag för dag, men även kommer att fullt utvecklas till framtiden. Tittar man vidare på litteraturen (Harman, 2012; Shafiq, et al., 2021; Aljanabi, et al., 2023) kring AI:s framtid just inom programutveckling tar den upp att AI har potentialen att revolutionera programutvecklingen i framtiden genom att främst automatisera uppgifter och snabba upp vissa processer. I framtiden, enligt litteraturen (Harman, 2012; Shafiq, et al., 2021) men även respondenterna, kan vi förvänta oss att AI inom programutveckling och programmering fortsätter att utvecklas och förbättras. Det är möjligt att nya AI-modeller och tekniker kommer att introduceras, vilket gör det möjligt för utvecklare att arbeta mer effektivt och skapa högkvalitativ mjukvara snabbare. Men det är viktigt att komma ihåg att mänsklig inblandning fortfarande är viktig för att fatta avgörande beslut och utvärdera resultatet av AI-applicering. Trots AI:s framsteg är det fortfarande nödvändigt med mänsklig kunskap och bedömning för att fatta viktiga beslut och utvärdera resultatet av AI-applicering. Detta understryker att AI bör ses som ett verktyg som kan komplettera och underlätta mänskligt arbete, snarare än att ersätta människor helt och hållet.

Det är viktigt att komma ihåg att AI inte är perfekt och kan ha brister i hur den presterar. Mänsklig inblandning behövs för att övervaka och korrigera sådana problem. Dessutom är det också nödvändigt att bedöma etiska konsekvenser och potentiella risker med AI-applicering. Precis som vissa av våra respondenter har varit tydliga med riskerna av att använda vissa AI-verktyg utan att veta mer om hur de fungerar, hur datan behandlas och vilka tillåtelser gäller. Reglering av AI-system spelar en viktig roll för att säkerställa att de används på ett sätt som är förenligt med mänskliga värderingar oavsett inom företag eller i samhället som helhet..

5.2.1 Hur kan rollerna komma att förändras

Respondenterna hade en ganska gemensam tro på att utvecklare och testare kommer vara tvungna att anpassa sig och dra nytta av de AI-verktyg som kommer att finnas tillgängliga för att stödja såväl utvecklare som testare i sina processer. Dock skiljde åsikterna sig något. Vissa av våra respondenter tog upp att de tror att yrkesverksamma kommer att behöva anpassa sig på det sätt att de ändrar hur de gör saker för att dra nytta av de potentiella fördelar som AI kan bidra med. Andra tog det ett steg längre i att de tror att vissa av de roller som idag finns i en programutvecklingsprocess kan komma att förändras. Simon och Hugo berättar hur han tror att man inte längre kommer behöva vara lika många i ett team som är med och utvecklar. Hugo tror dock även att det kan finnas potential för framförallt större företag att de istället tar sig an flera uppgifter inom teamen. Simon tror att det kan bli att man slår ihop testare och utvecklare till en roll och har färre personer i ett team. Detta tror han eftersom en hel del av de utvecklare som finns även kan en del om testning och vice versa. Även Adam tror att det kan bli att man slår ihop roller. Han tar dock upp att han tror att det kommer finnas en mer övergripande person som kan lite om varje del. Denna person menar Adam, istället skulle vara bra på att få en AI att genomföra de olika delarna åt honom genom att han endast instruerar dem med olika promptar. Hugo var inne på samma spår, han menar att utvecklare kommer bli problemformulerare och har syfte att översätta mellan människa och maskin istället, och se över den processen i helhet. David berättar hur han tror att teamen som man arbetar med kan bli mindre och att utvecklare utan att kunna programmera även kan komma att behöva utbildas i hur de på bästa möjliga sätt instruerar AI att göra det som en utvecklare vill. Att utvecklare kan komma att behöva utbildas är något som även Liam tar upp. Detta kan komma att bli något företag får göra eftersom det kommer att finnas en övergångsperiod där

utvecklare och testare som redan är färdigutbildade kan komma att behöva lära sig detta för att företagen ska kunna dra maximal nytta av den AI utveckling som sker. ‘

När det kommer till hur de två rollerna kan påverkas finns det dock ingen tidigare forskning som vi lyckats hitta som tar upp just hur rollerna kan komma att förändras eller inte.

6 Slutsats

Studien ämnar att öka kunskapen om användningen av AI inom programutvecklingsprocessen från två olika perspektiv. Det första perspektivet är dagens användning enligt yrkesverksamma, medan det andra perspektivet handlar om hur användningen kan komma att se ut i framtiden enligt yrkesverksamma. För att konkretisera forskningsämnet formulerade vi två forskningsfrågor som vi genom studien nu kan besvara. Den första forskningsfrågan var:

I vilken utsträckning används AI inom programutveckling idag?

