LUNDS UNIVERSITET
Ekonomihögskolan

Institutionen för informatik

DevSecOps – när AI flyttar in

IT-praktikers upplevelser rörande generativ AI:s roll i säkerhetsarbetet inom DevSecOps

Kandidatuppsats 15 hp, kurs SYSK16 i Informationssystem

Författare: Hugo Forsgren

Jacob Hallenborg

Johan Wahlgren

Handledare: Benjamin Weaver

Rättande lärare: Nicklas Holmberg

Umberto Fiaccadori

DevSecOps – när AI flyttar in: IT-praktikers upplevelser rörande generativ AI:s roll i säkerhetsarbetet inom DevSecOps

ENGELSK TITEL: DevSecOps – When AI Moves In: IT-practitioner experiences regarding generative AI's role in security work within DevSecOps

FÖRFATTARE: Hugo Forsgren, Jacob Hallenborg och Johan Wahlgren

UTGIVARE: Institutionen för informatik, Ekonomihögskolan, Lunds universitet

EXAMINATOR: Osama Mansour, Docent

FRAMLAGD: maj, 2026

DOKUMENTTYP: Kandidatuppsats

ANTAL SIDOR: 57

NYCKELORD: Generativ AI, DevSecOps, Affordance-teorin, Human-in-the-loop

SAMMANFATTNING (MAX. 200 ORD):
Integrationen av generativ AI inom DevSecOps medför genomgripande förändringar för IT-praktikers säkerhetsarbete, men befintlig forskning har huvudsakligen belyst teknisk effektivitet snarare än IT-praktikers subjektiva upplevelse. Genom en interpretativ ansats och sex semistrukturerade intervjuer med IT-praktiker analyseras empirin med stöd av Affordance-teorin och human-in-the-loop för att belysa hur IT-praktiker upplever generativ AI:s påverkan på säkerhetsaktiviteter inom DevSecOps. Resultaten visar att upplevelsen är relationell och varierar med utvecklingsfas, kompetensnivå och grad av automatisering för säkerhetsaktiviteten. Generativ AI fungerar som en hävstång för befintlig expertis snarare än ett neutralt produktivitetsverktyg, vilket riskerar att förstärka kompetensasymmetrier mellan seniora och juniora praktiker. Aktualiseringen av teknikens handlingspotential hindras främst av bristande förklarbarhet, vilket tvingar praktikern att kompensera genom iterativ verifiering och erfarenhetsbaserad kontextualisering. Yrkesrollen ompositioneras från kodproducent till strategisk referenspunkt där praktikern utgör en domänspecifik ”ground truth” för AI-systemets utdata. Studien bidrar med ett socioteknisk perspektiv som visar att IT-praktikern inte upplevs ersättas av generativ AI utan omförhandlar sin yrkesroll inom säkerhetsarbetet.

Innehåll

Figurer

Tabeller

1. Inledning

1.1 Bakgrund

Gartner (2026) uppskattar att de globala investeringarna i artificiell intelligens (AI) under 2026 kommer att uppgå till 2 500 miljarder dollar, vilket motsvarar en ökning med 44 procent jämfört med föregående år. Denna omfattande ekonomiska satsning påverkar direkt arbetsprocesserna inom mjukvaruindustrin. Enligt Stack Overflow (2025) använder en majoritet av professionella utvecklare numera AI i sitt dagliga arbete där 84 procent av de 49 000 tillfrågade uppger att de planerar att fortsätta använda AI-verktyg och 69 procent anser att AI-agenter redan har bidragit till ökad produktivitet.

Inom modern mjukvaruutveckling tillämpas generativ AI (GAI) i form av AI-verktyg och AI-agenter ofta benämnda AI-kodassistenter. I teknisk bemärkelse baseras dessa kodassistenter på underliggande modeller som bearbetar textbaserade instruktioner för att generera utdata. AI-kodassistenter kan exempelvis bearbeta en skriven funktionsdefinition för att därefter autonomt producera källkoden (Perry et al. 2023). I juni 2021 lanserades GitHub Copilot som markerar ett tekniskt skifte där AI-kodassistenter agerar som ett utvecklarstöd och läser av kontexten i den befintliga koden för att föreslå nya rader eller kompletta funktioner (Friedman, 2021). Denna tekniska acceleration ställer höga krav på arbetsflöden inom agil mjukvaruutveckling. Enligt Bain & Company (u.å.) präglas traditionella utvecklingscykler av en separation mellan utvecklande enheter och operativa enheter. Detta medför ett beroende av manuella processer vid överlämningar mellan funktionerna. Integrationen av utveckling och operativ förmåga inom samma team har dock möjliggjort en högre grad av automatisering och hastighet i arbetsprocesser (Bain & Company, u.å.). Kombinationen av utvecklings- och driftsfunktioner inom samma team går inom industrin ofta under benämningen DevOps eller DevSecOps. Dessa begrepp saknar dock en enhetlig definition och tolkas på olika sätt i praktiken (Gall & Pigni, 2022).

Cybersäkerhet har idag vuxit till att bli en strategisk ekonomisk prioritet snarare än enbart en teknisk stödfunktion då en genomsnittlig betydande cyberattack beräknas kosta företag cirka 250 000 dollar (World Economic Forum, 2026). Den snabba teknologiska utvecklingen innefattande AI har medfört att hotlandskapet förändrats markant där exempelvis AI-integrerade cyberattacker har ökat med 89 procent mellan 2024 och 2025 (CrowdStrike, 2026). Detta alltmer komplexa säkerhetslandskap medför ett behov av DevSecOps, där säkerhet integreras mer omfattande i kombinerade team mellan utvecklande och operativa funktioner (Mohammed et al. 2025). Övergången till detta arbetssätt beskrivs som transformativ då säkerhet omformas till ett delat ansvar genom hela utvecklingscykeln (Mohammed et al. 2025). DevSecOps utgör därmed en brygga mellan traditionella utvecklingscykler och det befintliga hotlandskapet, vilket möjliggör för organisationer att korta ned leveransintervaller utan att förbise kritiska säkerhetsaspekter (Mohammed et al. 2025).

Införandet av GAI i agil mjukvaruutveckling medför en transformation av arbetsprocesser. McKinsey & Company (2023) visar att utvecklare med hjälp av GAI kan halvera tidsåtgången för att skriva ny kod och färdigställa koddokumentation. Vidare visar studien att även mer komplexa uppgifter, som optimering av befintlig kod, kan genomföras på två tredjedelar av den ursprungliga tiden. I praktiken innebär detta att utvecklarens roll förändras från att

manuellt skriva kodrader till att kontrollera större volymer av genererad kod under korta tidsintervall. När utvecklingstakten och kodvolymerna ökar på detta sätt, ställs nya krav på de infrastrukturer och processer där koden ska testas och integreras.

1.2 Problemområde

Integrationen av AI inom DevSecOps beskrivs i forskning som ett betydande tillskott som har potential att fundamentalt förändra mjukvaruindustrins förmåga till säkerhet, effektivitet och innovation (Pakalapti et al. 2023). Tekniken kan väsentligt förbättra förutsättningarna för hotdetektion, prediktiv analys och automatisering, vilket hjälper organisationer att effektivisera säkerhetsarbetet (Camacho, 2024). Samtidigt kan AI-drivna kodassistenter accelerera själva mjukvaruproduktionen med upp till 55 procent (Alenezi & Akour, 2025). Trots dessa tekniska framsteg och den markant ökade hastigheten i utvecklingsflödet medför tekniken nya utmaningar som hindrar en bredare tillämpning. Forskning visar att generativ AI i agil mjukvaruutveckling introducerar komplikationer i form av falsklarm vid hotidentifiering, samt problem med skalbarhet som en konsekvens av bristande standardisering vid införandet (Mohammed et al. 2025).

För att hantera säkerheten i denna föränderliga och snabba miljö förlitar sig moderna organisationer på DevSecOps. Metodologin bygger i grunden på "shift-left"-principen, vilket innebär att säkerhetsrutiner integreras tidigt och kontinuerligt genom hela utvecklingscykeln istället för att vara en efterkonstruktion (Mohan & Othmane, 2016; Mohammed et al. 2025). I praktiken består dessa integrerade säkerhetsaktiviteter inom DevSecOps av en samverkan mellan automatiserade och mänskliga moment. Medan uppgifter som automatiserad testning och statisk kodgranskning framgångsrikt delegeras till tekniska system, kvarstår centrala uppgifter som hotmodellering och designgranskning som icke-automatiserade moment vilka uteslutande förlitar sig på mänsklig expertis (Rahman & Williams, 2016).

När generativ AI introduceras i detta arbetsflöde utmanas den etablerade balansen, vilket i grunden förändrar IT-praktikerns arbetsmiljö. IT-praktikerns roll omformas från att vara en primär skapare av kod till att i allt högre grad bli en operatör som behöver övervaka och kritiskt granska de stora volymerna av AI-genererade förslag (Alenezi & Akour, 2025; Chen et al. 2021). Detta förändrar IT-praktikerns upplevelse av ansvar och kontroll. Forskning visar exempelvis en kritisk paradox där utvecklare med AI-stöd kan uppleva en falsk trygghet, vilket leder till att de producerar och accepterar mindre säker kod (Perry et al. 2023). Integrationen tvingar fram en ständig justering av tilliten, särskilt eftersom AI-modeller riskerar att både dölja säkerhetsbrister och misstolka den specifika domänkontexten (Chen et al. 2021; Bedoya et al. 2024).

Sammanfattningsvis befinner sig DevSecOps i ett skede där AI visserligen driver hastighet, men samtidigt introducerar sociotekniska komplikationer för säkerhetsarbetet. Den befintliga litteraturen belyser dock primärt de tekniska och organisatoriska aspekterna av denna integration. Fokus ligger ofta på systemens prestanda eller kravet på underliggande organisatorisk mognadsgrad för att tekniken ska fungera (Bedoya et al. 2024). Litteraturen saknar därmed individnära studier om hur IT-praktikerna faktiskt uppfattar och hanterar denna förändring.

För att fullt ut förstå hur generativ AI påverkar det kontinuerliga säkerhetsarbetet räcker det således inte att enbart mäta teknisk hastighet eller effektivitet. Forskningen saknar idag en

kvalitativ, socioteknisk dimension som sätter människan i centrum. För att klargöra under vilka villkor tekniken faktiskt kan användas säkert i agila miljöer, finns det därför ett tydligt behov av att studera den enskilda IT-praktikerns upplevelse av hur integrationen av generativ AI påverkar säkerhetsaktiviteter inom DevSecOps.