Genom studien fann vi att AI inte används i särskilt stor utsträckning inom programutveckling bland svenska företag idag. Bland våra respondenter var det endast två personer som hade använt AI-verktyg i yrkessammanhang inom programutvecklingsprocessen, och endast en av dessa använde det för att skriva kod och utföra tester. Detta är anmärkningsvärt med tanke på att både våra respondenter och den tidigare litteraturen som vi tar upp nämner att det finns verktyg tillgängliga som kan bidra till processen. Givet att litteraturen som vi diskuterade, alltid gav en mer övergripande, bredare och oftast optimistisk syn på fördelarna med användningen av AI inom programutveckling. Dessa verktyg anses ha potential att tillföra mycket till processen, trots de begränsningar som för närvarande kan finnas kring dem. När vi undersökte varför respondenterna känner till verktygen men ändå bara en av dem använder dem, identifierade vi flera faktorer som kan ligga bakom detta.

En faktor som kan vara bidragande är företagens regler när det gäller implementeringen av AI-verktyg i deras arbetsprocesser. Skillnader i dessa regler kan påverka tidsåtgången för olika företag att implementera verktygen. Detta är särskilt relevant med tanke på att de AI-verktyg vi undersöker fortfarande är relativt nya fenomen.

En annan faktor som framkom i studien var de etiska och juridiska frågeställningar som omger AI. Dessa två områden är nära relaterade, eftersom de juridiska frågorna främst beror på osäkerheten kring hur AI-verktygen fungerar och hur de kommer fram till sina resultat. En av utvecklarna som intervjuades nämnde att de faktiskt hade fått instruktioner att inte använda AI-verktyg på grund av denna osäkerhet. Detta visar att företagen vill vara försiktiga och se till att undvika eventuella juridiska konsekvenser. Återigen verkar detta vara en fråga om företagspolicy eftersom vi hade en utvecklare som faktiskt använder AI-verktyg i sitt dagliga arbete redan nu. Vi upptäckte dock att både det etiska problemet med att inte veta hur AI-verktygen fungerar internt och andra etiska frågor diskuteras aktivt.

Den andra forskningsfrågan som vi formulerade handlade om hur yrkesverksamma förväntar sig att användningen av AI kommer att se ut inom programutveckling i framtiden.

Hur förväntas användningen av AI se ut inom programutveckling i framtiden?

Studien visade att det finns en stark tro bland yrkesverksamma att AI kommer att användas inom programutveckling i framtiden. Detta gäller både för programmering och testning.

Majoriteten av respondenterna tror att AI-verktyg kommer att bli ett verktyg som utvecklare och testare använder i sina processer. De tror inte att AI inom den närmsta framtiden kommer att kunna ersätta dessa yrken helt utan att det fortfarande behövs mänsklig styrning av AI. Det framgick dock tydligt bland respondenterna att de ansåg det vara omöjligt att helt utesluta möjligheten att AI i framtiden kan göra detta, särskilt med tanke på den snabba utvecklingen av AI-verktyg.

Respondenterna uttryckte också att de tror att rollerna som programmerare och testare kan behöva anpassas till följd av AI:s framsteg. Alla var överens om att man måste anpassa sig och dra nytta av AI-verktyg för att fortsätta vara relevanta på arbetsmarknaden. Detta kan innebära förändringar i den kunskap som testare och utvecklare besitter. I framtiden kan det vara mer nödvändigt att kunna instruera AI att skriva kod och utföra tester, samtidigt som man behåller en förståelse för hur kod och tester fungerar. Detta skulle kräva en bredare och mer övergripande kunskap för testare och utvecklare.

Emellertid var inte detta den enda förändring som respondenterna förväntar sig när det gäller rollerna. Vissa av respondenterna trodde att rollerna kan komma att förändras genom att slås samman till en roll som ansvarar för flera delar av programutvecklingsprocessen. Det skulle innebära att samma person utför både testning och kodning, eventuellt även andra uppgifter. En annan tanke som framkom var att teamens sammansättning kan förändras enligt vissa respondenter. Tanken som lyftes fram mest var att teamen skulle bli mindre eftersom färre personer skulle behövas för att genomföra samma process. Dock diskuterades även möjligheten att teamen behåller samma storlek men får fler uppgifter att utföra.

Avslutningsvis, slutsatsen av studien ger insikt om användningen av AI inom programutvecklingsprocessen och dess potentiella inverkan på dagens och framtida programutveckling. Resultaten visade enligt yrkesverksamma att AI inte utnyttjas i större omfattning inom programutveckling inom den svenska verksamhetsmiljön, vilket står i kontrast till den optimistiska syn som ofta presenteras i litteraturen. Denna observation har flera viktiga konsekvenser:

För det första ger resultaten en värdefull indikation på de faktiska utmaningar och hinder som AI-teknologin stöter på inom området för programutveckling. Bristen på utbredd användning antyder att det finns en oenighet mellan potentialen som diskuteras teoretiskt och praktisk implementering. Dessa resultat är därmed en påminnelse om att den realistiska inramningen av AI-integration är mer nyanserad än den ofta entusiastiska tonen i litteraturen.

För det andra indikerar resultaten att det finns ett behov av att adressera de faktorer som hindrar utbredningen av AI inom programutveckling. Frågor kring företagspolicy, etik och juridiska frågeställningar samt osäkerhet kring teknologins funktion och implikationer tycks påverka beslutsprocessen för att implementera AI-verktyg. Detta kräver ett närmare samarbete mellan AI-utvecklare, företag och reglerande organisationer för att skapa riktlinjer som balanserar teknologins potential med riskhantering.

Slutsatsen i sin helhet påminner om att dagens låga användning av AI inom programutveckling inte bör ses som ett stopp eller en stagnation, utan snarare som en början på en resa mot en mer välinformerad och balanserad integration av teknologin. Resultaten indikerar att AI-verktyg har potentialen att förbättra programutvecklingsprocessen, men att det finns hinder som företagspolicy, etiska frågor och juridiska aspekter som påverkar implementeringen av dessa verktyg. Framtiden förutspås vara präglad av en större användning av AI-verktyg, vilket kan innebära att rollerna för programmerare och testare behöver anpassas för att dra nytta av AI:s möjligheter. Framtida utveckling av AI bör vara fokuserad på att bemöta de utmaningar som identifierats och bygga lösningar som adresserar de specifika hinder som bromsar adaptionen.

6.1 Vidare forskning

I framtiden kan denna studie utvidgas genom att genomföra liknande forskning med en bredare och större grupp av deltagare för att undersöka om samma resultat kan erhållas. Denna studie utfördes med en begränsad grupp, därför hade den kunnat förstärkas genom att inkludera fler individer för att undersöka om våra resultat är representativa för åsikterna hos programmerare och testare i allmänhet eller om de enbart speglar åsikterna hos våra respondenter. Detta skulle kunna uppnås genom att genomföra ytterligare liknande intervjuer och analysera vad deltagarna har att säga, eller genom att utföra en kvantitativ undersökning för att avgöra om våra resultat överensstämmer med en bredare population.

Det vore också intressant att fördjupa forskningen i vissa av de mer specifika upptäckter som gjordes. Detta skulle kunna innefatta undersökningar av de etiska dilemman som är förknippade med användningen av AI, hur lagstiftningen behöver anpassas för att ta hänsyn till den ökande användningen av AI i samhället eller en djupare utforskning av vilka andra områden inom programutveckling än enbart testning och programmering som kan dra nytta av AI i framtiden.

Genom att utföra sådana ytterligare studier och undersökningar kan vi få en mer omfattande förståelse för de potentiella effekterna av AI-verktyg och teknologier inom programutvecklingsprocessen.

Referenser

Vetenskapliga referenser

Schank, R. (1987). What Is AI, Anyway?. *AI Magazine,* 8(4). ss. 59.
doi: 10.1609/aimag.v8i4.623.

Zhang, C. Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. *Journal of Industrial Information Integration journal,* 23.
doi: https://doi.org/10.1016/j.jii.2021.100224

Schuett, J. (2019). A Legal Definition of AI. *SSRN Electronic Journal*, januari 2019.
doi: 10.2139/ssrn.3453632

Blackwell, A. (2002). What is Programming?. I *14th Workshop of the Psychology of Programming Interest Group*. Brunel University, Storbritannien 18-21 juni 2002, ss. 204-218.