1.3 Forskningsfråga

De komplikationer som identifierats i tidigare forskning skapar ett behov av att rikta fokus från teknisk funktionalitet till IT-praktikers perspektiv. De tekniska effekterna kan redogöra för potentiella effektiviseringar, men det saknas kunskap om hur IT-praktiker hanterar GAI som hjälpmedel för säkerhetsaktiviteter inom agil mjukvaruutveckling. Utan en förståelse för hur individen uppfattar och aktualiserar handlingsmöjligheter med tekniken riskerar organisationer att implementera verktyg som negativt påverkar säkerhetsaktiviteter. För att belysa detta forskningsgap och bidra med insikter om de spänningar och erfarenheter som präglar IT-praktikers perspektiv undersöker studien följande forskningsfråga:

Hur upplever IT-praktiker generativ AI:s påverkan på säkerhetsaktiviteter inom DevSecOps?

1.4 Syfte

Syftet med studien är att undersöka IT-praktikers subjektiva upplevelser av hur GAI omformar säkerhetsaktiviteter inom ramen för DevSecOps. Detta sociotekniska perspektiv syftar till att belysa samspelet mellan en teknik med stor potential, och IT-praktikers upplevelser av den. Genom att tillämpa Affordance-teorin syftar studien till att identifiera hur IT-praktiker uppfattar handlingsmöjligheter med tekniken i samspel med en accelererande kodproduktionstakt och komplexa säkerhetskrav. Vidare syftar studien till att förstå IT-praktikers roll inom ramen för human-in-the-loop med särskilt fokus på upplevelser av GAI utifrån hur tekniken är integrerad och används. Studien tillhandahåller kunskap om hur mjukvaruindustrins säkerhetsarbete omformas på individnivå i denna spänning mellan teknik och människan där insikter, förståelse och upptäckter grundas i IT-praktikers upplevelser.

1.5 Avgränsningar

Den genomförda studien avgränsas till säkerhetsaktiviteter inom utvecklingsfasen (Dev) av DevSecOps. Studien avgränsas geografiskt till Sverige och deltagande respondenter är verksamma i roller direkt kopplade till säkerhet inom mjukvaruutvecklingens livscykel. Avgränsningen till utvecklingsfasen motiveras av IT-praktikers direkta interaktion med generativ AI i säkerhetssammanhang. Då agil mjukvaruutveckling befinner sig i en övergångsfas där det råder skiljaktigheter i tolkning av begreppen DevSecOps och DevOps inkluderas respondenter verksamma inom DevOps-kontexter som i praktiken hanterar säkerhet i enlighet med DevSecOps-principer. Rahman och Williams (2016) konstaterar att säkerhet i DevOps är ett alias för DevSecOps. Eftersom studien tillämpar en kvalitativ ansats med semistrukturerade intervjuer är resultaten inte avsedda att generaliseras statistiskt, utan syftar till att bidra med en djupare förståelse för det studerade fenomenet inom den avgränsade kontexten.

2. Litteraturgenomgång

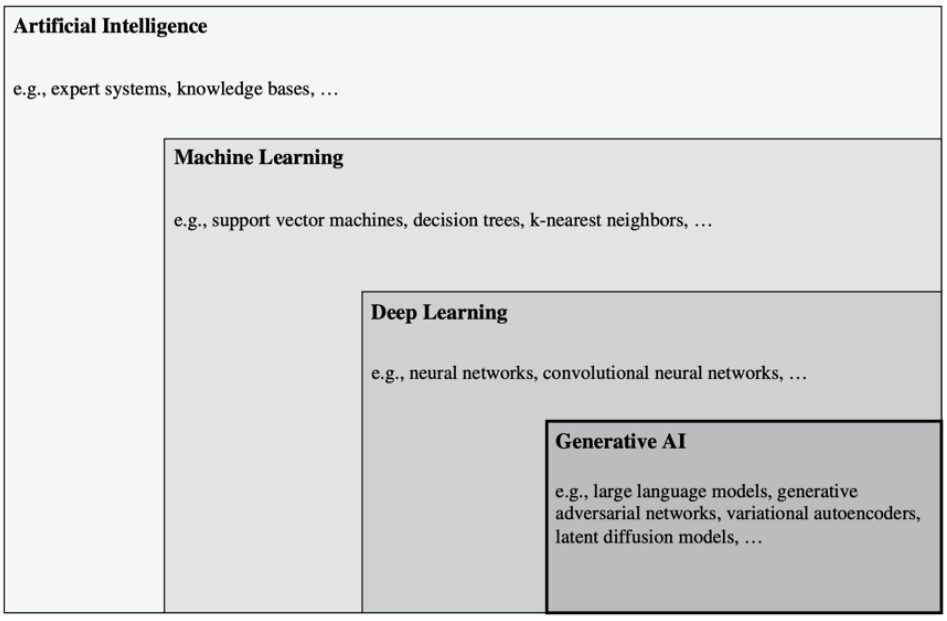
2.1 Artificiell intelligens

2.1.1 Definition artificiell intelligens

Litteraturen visar att begreppet AI officiellt introducerades inom akademin år 1956 under en konferens på Dartmouth College (Haenlein & Kaplan, 2019a; Buchanan, 2005). Trots att AI etablerades som en akademisk disciplin redan vid denna tidpunkt präglades det av en begränsad vetenskaplig utveckling och lågt praktiskt intresse under mer än ett halvt sekel (Haenlein & Kaplan, 2019a). Det är först i modern tid som teknologin gjort sitt definitiva intåg i affärslivet och den allmänna samhällsdebatten (Haenlein & Kaplan, 2019a).

Begreppet AI saknar dock en enhetlig definition i den vetenskapliga litteraturen och har kommit att användas som ett samlingsbegrepp för en rad olika tekniker och förmågor. Banh och Strobel (2023) beskriver AI som ett paraplybegrepp som omfattar olika beräkningsalgoritmer med förmåga att utföra uppgifter som traditionellt förutsätter mänsklig intelligens såsom förståelse av naturligt språk, mönsterigenkänning, beslutsfattande och inlärning från erfarenhet. Haenlein och Kaplan (2019b) definierar i sin tur AI utifrån ett systems förmåga att korrekt tolka externa data, lära sig från dessa och tillämpa denna kunskap för att uppnå specifika mål genom flexibel anpassning. Sammantaget framträder två centrala dimensioner i förståelsen av AI, vilka innefattar dess bredd som teknologiskt paraplybegrepp och dess kapacitet för adaptiv, målinriktad inlärning.

2.1.2 Generativ artificiell intelligens



Figur 2.1: ”Generative AI and other AI concepts” (Banh & Strobel, 2023)

För att förstå termen GAI är det nödvändigt att betrakta teknologin som en del av en hierarkisk struktur inom det bredare fältet AI (se Figur 2.1). Den dominerande formen av AI idag är artificiell snäv intelligens (Artificial Narrow Intelligence, ANI), vilken är specialiserad på specifika uppgifter snarare än mänsklig allmän-intelligens (Kalota, 2024). Inom ANI har fältet rört sig från symbolisk AI som styrs av programmerade fasta regler, till maskininlärning (ML) där algoritmer och tekniker möjliggör för maskiner att lära sig och förbättras på egen hand från data (Kalota, 2024).

Djupinlärning (DL) utgör en förgrening av ML där skillnaden främst ligger i hur data hanteras (Kalota, 2024; Banh & Strobel, 2023). Traditionell maskininlärning kräver ofta att människor i förväg definierar vilka särdrag modellen ska leta efter medan DL-modeller genom artificiella neurala nätverk automatiskt kan identifiera och lyfta fram komplexa mönster ur ostrukturerade data (Kalota, 2024; Banh & Strobel, 2023). Förmågan att hantera högdimensionella data såsom råtext eller bilder genom flera dolda lager är vad som enligt Banh och Strobel (2023) har lagt grunden för dagens mest avancerade applikationer.

Det sista steget i den tekniska progressionen är GAI vilket representerar en vidareutveckling av DL-tekniken. Till skillnad från tidigare nämnda modeller vars mål är att kategorisera befintlig data är GAI-modeller konstruerade för att producera helt nytt innehåll som text, bilder eller kod (Banh & Strobel, 2023). Till följd av ChatGPT:s lansering under 2022 har teknologin fått global uppmärksamhet, där Kalota (2024) belyser hur Large Language Models (LLM) har förmågan att omvandla indata från ett format till ett annat. Genom att tolka och bearbeta specifika instruktioner i en prompt kan systemet därmed generera unika datasekvenser vilket enligt Kalota (2024) innebär att tekniken går från att enbart göra förutsägelser till att aktivt producera nytt innehåll.

Denna förmåga att aktivt producera nytt innehåll, såsom mjukvarukod (Kalota, 2024), har lett till att GAI nu genomgår en snabb och global spridning bland mjukvaruutvecklare (Daniotti et al. 2026). För att kunna undersöka hur IT-praktiker faktiskt upplever denna teknologis påverkan på sitt dagliga säkerhetsarbete, är det nödvändigt att först etablera den kontext i vilken mjukvaruproduktionen sker. Denna kontext utgörs idag primärt av agil mjukvaruutveckling, vilket är ett paradigm vars genomgripande påverkan på mjukvaruindustrin nu driver en implementering tvärs över alla branscher (Gall & Pigni, 2022), samt dess säkerhetsfokuserade vidareutvecklingar (DevSecOps).

2.2 DevSecOps

2.2.1 Definition DevOps

I föreliggande studie används DevOps som en konceptuell utgångspunkt för att förstå övergången till DevSecOps. Gall och Pigni (2022) definierar DevOps som en vision att genom ett kulturellt skifte harmonisera mjukvaruutveckling med den operativa miljön. Vidare påpekar de att genom integration av ansvar för utveckling, kvalitetssäkring och drift inom samma team adresseras de svårigheter traditionella metoder haft med att förena hög kvalitet med snabb leveranshastighet. Trots DevOps centrala roll konstaterar Gall och Pigni (2022) att det saknas en homogen definition av begreppet, vilket gör att termen ofta används synonymt med närliggande tekniska termer. För att strukturera begreppet föreslår författarna en modell bestående av tre huvudområden: kontinuerlig kultur, kontinuerlig automation och kontinuerlig

övervakning, där kontinuiteten innebär att processerna fungerar som integrerade delkomponenter i mjukvarans hela livscykel.

Hemon-Hildgen och Rowe (2022) menar dock att strävan efter en entydig definition ofta bortser från branschens verklighet, då tolkningen i hög grad beror på vilken aktör som uttrycker den. Istället föreslås att DevOps betraktas som en sammansättning av samarbetsprinciper baserade på delad kultur, gemensamma mål och implementering av automatiserade processer. Att just automatisering utgör en central del av DevOps framhålls även av Gall och Pigni (2022), som beskriver det som en kontinuerlig automation med syfte att reducera manuella processer och samtidigt göra dem mer förutsägbara. I praktiken realiseras denna automation ofta genom så kallade CI/CD-pipelines (Continuous Integration/Continuous Delivery), vilket Shahin et al. (2017) definierar som en serie enskilda faser vilka automatiserar överföringen av kod från utveckling till produktionsmiljö. En central utmaning med detta är dock att en hög grad av automatisering förändrar teamets riskperception (Hemon-Hildgen & Rowe, 2022). I detta sammanhang lyfter författarna även kritiska frågor kring integrationen av AI och i vilken utsträckning organisationer kan förlita sig på tekniken utan att förlora den mänskliga kontrollen.