Harman, M. (2012). The Role of Artificial Intelligence in Software Engineering. Diss. University College London.
Harman, M. (2012) The role of Artificial Intelligence in Software Engineering. I *2012 First International Workshop on Realizing AI Synergies in Software Engineering (RAISE)*. 2012, ss. 1-6. doi: 10.1109/RAISE.2012.6227961

Lowry, M. (1992). Software Engineering in the Twenty-First Century. *AI Magazine*, 13(3). s. 71. doi: 10.1609/aimag.v13i3.1012

Ernst, N. A. och Bavota, G. (2021). AI-driven Development Is Here: Should You Worry?. *IEEE Software*, 39(2). ss. 106-110. doi: 10.1109/MS.2021.3133805

Shafiq, S. Mashkoor, A. Mayr-Dorn C. and Egyed A. (2021). A Literature Review of Using Machine Learning in Software Development Life Cycle Stages. *IEEE Access*, 9. ss. 140896-140920. doi: 10.1109/ACCESS.2021.3119746

Adamopoulou, E., Moussiades, L. (2020). An Overview of Chatbot Technology. *Artificial Intelligence Applications and Innovations*, 584. ss. 373–383.
doi: 10.1007/978-3-030-49186-4_31

Jacobsen, D. I. och Andersson, S. (2017) *Hur genomför man undersökningar?* : introduktion till samhällsvetenskapliga metoder. 2 uppl. Lund: Studentlitteratur AB.

Humphrey, W. S. (1989) The software engineering process: definition and scope. *Software engineering notes, 14*(4), ss. 82–83.

P. Bourque, R. Dupuis, A. Abran, J. W. Moore and L. Tripp, "The guide to the Software Engineering Body of Knowledge," *IEEE Software, vol.* 16(6), ss. 35-44, Nov.-Dec. 1999, doi: 10.1109/52.805471.

Fishpool, B. & Fishpool, M. (2020) *Software Development in Practice*. Swindon: BCS Learning & Development Limited

McCarthy, J. (2007) *WHAT IS ARTIFICIAL INTELLIGENCE?* https://www.diochnos.com/about/McCarthyWhatisAI.pdf

Kaplan, A och Haenlein, M. (2019) *A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence*. doi: 10.1177/0008125619864925.

Zhang B, Zhu J, Su H. (2023) *Toward the third generation artificial intelligence*. Sci China Inf Sci, 2023, 66(2): 121101, https://doi.org/10.1007/s11432-021-3449-x

Encyclopedia of Database Systems, chapter (*Language Models*)
Hiemstra, D. (2009). Language Models. In: LIU, L., ÖZSU, M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA.
doi: https://doi.org/10.1007/978-0-387-39940-9_923

Zhang, C och Lu, Y. (2021) *Study on artificial intelligence: The state of the art and future prospects.* doi: https://doi.org/10.1016/j.jii.2021.100224

Aljanabi, M. ., Mohanad Ghazi, Ahmed Hussein Ali, Saad Abas Abed and ChatGpt (2023) *ChatGpt: Open Possibilities*, Iraqi Journal For Computer Science and Mathematics, 4(1), pp. 62–64. doi: 10.52866/20ijcsm.2023.01.01.0018.

Biswas, S. (2023). *Role of ChatGPT in Computer Programming*. Mesopotamian Journal of Computer Science, Vol. (2023), pp 8-16. doi: 10.58496/MJCSC/2023/002.

El Naqa, I., Murphy, M.J. (2015). What Is Machine Learning?. In: El Naqa, I., Li, R., Murphy, M. (eds) Machine Learning in Radiation Oncology. Springer, Cham.
https://doi.org/10.1007/978-3-319-18305-3_1

Icke vetenskapliga referenser

GitHub. (2023, april 30). About GitHub Copilot for Individuals. GitHub Docs. https://docs.github.com/en/copilot/overview-of-github-copilot/about-github-copilot-for-individuals

OpenAI. (2023, maj 01). How GPT-3 and our language models are developed. Help Center. https://help.openai.com/en/articles/7842364-how-chatgpt-and-our-language-models-are-developed

2020/21:FPM109. Förordning om artificiell intelligens. Infrastruktursdepartementet.

Bilagor

Bilaga A - Intervjufrågor

- Kan du bekräfta att du förstår att din personliga information kommer att hållas konfidentiell och endast kommer att användas för syftet med denna studie?
- Är du bekväm med att dina svar kommer att spelas in och användas för forskningsändamål?
- Är du medveten om att du kommer att hållas helt anonym i vår uppsats?
- Har du några tvivel om sekretessen för din personliga information eller hur den kommer att användas i denna studie?
- Har du några frågor eller funderingar om sekretessen för din personliga information som du vill diskutera innan intervjun?
- Kan du berätta om dig själv och din bakgrund inom programutveckling?
- Vilka är dina erfarenheter inom programutveckling?
- Vilka typer av projekt har du arbetat med, inom testning eller programmering?
- Har du upplevt AI som i din yrkesroll?
- På vilket sätt och i vilken utsträckning?
- Vad är din syn på artificiell intelligens och dess användning inom programutveckling?
- Känner du till vad en språkmodell är?
- Känner du till vad ChatGPT är?
- Med hänsyn till ChatGPT och den senaste utvecklingen av liknande AI-verktyg.
- Tror du man kommer kunna använda AI inom programutveckling?
- Varför/Varför inte?
- I vilka områden och på vilka sätt?
- Hur påverkar det ditt yrke men även andra roller inblandade inom området för programutveckling?
- Är AI-system kapabla att ersätta mänskliga roller?
- Vilka roller är det?
- I vilken utsträckning?
- Vilka uppgifter syftar du på att AI kan utföra bättre?
- Vad är din syn på hur verksamheter behöver anpassa sig till den ökande användningen av AI idag?
- Särskilt inom företag som är verksamma inom programutveckling?
- Hur tror du att AI kan bidra till att förbättra programmeringsprocessen/testningsprocessen och resultatet?
- Finns det några risker eller nackdelar med AI inom programutveckling?
- Din ståndpunkt kring AI? Orolig? Optimistisk? Skeptisk?..

Bilaga B - Intervjumallen

Fråga	*Mål*
Intro	
Berätta om studien och ämnet vi vill undersöka	Säkerställ att vi får relevanta svar Skapa en förståelse för vad det är vi undersöker och ser till att intervjun håller sig till rätt ämne
Sekretessfrågor	
Kan du bekräfta att du förstår att din personliga information kommer att hållas konfidentiell och endast kommer att användas för syftet med denna studie?	Säkerställ sekretess samtycke
Är du bekväm med att dina svar kommer att spelas in och användas för forskningsändamål?	Säkerställ sekretess samtycke
Är du medveten om att du kommer att hållas helt anonym i vår uppsats.	Säkerställ sekretess samtycke
Har du några tvivel om sekretessen för din personliga information eller hur den kommer att användas i denna studie?	Säkerställ sekretess samtycke
Har du några frågor eller funderingar om sekretessen för din personliga information som du vill diskutera innan intervjun?	Säkerställ sekretess samtycke
Respondentens Bakgrund	
Kan du berätta om dig själv och din bakgrund inom programutveckling?	Säkerställ att respondenten har relevant kunskap Få förståelse kring deras utbildning och relation till programutveckling
Vilka är dina erfarenheter inom programutveckling? - Vilka typer av projekt har du arbetat med, inom testning eller programmering?	Skapa förståelse för arbetserfarenheter Skapa förståelse för vilken typ av arbetsuppgifter respondenten har
Upplevelse och användning av AI idag	
Har du upplevt AI som i din yrkesroll? - På vilket sätt och i vilken utsträckning?	Erfarenheter av AI i arbetslivet Åsikt om AI i programutveckling idag Kännedom kring språkmodeller och ChatGPT

- Vad är din syn på artificiell intelligens och dess användning inom programutveckling?
- Känner du till vad en språkmodell är?
- Känner du till vad ChatGPT är?

Tankar kring AI i framtiden

Med hänsyn till ChatGPT och den senaste utvecklingen av liknande AI-verktyg.

- Tror du man kommer kunna använda AI inom programutveckling?
- Varför/Varför inte?
- I vilka områden och på vilka sätt?
- Hur påverkar det ditt yrke men även andra roller inblandade inom området för programutveckling?

Skapa en uppfattning om respondenternas åsikter kring potentialen och möjligheterna som AI kan erbjuda inom programutveckling. Även deras åsikter kring AI-system och ifall det finns potential att de kan ersätta en mänsklig roll.

Är AI-system kapabla att ersätta mänskliga roller?

- Vilka roller är det?
- I vilken utsträckning?
- Vilka uppgifter syftar du på att AI kan utföra bättre?

Vad är din syn på hur verksamheter behöver anpassa sig till den ökande användningen av AI idag? Särskilt inom företag som är verksamma inom programutveckling?

Respondentens syn på hur företag inom programutveckling behöver anpassa sig till ökad användning av AI och vilka åtgärder som kan krävas för att dra nytta av teknologin.

Hur tror du att AI kan bidra till att förbättra programmeringsprocessen/testningsprocessen och resultatet?
Finns det några risker eller nackdelar med AI inom programutveckling?

Diskutera hur AI kan förbättra programmerings- och testningsprocessen samt resultatet. Även samtidigt undersöka eventuella risker eller nackdelar med AI inom programutveckling.

Din ståndpunkt kring AI? Orolig? Optimistisk? Skeptisk?..

Få förståelse för respondenternas egna känslor kring AI inom programutveckling och ifall det uttrycks någon oro.

HÖGSKOLAN
I BORÅS

Besöksadress: Allégatan 1 · Postadress: 501 90 Borås · Tfn: 033-435 40 00 · E-post: registrator@hb.se · Webb: www.hb.se