Risken att förlora denna kontroll förvärras av den pressen på hastighet som Zhou et al. (2023) identifierar inom DevOps. Samtidigt som Hemon-Hildgen och Rowe (2022) varnar för hur automation påverkar teamens syn på risker, pekar Zhou et al. (2023) på ett större strukturellt problem. De visar att säkerheten ofta nedprioriteras när den ställs mot kraven på en snabb och agil utveckling. Denna inbyggda friktion gör att säkerhetsarbetet tvingas in i en reaktiv roll i slutet av utvecklingscykeln (Zhou et al. 2023). Enligt författarna blir konsekvensen en minskad tillit inom teamen och en arbetsmiljö där säkerheten marginaliseras.

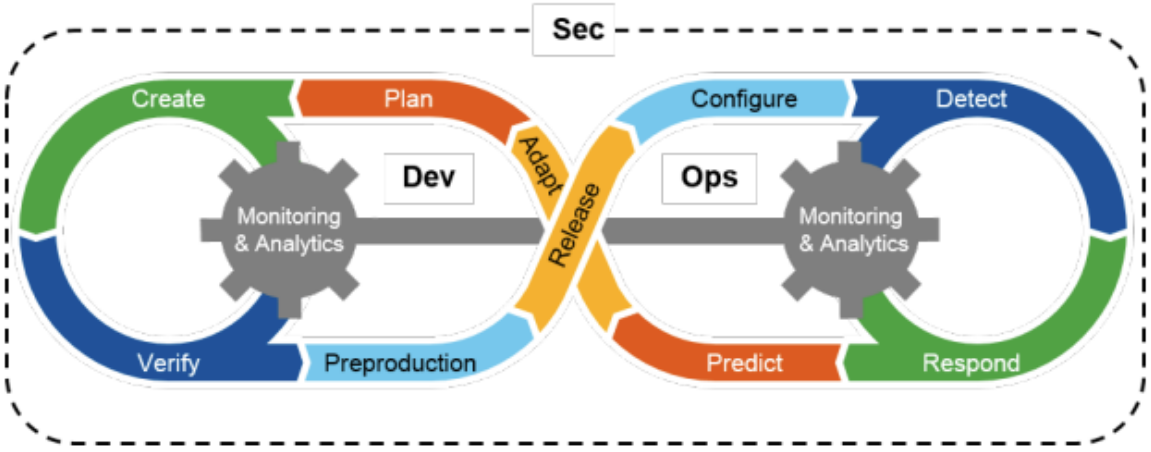
2.2.2 Definition DevSecOps

För att hantera dessa strukturella brister krävs ett systematiskt tillvägagångssätt som integrerar säkerhet med övriga processer (Zhou et al. 2023). Det är ur detta behov konceptet DevSecOps har vuxit fram. Även om det råder generell konsensus om att DevSecOps utgör en förlängning av DevOps, delar de båda koncepten en central problematik vilket är avsaknaden av en allmänt vedertagen definition. Precis som Hemon-Hildgen och Rowe (2022) konstaterar gällande DevOps, påpekar Zhou et al. (2023) att förståelsen av DevSecOps är högst kontextuell. Betydelsen av begreppet varierar i såväl akademi som industri, primärt beroende på organisationens mognadsgrad och den enskilda utövarens profession (Zhou et al. 2023). I en litteraturstudie genomförd av Zhao et al. (2024) framgår det dock att den mest etablerade definitionen inom vetenskaplig litteratur är formulerad av Mohan och Othmane (2016). Författarna menar att DevSecOps ska ses som en nödvändig utvidgning av DevOps-paradigmet vilket innebär att säkerhetsrutiner ska inkorporeras i de existerande processerna genom att aktivt främja samarbete mellan utveckling, drift och säkerhet.

Denna integrering av säkerhetsrutiner refereras centralt till som ”shift-left”-principen, vilket traditionellt betonar identifiering av sårbarheter tidigt i utvecklingsfasen (Mohammed et al. 2025). Denna förståelse breddas dock av Zhou et al. (2023), som argumenterar för att ”shift-left” inte endast är en temporär förskjutning av säkerhetstester, utan innebär en kontinuerlig integrering genom hela mjukvarans livscykel. För att säkerheten ska kunna integreras i hela arbetsflödet utan att hindra agiliteten måste sättet säkerhetstester utförs på förändras. Både Mohammed et al. (2025) och Zhou et al. (2023) belyser att en hög grad av automatisering i pipelinen är avgörande. Zhou et al. (2023) poängterar specifikt att automatiserade

säkerhetsaktiviteter kraftigt reducerar både tidsåtgång och kostnader för felhantering, vilket direkt utmanar och marginaliserar traditionella manuella processer.

För att operationalisera ”shift-left”-principen i praktiken krävs, i enlighet med Zhou et al. (2023), att säkerhetsarbetet integreras och studeras över DevSecOps-faserna. Myrbakken och Colomo-Palacios (2017) påvisar dock i sin litteraturstudie en påtaglig avsaknad av mogna branschmodeller för DevSecOps. Eftersom implementeringen varierar kraftigt utifrån organisationens mognadsgrad och utövarens profession (Zhou et al. 2023) är en universell modell svår att definiera. Trots denna fragmentering framhåller Myrbakken och Colomo-Palacios (2017) att Gartners (2016) modell utgör ett vedertaget undantag, då den nått bred acceptans inom både akademi och industri. Modellen visualiserar DevSecOps som en iterativ och kontinuerlig loop (se Figur 2.2) där agila iterationer i utvecklingen (Dev) övergår i drift (Ops) med ständig övervakning och analys som kärna. Modellens centrala poäng är att säkerheten (Sec) omsluter hela processen, vilket illustrerar hur säkerhetsarbetet ständigt måste integreras utan att skapa friktion i flödet.



Figur 2.2: ”DevSecOps” (Gartner, 2016, p.5)

2.3 Transformation av arbetsprocesser

2.3.1 Kategorisering av säkerhetsaktiviteter i DevSecOps

Att de traditionella manuella processerna marginaliseras innebär emellertid inte att den mänskliga aktören utesluts från arbetsflödet. Rahman och Williams (2016) strukturerar denna arbetsfördelning genom sin kategorisering av automatiserade respektive icke-automatiserade säkerhetsaktiviteter. Författarna identifierar att processer som systematisk kodgranskning, säkerhetstestning och kontinuerlig systemövervakning framgångsrikt delegeras till tekniska system (Rahman & Williams, 2016). Utöver dessa automatiserade processer identifierar Rahman och Williams (2016) en rad övergripande säkerhetsaktiviteter som förblir icke-automatiserade. Utifrån författarnas kartläggning kräver uppgifter såsom hotmodellering, designgranskning och analys av säkerhetskrav en förmåga att identifiera, kategorisera och utvärdera systemövergripande brister och hotaktörer. Dessa är uppgifter som förlitar sig på mänsklig expertis (Rahman & Williams, 2016). DevSecOps vilar således på en tydlig ansvarsuppdelning där tekniska verktyg hanterar verifiering och volym, medan fundamentala säkerhetsbeslut kvarstår som ett mänskligt ansvar (Rahman & Williams, 2016).

Initialt redovisas Rahman och Williams (2016) avgränsade ursprungskategorisering för utvecklingsfasen, tillsammans med de specifika definitioner författarna tillämpar i sin studie (se Tabell 2.1).

Tabell 2.1: Egen tabell innefattande Rahman och Williams (2016) ursprungskategorisering

Säkerhetsaktivitet	Kategori	Definition
Analys av säkerhetskrav (Security Requirement Analysis)	Icke-automatiserad	Identifiering av de säkerhetsförmågor mjukvaran måste besitta för att förhindra obehörig åtkomst och uppfylla kravspecifikationer.
Hotmodellering (Threat Modeling)	Icke-automatiserad	Identifiering, beskrivning och kategorisering av systemets potentiella hot samt de aktörer som är associerade med dessa.
Riskanalys (Risk Analysis)	Icke-automatiserad	Skapande och utvärdering av säkerhetsrelevanta designspecifikationer utifrån systemets riskprofil.
Designgranskning (Design Review)	Icke-automatiserad	Utvärdering av mjukvarans övergripande arkitektur och enskilda moduler för att identifiera potentiella säkerhetsbrister.
Validering av indata (Input Validation)	Icke-automatiserad	Utförande av datavalidering samt avvisning av icke-konform data som går in i eller ut ur mjukvaran.
Isolering av opålitliga indata (Isolation of Untrusted Inputs)	Icke-automatiserad	Identifiering och applicering av säkerhetsåtgärder på icke-verifierade resurser, såsom tredjepartsbibliotek.
Manuella säkerhetstester (Performing Manual Security Tests)	Icke-automatiserad	Utförande av riskbaserade tester genom att simulera en angripare (exempelvis via penetrationstestning) för att bekräfta mjukvarans funktionalitet.
Automatiserad kodgranskning (Automation of Code Review)	Automatiserad	Granskning av källkod och återkoppling till utvecklare med hjälp av statiska analysverktyg.
Automatiserad testning (Automation of Testing)	Automatiserad	Automatiskt utförande av funktionella, integrations- och enhetstester.

2.3.2 Klassificeringsmodell för säkerhetsaktiviteter

För att kunna tillämpa Rahman och Williams (2016) breda ansvarsuppdelning på studiens avgränsade utvecklingsfas (Dev), måste deras aktiviteter placeras i en specifik kontext. Därför integreras Rahman och Williams (2016) kategorisering här med Prates och Pereiras (2024) kartläggning av DevSecOps-verktyg. Eftersom Prates och Pereira (2024) kartlägger specifika säkerhetsaktiviteter direkt mot faserna Gartners (2016) DevSecOps-modell, fungerar deras ramverk som ett filter. Genom denna överlappning kan de aktiviteter hos Rahman och Williams (2016) som tillhör operativ drift (Ops), som exempelvis systemövervakning, systematiskt identifieras och exkluderas. Kvar återstår de säkerhetsmekanismer som präglar den faktiska mjukvaruproduktionen, fördelade över utvecklingsfaserna plan, create, verify och

preproduction (se Tabell 2.2). För att säkerställa att denna kartläggning förblir fokuserad och relevant för studien appliceras ytterligare en strikt avgränsning. Endast de säkerhetsaktiviteter som Prates och Pereira (2024) klassificerar som fundamentala (core) eller viktiga (important) inkluderas, medan sekundära verktyg exkluderas.

Att därefter kunna integrera denna teoretiska uppdelning med Prates och Pereiras (2024) konkreta branschmetoder kräver en noggrann jämförelse av båda källornas arbetsdefinitioner. Även om källorna ofta delar terminologi kan innebörden av ett begrepp skilja sig åt beroende på kontext. Exempelvis definierar Prates och Pereira (2024) termen kodgranskning (Code Review) som en manuell granskning utförd av kollegor, medan Rahman och Williams (2016) använder samma term för att specifikt beskriva en automatiserad process driven av statiska analysverktyg. Genom att korsreferera källornas definitioner kan därmed falska terminologiska matchningar undvikas. Den automatiserade kodgranskningen matchas istället mot Prates och Pereiras (2024) metod Static Application Security Testing (SAST), medan Prates och Pereiras (2024) mänskliga kodgranskningen placeras under icke-automatiserade designgranskning.

Vidare, i de fall teknologins utveckling har förskjutit gränserna sedan Rahman och Williams (2016) studie, görs ett medvetet analytiskt val att prioritera den praktiska kontexten i Prates och Pereiras (2024) kartläggning. Exempelvis matchas dynamiska analysverktyg som Dynamic Application Security Testing (DAST) och Interactive Application Security Testing (IAST) mot Rahman och Williams kategori för automatiserad testning. Detta grundar sig i de arbetsdefinitioner som Prates och Pereira (2024) sammanställt utifrån sin litteraturstudie. Den arbetsdefinition av DAST som författarna tillämpar beskriver metoden som testning av webbapplikationer, vilket de i sin tur kategoriserar som en form av automatiserad säkerhetstestning. Gällande IAST utgår författarna från en arbetsdefinition där metoden explicit beskrivs som en process som kontinuerligt och i realtid övervakar applikationen via en integrerad agent under den funktionella testfasen. Det är just denna agentdrivna och kontinuerliga realtidsanalys som bekräftar processens höga grad av automatisering. Trots att båda verktygen utför simulerade attacker, vilket är en uppgift som enligt Rahman och Williams (2016) uppdelning klassificerades som manuella säkerhetstester, så placeras de därmed i den automatiserade kategorin för att korrekt spegla DevSecOps-branschens nuvarande pipeline-integration.

Denna filtrerade överlappning resulterar i studiens slutgiltiga klassificeringsmodell (Tabell 2.2). Det bör understrykas att denna modell inte agerar som ett verktyg som på egen hand besvarar studiens frågeställning. Istället fungerar den som ett hjälpande analytiskt ramverk. Genom att sammanställa huruvida säkerhetsaktiviteten historiskt hanterats manuellt eller automatiserat och vilken utvecklingsfas den tillhör, tillhandahåller modellen en nödvändig struktur. Denna struktur möjliggör en systematisk prövning av studiens empiri genom att undersöka om IT-praktikernas upplevelser av generativ AI:s påverkan skiftar beroende på om tekniken integreras i en expertuppgift eller i en redan automatiserad process (se Tabell 2.2).

Tabell 2.2: Syntes av författarna, Rahman och Williams (2016), Prates och Pereira (2024) och Gartner (2016)

Säkerhetsaktivitet (Rahman & Williams, 2016)	Kategori	Specifik säkerhetsmetod/verktyg (Prates & Pereira, 2024)	Fas (Gartner)
Analys av säkerhetskrav	Icke-automatiserad	Insamling av säkerhetskrav (Security requirement gathering)	Plan

Hotmodellering	Icke-automatiserad	Hotmodellering (Threat Modelling)	Plan
Riskanalys	Icke-automatiserad	Riskbedömning (Risk assessment)	Plan
Validering av indata	Icke-automatiserad	Säker kodning (secure coding)	Create
Designgranskning	Icke-automatiserad	Kodgranskning (Code review)	Verify
Automatiserad kodgranskning	Automatiserad	SAST (Static Application Security Testing)	Verify
Automatiserad testning	Automatiserad	Automatiserad säkerhetstestning (Automated security testing)	Verify
Automatiserad testning	Automatiserad	DAST (Dynamic Application Security Testing)	Verify
Automatiserad testning	Automatiserad	IAST (Interactive Application Security Testing)	Verify
Manuella säkerhetstester	Icke-automatiserad	Penetrationstestning	Preproduction

2.3.3 AI inom DevSecOps

Med den kontextuella grunden för säkerhetsuppgifterna etablerad så undersöker detta kapitel hur AI integreras i dessa arbetsflöden. Kodassistenter drivna av AI utgör en specialiserad tillämpning av GAI som är integrerad direkt i utvecklarens arbetsmiljö för att erbjuda kodförslag baserat på existerande kod och prompts från användaren (Perry et al. 2023, Daniotti et al. 2026). Verktygen använder sig ofta av maskininlärningsmodeller som OpenAI:s Codex vilken är tränad på massiva dataset innehållande publik källkod från GitHub (Alenezi & Akour 2025; Chen et al. 2021; Perry et al. 2023). Genom att analysera omgivande variabler, funktioner och kommentarer (Corso et al. 2024) kan assistenten autonomt generera komplexa kodsegment, såsom hela funktionskroppar (Daniotti et al. 2026), utifrån enkla instruktioner.

Införandet av assistenterna har medfört betydande produktivitetsvinster där studier visar på en acceleration av kodningsarbetet med upp till 55 % (Alenezi & Akour, 2025). Den vetenskapliga litteraturen är emellertid oenig kring hur denna effektivisering av kodproduktion fördelas över erfarenhetsnivåer. Daniotti et al. (2026) menar att seniora utvecklare drar mer nytta av kodassistenter och ökar sin produktivitet medan juniora utvecklare inte ser några mätbara fördelar trots att de använder verktygen mest frekvent. I motsats till detta menar Cui et al. (2026) samt Hoffmann et al. (2025) att juniora utvecklare ser fler fördelar och högre effektivitet vid användning av kodassistenter. Mjukvaruutvecklare använder dock inte bara GAI för att producera kod utan tenderar att delegera uppgifter som upplevs som minst lustfyllda, såsom skrivande av tester och teknisk dokumentation till assistenten (Sergeyuk et al. 2025).

Utöver dessa generella effektiviseringar i det dagliga utvecklingsarbetet har teknologin även en direkt inverkan på det specifika säkerhetsarbetet. Bedoya et al. (2024) belyser hur integrationen av LLM:er kan effektivisera och automatisera säkerhetsprocesser inom DevSecOps. Författarna lyfter fram att GAI bland annat kan påskynda skapandet av hotmodeller och förbättra precisionen vid identifiering av sårbarheter. Deras studie föreslår att LLM:er kan användas för att skapa attack-försvarsträd, vilket utgör ett viktigt komplement och en extra säkerhetsnivå till traditionella verktyg som SAST och DAST. Detta sparar inte bara tid för säkerhetsanalytiker, utan hjälper även till att upptäcka sårbarheter som konventionella säkerhetsverktyg i en CI/CD-pipeline annars riskerar att missa (Bedoya et al. 2024).

I likhet med Bedoya et al. (2024) breddar Fu et al. (2025) perspektivet genom att undersöka denna integration över samtliga faser i DevSecOps. Författarna konstaterar att verktyg baserade på LLM:er, som Security Copilot, är högst relevanta för just hotmodellering. Utöver hotmodellering utgör hanteringen av mjukvaruberoenden ett tydligt exempel på hur AI rent tekniskt integreras som beslutsstöd i dessa faser. Fu et al. (2025) beskriver hur intelligenta verktyg kontinuerligt kan övervaka externa komponenter och strukturera data. Verktygen ger sedan utvecklare rekommendationer för att prioritera uppdateringar baserat på risknivå och arbetsinsats (Fu et al. 2025). På liknande sätt används modellerna för att säkra det kontinuerliga integrationsflödet genom sårbarhetsprediktion. Genom att analysera historiska data och kodkomplexitet kan systemen förutse om en specifik koduppdatering riskerar att introducera sårbarheter (Fu et al. 2025). Detta ger utvecklingsteamet möjlighet att proaktivt upptäcka och åtgärda brister, såsom felaktig indatavalidering i ett mycket tidigt skede (Fu et al. 2025).

Som en förlängning av denna förmåga att proaktivt förutse hot lyfter Fu et al. (2025) fram områden där AI faktiskt kan ta automatiseringen ytterligare ett steg genom att överbygga begränsningarna hos dagens verktyg. Ett sådant område är enligt författarna "Automated Vulnerability Repair". Medan traditionella statiska analysverktyg är begränsade till att föreslå enkla, små korrigeringar, kan moderna AI-lösningar baserade på djupinlärning rekommendera mer komplexa kodlagningar (Fu et al. 2025). Genom att integrera dessa modeller direkt i utvecklarens arbetsmiljö kan verktygen enligt författarna ge åtgärdsförslag i nästintill realtid. Detta integrerar inte bara säkerhetsarbetet direkt i utvecklingsfasen, utan adresserar också den tidsödande och arbetsintensiva naturen av manuell kodreparation (Fu et al. 2025).

I linje med "Automated Vulnerability Repair" som Fu et al. (2025) beskriver konkretiserar Alenezi och Akour (2025) det praktiska värdet av att integrera AI-verktyg direkt i utvecklingsplattformarna. Genom att dessa verktyg analyserar varje enskild koduppdatering skapas en omedelbar loop av återkoppling som låter team proaktivt fånga upp problem tidigt i processen. Alenezi och Akour (2025) påvisar att organisationer som applicerar sådana verktyg upplever en markant förbättring av säkerheten, vilket specifikt inkluderar en substantiell minskning av sårbarheter samt att tiden för manuella kodgranskningar halveras.

2.3.4 Hastighet kontra säkerhetskvalitet

Trots dessa mätbara framsteg gällande arbetstillfredsställelse och den tidigare nämnda accelerationen av kodningsarbetet (Alenezi & Akour, 2025) belyser litteraturen en kritisk diskrepans mellan den genererade kodens funktionella korrekthet och dess faktiska säkerhet. Eftersom modeller som Codex tränas på publik källkod riskerar de att lära sig och sprida vidare existerande programmeringsfel. Detta resulterar i förslag som kan verka korrekta, men

som i själva verket innehåller dolda säkerhetsbrister (Chen et al. 2021). En central riskfaktor i detta sammanhang är användarens tillit till verktyget. Perry et al. (2023) identifierar här en kritisk paradox i att utvecklare med tillgång till AI-kodassistenter genererar mindre säker kod, men samtidigt är mer benägna att tro att de har löst uppgiften på ett säkert sätt jämfört med utvecklare utan tillgång till verktygen. Detta fenomen skapar en falsk trygghet och ett bristande kritiskt förhållningssätt gällande kodens kvalitet, vilket leder till att användare oftare accepterar AI-genererade förslag utan den noggranna verifiering som krävs för att identifiera dolda sårbarheter (Perry et al. 2023).

Som ett ytterligare lager till denna problematik identifierar Bedoya et al. (2024) flera kritiska utmaningar med LLM:er som direkt påverkar kodens säkerhet. Författarna menar att en betydande risk är att LLM:er ibland misstolkar kontexten och därmed genererar felaktiga eller meningslösa svar. En annan begränsning rör potentiell bias i träningsdata samt de inbyggda innehållsfilter som AI-leverantörerna använder (Bedoya et al. 2024). Dessa filter kan i praktiken hindra verktygen från att presentera fullständiga vägar för hur sårbarheter utnyttjas, vilket försämrar utvecklarens förståelse för den faktiska hotbilden (Bedoya et al. 2024).

2.4 Affordance-teorin

För att förstå varför fenomen som falsk trygghet uppstår och mer övergripande hur IT-praktiker upplever att AI påverkar deras säkerhetsaktiviteter, är det nödvändigt att byta perspektiv från de rent tekniska specifikationerna. Det krävs istället en undersökning av den komplexa relationen mellan systemet och dess användare. Detta förutsätter ett socioteknisk perspektiv där fokus skiftar till hur tekniken faktiskt upplevs och integreras i praktiken.

Begreppet affordance beskriver de handlingsmöjligheter som uppstår i relationen mellan ett subjekt och ett objekt (Valbø, 2021). Termen fick sin fullständiga teoretiska utformning i Gibson (1979) citerat i Valbø (2021). Författaren grundade teorin för att beskriva hur varelser i sin miljö inte främst uppfattar objekts fysiska former, utan snarare vad dessa objekt "erbjuder" (affords) varelserna i termer av handling. En bärande princip i Gibsons (1979) citerat i Valbø (2021) ursprungliga tes är att en affordance existerar oberoende av om den uppfattas eller inte. För att en affordance ska få praktisk betydelse måste den dock uppfattas och aktualiseras av aktören (Gibson, 1979 citerat i Valbø, 2021).

Inom IS-forskning har teorin vidareutvecklats för att förklara interaktionen mellan användare och IT-artefakter (Valbø, 2021). En annan tolkning, ursprungligen från Norman (1988) citerat i Valbø (2021), är att likställa affordances med inbyggda tekniska funktioner och designledtrådar. För att undvika denna begreppsliga sammanblandning följer föreliggande studie Valbøs (2021) rekommendation att uteslutande betrakta affordances som en relationell handlingspotential.

2.4.1 Funktionella handlingsmöjligheter

För att precisera begreppet i en organisatorisk kontext har flera forskare betonat affordances relationella natur (Valbø, 2021). Markus och Silver (2008) konkretiserar denna relationella utgångspunkt genom konceptet funktionella handlingsmöjligheter (functional affordances). Likt Valbø (2021) definierar Markus och Silver (2008) detta strikt som en relation mellan ett tekniskt objekt och en specifik användargrupp, snarare än som en inneboende egenskap hos

teknologin i sig. Detta innebär att vilka målorienterade handlingar som möjliggörs helt och hållet dikteras av den specifika användarens mål och förmågor. Ett och samma system kan erbjuda avgörande handlingsmöjligheter för en expert, men inte erbjuda något alls för en användare som saknar rätt kompetens (Markus & Silver, 2008). Vidare betonar Markus och Silver (2008) att funktionella handlingsmöjligheter uteslutande rör potentiell användning. Denna potential utgör ett nödvändigt villkor för, men garanterar inte, det faktiska användandet.

Medan funktionella handlingsmöjligheter belyser de potentiella handlingar en användare uppfattar, fungerar symboliska uttryck (symbolic expressions) som ett nödvändigt komplement för att förstå hur användaren tolkar systemet (Markus & Silver, 2008). Författarna menar att symboliska uttryck går motsatt riktning och belyser IT-artefaktens kommunikativa möjligheter i relation till en specifik användargrupp. I stället för att enbart fokusera på vad tekniken tillåter användaren att göra, förklarar denna relationella dimension vad IT-artefakten signalerar om systemets, eller dess skapares, underliggande värderingar och intentioner (Markus & Silver, 2008).

2.4.2 Aktualisering av handlingsmöjligheter

Även om funktionella handlingsmöjligheter och symboliska uttryck förklarar vad en användare kan göra och hur de uppfattar systemet, utgör dessa uppfattningar enbart en latent potential fram tills att handlingsmöjligheten aktualiseras. Strong et al. (2014) menar att en uppfattad handlingsmöjlighet identifierar en potentiell funktion, medan det faktiska utfallet och dess struktur först träder fram under själva aktualiseringen. Vidare antyder författarna att specifikt och förväntat utfall från aktualiseringen är ett omedelbart och konkret resultat som aktören uppfattar som användbart för att förverkliga övergripande mål. Redan innan tekniken används agerar individer som målinriktade aktörer och utvärderar vilka utfall de kan uppnå, vilka handlingar de måste vidta, och uppfattar huruvida dessa handlingar leder till önskvärda resultat som bidrar till deras mål (Strong et al. 2014). Själva aktualiseringsprocessen utgör därmed en icke-linjär och individuell resa som upplevs olika av varje person som vidtar dessa målorienterade handlingar (Strong et al. 2014). Författarna påpekar även att när aktörer stöter på begränsningar rörande de egna förmågorna eller systemets egenskaper påverkar det möjligheten för aktualisering.

En specifik systemegenskap som ofta utgör precis en sådan begränsning för IT-praktikerns aktualisering är bristande förklarbarhet. Bauer et al. (2023) undersöker interaktionen med XAI (Explainable AI), vilket innebär AI-system designade för att göra sina bakomliggande beslut transparenta och begripliga för människan (Doshi-Velez & Kim, 2017). Författarna konstaterar att viljan att faktiskt följa och aktualisera ett AI-systems råd är starkt beroende av om XAI-förklaringarna stämmer överens med praktikerns existerande mentala modeller. Om AI-systemet presenterar en lösning som motsäger användarens tidigare övertygelser skapas en kognitiv dissonans (Bauer et al. 2023). Detta leder ofta till att användaren avfärdar rekommendationen och därmed förblir handlingsmöjligheten ej aktualiserad (Bauer et al. 2023). Författarna visar vidare att fenomenet i högsta grad gäller för erfarna experter. Enligt Bauer et al. (2023) har experter en stark tendens att ägna sig åt bekräftelsebias där de gärna tar till sig AI-systemets råd när dessa bekräftar deras egna teorier, men aktivt ignorerar förklaringar som utmanar deras etablerade kunskap.

Bilden av att en bristande samstämmighet mellan människa och AI-system enbart utgör ett hinder nyanseras dock av von Zahn et al. (2025). Genom att applicera ett metakognitivt

perspektiv belyser författarna hur XAI även kan bidra positivt till användningen av systemet. Forskningen visar att när XAI synliggör en diskrepans mellan AI-systemet och användarens logik kan detta få användaren att omvärdera sin egen förmåga (von Zahn et al. 2025). Istället för att enbart leda till kognitiv dissonans resulterar detta ofta i en förbättrad metakognitiv kalibrering, vilket innebär att användarens tidigare övertro på sin egen kompetens minskar (von Zahn et al. 2025).

Processen kring metakognition, det vill säga förmågan att reflektera över och värdera sitt eget tänkande, är högst relevant för hur XAI faktiskt används i praktiken (von Zahn et al. 2025). Författarna påvisar att en mer korrekt insikt om de egna begränsningarna inte bara ökar frekvensen av delegering till AI-systemet utan även förbättrar beslutens övergripande effektivitet. Det är genom metakognitiv kontroll som användaren blir kapabel att objektivt identifiera vilka uppgifter som är för komplexa att hantera manuellt (von Zahn et al. 2025). Genom att XAI hjälper användaren att kalibrera sin tillit ökar benägenheten att överlåta svåra beslut till AI-systemet (von Zahn et al. 2025).

Utmaningarna med kognitiv dissonans och behovet av kalibrering blir särskilt påtagliga inom ramen för det specifika säkerhetsarbetet. Som tidigare nämnts gällande de automatiserade processernas förmågor konstaterar både Bedoya et al. (2024) och Fu et al. (2025) att DevSecOps trots allt förblir djupt beroende av mänskliga experter. För att detta samspel ska fungera understryker Fu et al. (2025) ett kritiskt krav i form av just XAI. Eftersom många avancerade modeller fungerar som obegripliga svarta lådor får säkerhetsoperatörer svårt att bygga tillit till AI-systemens beslut eller felsöka misstag (Fu et al. 2025). Utan XAI begränsas därmed människans möjlighet att applicera sin expertis och effektivt vägleda systemen (Fu et al. 2025).

2.5 Human-in-the-loop

Efter att ha redogjort för hur handlingsmöjligheter uppfattas och aktualiseras rent teoretiskt övergår fokus nu till hur detta omsätts i praktiken. När IT-praktikern väl aktualiserar dessa handlingsmöjligheter transformeras arbetsflödet, men själva upplevelsen av verktyget dikteras i hög grad av hur integrationen och samarbetet struktureras. För att förstå hur detta nya arbetsflöde tar form krävs därför en närmare granskning av det aktiva samspelet mellan människa och teknik.

2.5.1 Komplementär teamprestation och asymmetri

De identifierade utmaningarna vid integration av AI i DevSecOps leder fram till frågan om hur dessa bäst kan hanteras genom ett samspel mellan människa och system. Hemmer et al. (2025) utvecklar denna dynamik och betonar att effektiviteten i human-in-the-loop-metoder bygger på att det finns en informations- och förmågeasymmetri mellan människa och AI-system. Forskarna utgår från att ett samarbete inte vore meningsfullt om båda parter alltid producerade identiska lösningar på ett problem. Eftersom människa och AI tenderar att göra olika typer av fel kan dessa asymmetrier, om de utnyttjas rätt, resultera i mer exakta gemensamma beslut (Hemmer et al. 2025).

Hemmer et al. (2025) bekräftar genom experimentella tester att ett asymmetriskt samspel mellan människa och AI-verktyg leder till förbättrade resultat. Författarna benämner detta

fenomen som komplementär teamprestation och visar att beslutsfelen minskar signifikant när människor och AI samarbetar jämfört med när aktörerna arbetar isolerat. Vidare belyser studien att användarna aktivt integrerade AI-systemets förslag när det bedömdes höja kvaliteten på det slutgiltiga resultatet. Hemmer et al. (2025) poängterar även att användarna behöll sitt förtroende för AI-verktyg trots att de observerade hur dessa genererade felaktiga svar på grundläggande uppgifter.

Utifrån dessa resultat avråder Hemmer et al. (2025) organisationer från att betrakta AI enbart som ett verktyg för total automatisering eller att försöka bygga modeller som bara imiterar mänskliga experter. För att uppnå synergier bör fokus istället ligga på att bygga modeller som utmärker sig på de specifika uppgifter där mänskligt beslutsfattande brister. Ur ett ledningsperspektiv understryker Hemmer et al. (2025) att detta paradigmskifte även kräver att organisationer aktivt investerar i att utveckla och bevara den unika mänskliga kunskap som AI-verktygen inte kan täcka. Därmed måste organisationer strategiskt bygga team där dessa asymmetriska styrkor tillåts komplettera varandra (Hemmer et al. 2025).

Denna teoretiska ståndpunkt kring asymmetri och komplementära styrkor speglas tydligt i den mer praktiskt inriktade litteraturen. Trots den övergripande strävan efter automatisering inom DevSecOps visar Fu et al. (2025) att AI ofta fungerar bäst som ett avancerat beslutsstöd snarare än en helt självständig aktör. Författarna beskriver uttryckligen integreringen av dessa AI-verktyg som en människoassisterad process. Denna slutsats delas även av Bedoya et al. (2024), vilka likt Fu et al. (2025) poängterar att avancerade säkerhetsprocesser förblir djupt beroende av mänskliga experter. Den samlade litteraturen bekräftar därmed Hemmer et al. (2025) tes att det är genom mänsklig översyn och styrning som systemens fulla potential bäst kan realiseras på ett säkert sätt.

Alenezi och Akour (2025) benämner ett exempel där GAI-verktyg inte är en självständig aktör utan främjar IT-praktikerns arbetstillfredsställelse genom att avlasta och komplettera användaren från repetitiva och felbenägna moment knutna till manuella granskningar. Författarna förklarar att framförallt seniora praktiker ges möjligheten att istället fokusera sin tid på mer komplexa och högkvalitativa arbetsuppgifter (Alenezi & Akour, 2025).

2.5.2 Helhetsperspektiv och ”ground truth”

För att förstå hur komplementär teamprestation appliceras i praktiken undersöker Grønsund och Aanestad (2020) hur human-in-the-loop-konfigurationer organisatoriskt uppstår vid införandet av nya AI-system. Medan Hemmer et al. (2025) primärt förklarar varför samarbete är effektivt utifrån kognitiva asymmetrier, breddar Grønsund och Aanestad (2020) perspektivet genom att visa hur detta samspel struktureras som en iterativ arbetsprocess. Deras fallstudie visar att när ett AI-system introduceras för att automatisera analyser i syfte att hantera större datavolymer och öka hastigheten elimineras inte det mänskliga arbetet. Istället omformas det till vad Grønsund och Aanestad (2020) kallar förstärkande arbete. Detta arbete utgörs av en ständig feedbackloop bestående av två ömsesidigt beroende faser. Den första fasen benämns auditing och handlar om att övervaka och utvärdera AI-systemets resultat. Den andra fasen är altering, vilket innebär att justera och förbättra systemet utifrån denna utvärdering (Grønsund & Aanestad, 2020).

Denna process belyser en tydlig överlappning med Hemmer et al. (2025) gällande AI-systemens brister. I likhet med resonemanget hos Hemmer et al. (2025) om att systemen är strikt begränsade av sin träningsdata poängterar Grønsund och Aanestad (2020) att algoritmer

inom AI-system är extremt känsliga för felaktig, ostrukturerad eller manipulerad externa data. Eftersom AI-systemet saknar förmågan att på egen hand avgöra vad som är verkligt och rätt i komplexa situationer krävs ett objektivt referensvärde för att den överhuvudtaget ska kunna lära sig av sina misstag, en så kallad ”ground truth” (Grønsund & Aanestad, 2020). Grønsund och Aanestad (2020) observerar att lösningen på detta problem blir att det traditionella manuella arbetet inte avvecklas, utan omdirigeras till att agera just som denna referenspunkt. Det är genom att kontinuerligt jämföra AI-systemets utdata med den mänskliga sanningen som gapet kan identifieras och systemet kan tränas vidare (Grønsund & Aanestad, 2020).

Denna omdirigering av det mänskliga arbetet blir särskilt påtaglig när man betraktar den tidigare diskuterade framväxten av LLM:er och kodassistenter. Utvecklingen innebär att mjukvaruutvecklarens yrkesroll fundamentalt omformas från att vara en primär skapare av kod till att i allt högre grad fungera som en operatör som övervakar och kritiskt granskar förslag genererade av AI (Alenezi & Akour, 2025, Chen et al. 2021). Behovet av mänsklig expertis kvarstår därmed som en väsentlig komponent för att navigera de tekniska begränsningar som uppstår när verktygen brister i förståelsen för komplex affärslogik och projektövergripande sammanhang (Alenezi & Akour, 2025, Sergeyuk et al. 2025). Eftersom dessa AI-baserade verktyg ofta saknar insikt i den specifika säkerhetskontexten krävs en rigorös mänsklig validering för att förhindra att funktionellt korrekt kod introducerar dolda sårbarheter i systemet (Perry et al. 2023, Corso et al. 2024).

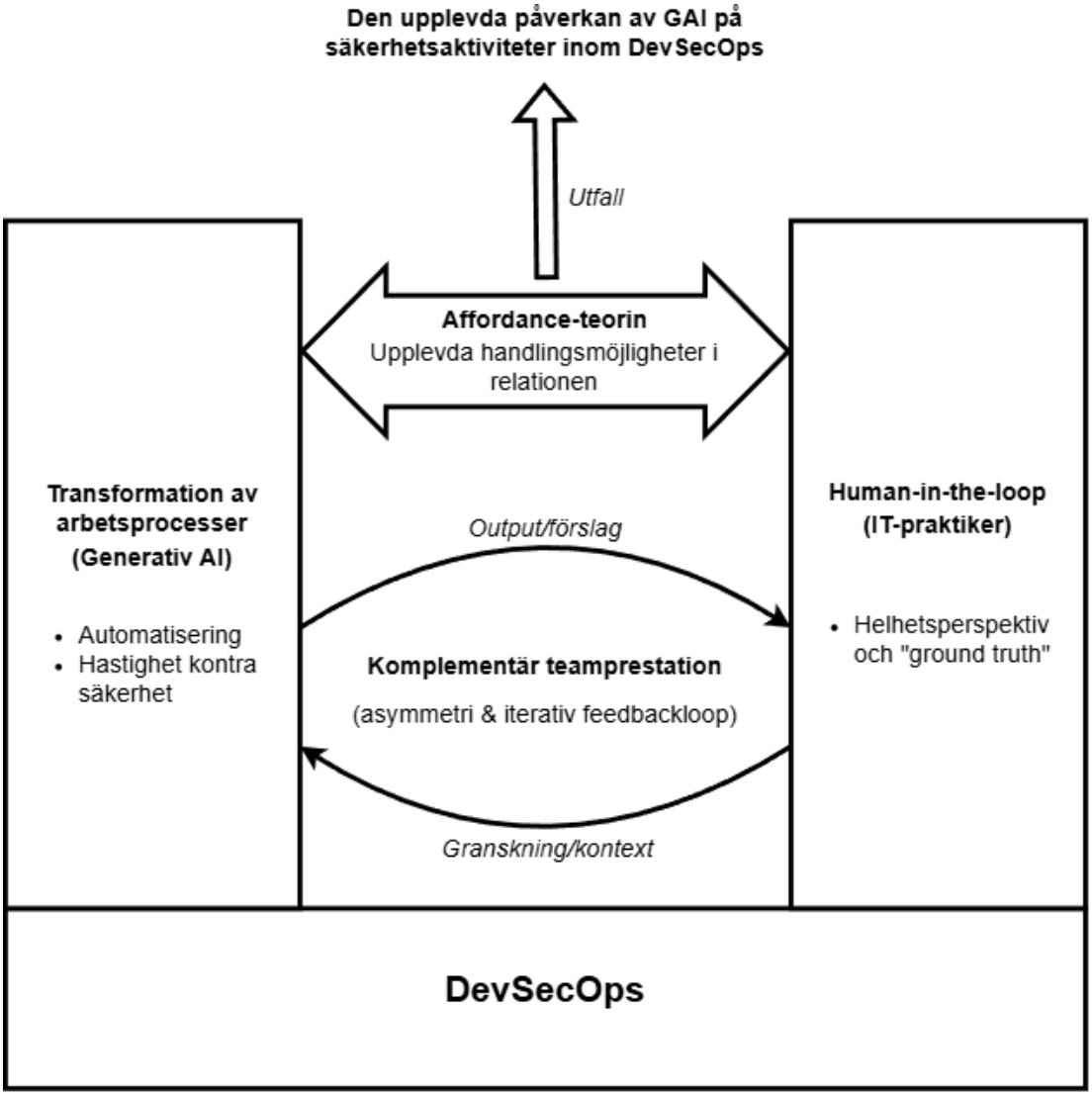
Denna förändrade positionering av människan som en nödvändig granskare speglar direkt de utmaningar kring DevSecOps som diskuterats tidigare. Bedoya et al. (2024) konstaterar att effektiviteten av dessa verktyg är starkt beroende av organisationens mognadsgrad, eftersom tekniken kräver en underliggande mänsklig medvetenhet om cybersäkerhet för att kunna integreras framgångsrikt. Genom att erfarna experter tillhandahåller den kontextuella sanning som AI:n saknar utgör human-in-the-loop-ansatsen därmed inte enbart en passiv kontrollspärr. Det är istället den fundamentala mekanism som krävs för att AI-verktygen överhuvudtaget ska kunna integreras och skapa värde i praktiken (Bedoya et al. 2024, Fu et al. 2025).

Dessa studier landar därmed i en liknande slutsats från två olika vinklar. Human-in-the-loop är inte enbart en kontrollmekanism för att undvika fel, utan en fundamental drivkraft för lärande och värdeskapande. Där Hemmer et al. (2025) ser asymmetrin i beslutsprocessen som nyckeln till överlägsna resultat identifierar Grønsund och Aanestad (2020) människans roll att tillhandahålla en referenspunkt och driva den kontinuerliga feedbackloopen som en central strategisk förmåga. Genom att integrera människans kontextuella förståelse i AI-systemets utveckling bygger organisationen en dynamisk och reflexiv kapacitet som är nödvändig för att anpassa sig till en föränderlig omvärld (Grønsund & Aanestad, 2020).

2.6 Konceptuellt ramverk

Figur 2.3 illustrerar studiens konceptuella ramverk. Modellen bygger på ett sociotekniskt perspektiv för att besvara studiens forskningsfråga och vilar på DevSecOps som den kontextuella bottenplattan. Den vänstra pelaren representerar transformationen av arbetsprocesser och hur generativ AI har påverkat dessa genom att erbjuda en potential för ökad automatiseringsgrad och hastighet, samtidigt som den introducerar friktionen mellan hastighet och säkerhetskvalitet. Den högra pelaren representerar IT-praktikern och dennes roll inom human-in-the-loop, där helhetsperspektivet och funktionen som "ground truth" utgör en nödvändig komponent. Eftersom tekniken i sig aldrig dikterar användandet sammanlänkas

pelarna av Affordance-teorin. Denna teoretiska brygga visualiserar hur tekniska funktioner översätts till uppfattade funktionella handlingsmöjligheter. Samspelet modelleras som en iterativ feedbackloop där den övre pilen representerar AI-systemets output och den nedre pilen representerar praktikerns granskning och tillförande av kontext. Denna iterativa feedbackloop ger upphov till en komplementär teamprestation där asymmetrin mellan människa och AI-verktyg utnyttjas. Det är i denna relationella skärningspunkt som IT-praktikerns slutgiltiga upplevelse formas när maskinens output möter människans yrkesmässiga omdöme och aktualisering.



Figur 2.3: Konceptuellt ramverk

2.7 Litteratursammanfattning

Litteraturen etablerar att DevSecOps omfattar både automatiserade flöden och icke-automatiserade processer vilka är beroende av mänsklig expertis (Rahman & Williams, 2016). Den aktuella forskningen visar att generativ AI bär på en potential att effektivisera båda dessa områden, från automatiserad sårbarhetshantering till avancerad hotmodellering (Bedoya et al. 2024; Fu et al. 2025; Alenezi & Akour, 2025). Denna potential till accelererad kodproduktion

åtföljs dock av en kritisk diskrepans mellan kodens funktionella korrekthet och dess faktiska säkerhet, då AI-genererade förslag riskerar att innehålla dolda säkerhetsbrister (Chen et al. 2021) och samtidigt skapa en falsk trygghet hos användaren (Perry et al. 2023). Enligt Affordance-teorin är denna potential dock avhängig användaren då IT-praktikern inledningsvis måste uppfatta handlingsmöjligheterna i relation till AI-verktygen (Gibson, 1979 citerat i Valbø, 2021). Vilka handlingsmöjligheter som faktiskt uppfattas skiljer sig mellan olika användargrupper. Detta beror på att handlingsmöjligheterna är funktionella och därmed styrs av användarens specifika mål och förmåga (Markus & Silver, 2008), vilket synliggörs i effektivitetsskillnaderna mellan juniora och seniora praktiker (Daniotti et al. 2026; Cui et al. 2026; Hoffmann et al. 2025). Slutligen måste den uppfattade handlingsmöjligheten aktualiseras i praktiken (Strong et al. 2014). Litteraturen visar här att aktualiseringen påverkas av användarens kognitiva processer och AI-systemets förklarbarhet (XAI) (Bauer et al. 2023; von Zahn et al. 2025; Fu et al. 2025), samt av teknikens inneboende begränsningar såsom misstolkning av kontext och bias i träningsdata (Bedoya et al. 2024). Avgörande är även IT-praktikerns uppfattning om huruvida det förväntade utfallet bidrar till de övergripande målen (Strong et al. 2014). Om systemets utdata utmanar en erfaren praktikers etablerade kunskap skapas en kognitiv dissonans vilket ofta leder till att handlingsmöjligheten förblir icke aktualiserad (Bauer et al. 2023). För att hantera dessa kognitiva barriärer och möjliggöra ett framgångsrikt nyttjande av teknologin förutsätts ett strukturerat samarbete baserat på komplementär teamprestation (Hemmer et al. 2025). Detta samarbete drivs av den asymmetri i förmåga som råder mellan AI-system och människa (Hemmer et al. 2025) där IT-praktikern agerar som en objektiv referenspunkt genom förstärkande arbete (Grønsund & Aanestad, 2020) för att kompensera för verktygens bristande förståelse för komplex affärslogik och specifik säkerhetskontext (Sergeyuk et al. 2025; Corso et al. 2024).

Det saknas emellertid kvalitativa studier som undersöker hur IT-praktiker upplever denna påverkan på säkerhetsaktiviteter. Tidigare forskning har primärt utvärderat teknisk effektivitet. Det existerar därmed en kunskapslucka kring hur praktikerna uppfattar och agerar i denna human-in-the-loop-konfiguration. Föreliggande studie ämnar fylla detta gap genom att flytta fokus till den sociotekniska upplevelsen.

Studiens undersökningsramverk bryter ner problematiken i tre interagerande delar vilka appliceras genom studiens klassificeringsmodell. Klassificeringsmodellen agerar som ett kontextuellt filter för att sortera empirin utifrån säkerhetsaktivitetens utvecklingsfas och historiska automatiseringsgrad (Rahman & Williams, 2016; Prates & Pereira, 2024; Gartner, 2016). På denna strukturerade bas appliceras sedan de teoretiska linserna. Transformation av arbetsprocesser etablerar den tekniska kontexten för varje aktivitet. Teorin om affordance används därefter för att granska hur uppfattningen av systemet skiljer sig åt beroende på aktivitetens karaktär (Markus & Silver, 2008; Strong et al. 2014). Slutligen utgör human-in-the-loop den tredje delen för att belysa hur själva arbetsflödet formar IT-praktikerns upplevelse (Hemmer et al. 2025; Grønsund & Aanestad, 2020). Litteraturen visar att utfallet av denna interaktion är högst varierande. När samarbetet resulterar i en komplementär teamprestation där den artificiella intelligensen avlastar praktikern från repetitiva granskningar skapas utrymme för komplext expertarbete (Hemmer et al. 2025; Alenezi & Akour, 2025). Detta specifika samarbetsläge genererar en bevisat positiv upplevelse i form av direkt ökad arbetstillfredsställelse (Alenezi & Akour, 2025). Denna integrerade ansats möjliggör en systematisk prövning av huruvida IT-praktikernas upplevelser varierar beroende på var och hur tekniken integreras.

Integrationen av generativ AI utgör därmed ett tydligt socioteknisk fenomen. Ett helhetsperspektiv som inkluderar teknikens funktioner, användarens uppfattning och det resulterande samarbetet är helt avgörande för att besvara hur IT-praktiker upplever säkerhetsarbetets utveckling.

2.8 Undersökningsramverk

Undersökningsramverket sammanfattar studiens teoretiska utgångspunkter och hur dessa operationaliseras i den empiriska undersökningen. Ramverket är strukturerat i tre huvudområden som tillsammans belyser hur generativ AI påverkar IT-praktikers arbete inom DevSecOps: transformation av arbetsprocesser, Affordance-teorin och human-in-the-loop. Varje huvudområde bryts ner i ett antal teman med tillhörande konstrukt, vilka i sin tur operationaliseras till aspekter som kan undersökas empiriskt genom intervjuer. På så sätt knyts den teoretiska litteraturen samman med studiens datainsamling. Det konceptuella ramverket (se Figur 2.3) förtydligar ytterligare samspelet mellan huvudområden och teman.

Tabell 2.3: Undersökningsramverk

Huvudområde	Tema	Konstrukt	Operationalisering	Litteratur
Transformation av arbetsprocesser	Säkerhetsaktiviteter	Arbetsfördelning: Automatiserade och icke automatiserade säkerhetsaktiviteter	Identifiering av säkerhetsaktiviteter i DevSecOps	Rahman & Williams, 2016; Prates & Pereira, 2024; Gartner, 2016; Perry et al. 2023; Daniotti et al. 2026; Alenezi & Akour, 2025; Chen et al. 2021; Corso et al. 2024; Cui et al. 2026; Hoffmann et al. 2025; Sergeyuk et al. 2025; Bedoya et al. 2024; Fu et al. 2025.
	AI inom DevSecOps	Teknisk integration	Användande av AI-verktyg i yrkesrollen	
	Hastighet kontra säkerhetskvalitet	Produktivitetsparadox	Upplevda effekter vid kodproduktion ställt mot upplevd osäkerhet och kontroll	
Affordance-teorin	Funktionella handlingsmöjligheter	Relationell uppfattning av möjligheter med GAI	IT-praktikerns upplevelse av GAI:s potential utifrån unik kompetens	Valbø, 2021; Markus & Silver, 2008; Strong et al. 2014; Bauer et al. 2023; Doshi-Velez & Kim, 2017;
	Aktualisering av handlingsmöjligheter	Aktualiseringsprocessen av GAI	Upplevda möjligheter eller hinder i förhållande till aktualisering	

				von Zahn et al. 2025; Bedoya et al. 2024; Fu et al. 2025.
Human-in-the-loop	Komplementär teamprestation och asymmetri	Samarbetet mellan GAI och IT-praktikern	Upplevelsen av GAI som verktyg integrerat i en arbetsprocess	Hemmer et al. 2025; Fu et al. 2025; Bedoya et al. 2024; Alenezi & Akour, 2025; Grønsund & Aanestad, 2020; Chen et al. 2021; Sergeyuk et al. 2025; Perry et al. 2023; Corso et al. 2024.
	Helhetsperspektiv och ”ground truth”	Ansvarsfördelning vid nyttjandet av GAI	Upplevelsen av GAI:s begränsningar som verktyg i en arbetsprocess	

3. Metod

3.1 Analys av litteraturinsamling

Den genomförda studien inleds med en litteraturstudie av insamlat litteraturmaterial i syfte att etablera ett teoretiskt fundament inom ämnet. Vid insamling av litteratur nyttjades Oates et al. (2022) tillvägagångssätt där insamlingen av litteratur delas in i olika aktiviteter i syfte att skapa struktur och ordning i slutprodukten. Oates et al. (2022) delar upp processen i följande aktiviteter: ”searching”, ”obtaining”, ”assessing”, ”reading”, ”critically evaluating” och “writing a critical review”.

Den initiala fasen (”Searching”) påbörjades med identifiering av breda teman och intresseområden genom en diskussion mellan uppsatsförfattarna. Oates et al. (2022) föreslår att de breda temana ska definieras i meningar som i sin tur delas ned i delkoncept. Vi väljer att kalla denna initiala process för konceptualisering och kategorisering. Processen resulterar i en tabell som porträtterar centrala teman, intresseområden och delkoncept som vidare kan undersökas (se Tabell 3.1).

Tabell 3.1: Initial sökning: konceptualisering och kategorisering

Breda teman	Intresseområde	Delkoncept
Generativ AI i agil mjukvaruutveckling	Hur har generativ AI och LLM:er förändrat och integrerats i den agila mjukvaruutvecklingen?	Generativ AI, LLM, AI-kodassistenter, Kodproduktion, Kodgenerering
Säkerhetsarbete inom DevSecOps	Identifiering av säkerhetsaktiviteter och deras integration i olika utvecklingscykler.	DevSecOps, DevOps, säkerhetsarbete, säkerhetsaktiviteter, shift-left, CI/CD-security
Socioteknisk interaktion	Förståelse för samspelet mellan IT-praktikern och tekniken med fokus på yrkeskontext och upplevelser av en pågående förändring	Human-in-the-loop, transformation i mjukvaruutveckling, sociotekniska perspektiv, människa-teknik
Aktuellt nyttjande av AI	I vilken utsträckning nyttjas generativ AI idag?	Handlingsmöjligheter, Automatisering, Beslutsstöd, tillit till AI, riskhantering

Den första aktiviteten “searching” följs upp av “obtaining” med syftet att nå värdefull information i källorna (Oates et al. 2022). De plattformar som har nyttjats för insamling av litteratur är Finn (LUBsearch), Google Scholar och AIS eLibrary. Vissa av de källor som har noterats intressanta finns bakom betalväggar och med studiens omfattning har dessa valts bort om de inte funnits tillgängliga vid något bibliotek vid Lunds universitet. Aktiviteterna ”searching” och ”obtaining” sker dynamiskt och iterativt där de initiala sökningarna med breda teman utmynnar i en mängd litteratur som sedan sorteras. Mer specifika teman och delkoncept växer fram och sökord omformuleras till söksträngar som nyttjas på litteraturplattformarna.

Efter att potentiella källor identifierats via sökströngar i de valda databaserna genomfördes aktiviteten ”assessing” av källornas trovärdighet. Oates et al. (2022) understryker vikten av att kritiskt granska utgivaren samt vid vetenskapliga tidskrifter analysera tidskriftens syfte och anseende. Författarna tillhandahåller ett ramverk med kontrollfrågor för att validera källans reliabilitet. I enlighet med dessa riktlinjer och med målet att inkludera material från ledande publikationer inom informationssystem, har studien prioriterat källor publicerade i tidskrifter listade av Association for Information Systems (2025), även om andra källor förekommer. De källor som inte återfinns bland de listade tidskrifterna har valts utifrån deras höga antal citeringar och att de i allra flesta fall är ”peer reviewed”. Därutöver har urvalet kompletterats med välrenommerade tidskrifter från andra vetenskapliga discipliner i de fall de bidragit med perspektiv som är väsentliga för studiens specifika kontext. Detta urval av källor har säkerställt vetenskapligt granskad forskning med högt anseende.

Givet studiens tidsfrist och det omfattande antalet sökresultat tillämpas ett urval vid läsning av litteratur. Aktiviteten ”reading” har tillämpat en akademisk lästeknik som förespråkas av Oates et al. (2022). Författarna menar att akademiskt läsande är annorlunda och bör inledas i abstrakten för att sedan gå till slutsatser i syfte att finna litteraturens relevans för genomförd studie. Urvalet av lästa källor är baserat på vad som ansågs relevant för studiens omfattning och ett strategiskt stickprov av relevanta källor har genomförts.

Innan skrivprocessen börjar sker aktiviteten ”critically evaluating” där det beslutas ifall källan är relevant för studien (Oates et al. 2022). Aktivitetens resultat blir en lista med de användbara källorna, en aktivitet som Oates et al. (2022) benämner som ”recording”. Källorna samlas även i en litteraturmatris som kopplar samman respektive källa till relevant tema.

Slutligen påbörjas sista aktiviteten i skapandet av litteraturgenomgången, ”writing a critical review” (Oates et al. 2022). Litteraturmatrisen fungerar som utgångspunkt för reflektion över potentiella forskningsgap och här identifieras vår forskningsfråga då vi inte finner litteratur som belyser ämnet på det sätt som vi finner intressant. Det finns mycket litteratur som samspelar med forskningsfrågan, men vi finner inget under vår sökprocess som är likt denna studies specifika kontext.

3.2 Metodval

3.2.1 Forskningsfilosofi

Den genomförda studien antar en interpretativ ansats vilket innebär att studiens fokus är att förstå den sociala kontexten av hur teknik tolkas av människor (Oates et al. 2022). Den interpretativa ansatsen syftar till att utforska och förklara hur olika faktorer samspelar i en social miljö, till skillnad från den positivistiska ansatsen som ser möjligheten att genomföra studier helt objektivt (Oates et al. 2022). Med studiens angivna bakgrund, syfte och forskningsfråga tydliggörs ett intresse i hur IT-praktiker upplever en förändring som konsekvens av en innovativ teknik i deras vardagliga yrkeskontext. I enlighet med den interpretativa ansatsen undersöks alltså en upplevelse där forskningens syfte är att identifiera och förstå interaktionen som skapas mellan människan, tekniken och kontexten. Syftet är inte att finna sanningen inom området, utan att adressera individers subjektiva verkligheter rörande ämnet där flera uppfattningar bildas och diskuteras (Oates et al. 2022).

Studien tillämpar Affordance-teorin som analytisk lins, vilken utgår från hur en aktör i sin miljö uppfattar ett objekts handlingsmöjligheter. För studien blir den interpretativa ansatsen relevant för att nyttja teorin på rätt sätt då syftet är att undersöka uppfattningen som åligger IT-praktikern med generativ AI i det dagliga säkerhetsarbetet. Handlingsmöjligheterna är relationella och påverkas av IT-praktikern och dess kontext vilket medför att den interpretativa ansatsen tillåter oss att identifiera varför vissa säkerhetsaktiviteter upplevs annorlunda som konsekvens av generativ AI för olika IT-praktiker och dess olika erfarenheter. Human-in-the-loop appliceras för att förstå hur själva upplevelsen av verktyget formas av hur det praktiska samspelet struktureras efter att handlingsmöjligheterna har aktualiserats.

Den interpretativa ansatsen medför att de genomförande individerna av en interpretativ studie inte är helt neutrala, något som Oates et al. (2022) benämner som reflexivitet. En interpretativ positionering medför därför att respondenternas svar diskuteras och tolkas utifrån författarnas tidigare erfarenheter och kompetenser inom generativ AI, DevSecOps och det sociotekniska perspektivet.

3.2.2 Kvalitativ forskningsmetod

För att besvara forskningsfrågan ”*Hur upplever IT-praktiker generativ AI:s påverkan på säkerhetsaktiviteter inom DevSecOps?*” med en interpretativ ansats krävs personliga svar av IT-praktiker i DevSecOps-kontexten där generativ AI nyttjas. Det är även av vikt att den insamlade datans ursprung är IT-praktiker med kompetens före och efter generativ AI introduceras. Syftet med den kvalitativa forskningsmetoden är att fungera som en vägledning i hur man förstår människan i sin sociala och kulturella kontext till skillnad från den kvantitativa forskningsmetoden som härstammar från naturvetenskapen ofta med syftet att analysera numeriska och matematiska modeller (Myers och Avison, 2002). Oates et al. (2022) redogör att den kvalitativa forskningsmetoden inte nödvändigtvis behöver associeras med en interpretativ ansats, men i vår studie blir den kvalitativa metoden en nödvändig vägledning av den interpretativa ansatsen. I enlighet med redogörelsen ovan kan man motivera valet då de upplevelser som studien undersöker inte kan klassificeras som numeriska, statistiska eller som svar på en hypotes. Upplevelsen av generativ AI:s påverkan vid säkerhetsaktiviteter inom DevSecOps-kontexten är inte en konstant, utan en individuell reflektion. Den kvalitativa ansatsen tillför insamling av relevanta individers uppfattning samt analys, jämförelse och reflektion av den insamlade datan.

3.3 Datainsamling

3.3.1 Semistrukturerade intervjuer

Genomförandet av intervjuer tillåter respondenterna att dela med sig av sina erfarenheter och uppfattningar i en miljö där målet är att utforska snarare än att fastställa ett enhetligt korrekt svar (Oates et al. 2022). En semistrukturerad intervju karaktäriseras av en flexibel interaktion där intervjuaren utgår från förutbestämda teman, men agerar dynamiskt för att skapa en samtalsliknande miljö (Oates et al. 2022). Detta ger respondenten utrymme att fritt belysa områden som denne finner relevanta för sin yrkeskontext. För att omvandla dessa svar till värdefulla data krävs ett förhållningssätt där respondenternas uttalanden betraktas som

narrativ snarare än absoluta fakta då dessa sanningar produceras och formas i intervjustunden (Schultze & Avital 2010).

Schultze och Avital (2010) betonar vikten av ett strukturerat ramverk för att vägleda deltagaren, vilket understödjer dennes förmåga att artikulera och tolka sina erfarenheter. För att generera data med tillräckligt djup tillämpas i denna studie inslag av tekniken ”laddering interviews” (Schultze & Avital 2010). Tekniken utgör en form av semistrukturerad intervju som genom en sociokognitiv lins syftar till att ta fram innehållet i respondentens upplevelser på ett djupgående plan. Syftet är inte bara att förstå vad respondenten uttrycker, utan även varför dessa uppfattningar har formats (Schultze & Avital 2010). I praktiken innebär detta ett iterativt användande av följdfrågor som på ett respektfullt sätt ifrågasätter ytliga beskrivningar för att nå underliggande logik.

Valet av den semistrukturerade intervjun motiveras av behovet att balansera den teoretiska bakgrunden med respondentens individuella reflektioner. Studiens tillämpade teorier kräver en öppen dialog där svar växer fram i en dialog, snarare än att besvara strukturerade frågor.

3.3.2 Urval av respondenter

I syfte att motivera studiens urval av respondenter kan det vara relevant att redogöra en ideal sammansättning av respondenter. Ett idealt urval givet obegränsade resurser hade omfattat en bredare sammansättning av respondenter från en stor mängd olika organisationer och med en mer omfattande spridning i yrkesroller inom DevSecOps. I syfte att navigera studiens omfattning och tillse värdefull insamlad data har Oates et al (2022) ramverk för att identifiera och därefter selektera respondenter nyttjats.

Det faktiska urvalet av respondenter har skett genom ett målstyrt urval vilket innebär att respondenterna väljs ut baserat på deras relevans för studien (Oates et al. 2022). Inom ramen för det målstyrda urvalet har även ett kriteriebaserat urval genomförts där utvalda respondenter säkerställs uppfylla vissa kriterier (Oates et al. 2022). För studien har följande kriterier selekterats:

- Respondenten ska vara en IT-praktiker som i sin yrkesroll berör säkerhet inom mjukvaruutvecklingens livscykel.
- Respondenten ska vara verksam i en kontext som utnämns eller associeras med DevSecOps.
- Respondenterna ska representera olika yrkesroller med olika erfarenhetsnivåer.
- Respondenten ska ha varit yrkesverksam innan och efter generativ AI:s större spridning.

Målstyrda urval kombineras ofta med flera tekniker (Bryman, 2012). För att på ett tidsbesparande sätt få tillgång till respondenter har de valts ut från författarnas professionella kontaktnät. Detta kan tolkas som ett bekvämlighetsurval där respondenterna valts ut då de varit lättillgängliga för författarna (Oates et al. 2022). Dock är det viktigt att understryka att respondenterna i första hand har valts utifrån kriterierna ovan och inte utifrån personlig närhet. Författarna hade ingen tidigare direkt relation till respondenterna före studien. Nyttjandet av de egna professionella nätverken har således fungerat som en strategi för att effektivisera urvalet.

Vid initial kontakt med potentiella respondenter gavs en kort presentation via sms. För de individer som bekräftade att de uppfyllde studiens fastställda kriterier följde en mer fördjupad genomgång av kontexten. Respondenterna tillhandahölls en utförlig beskrivning av studiens syfte, forskningsfråga samt de generella teman och delkoncept som utgör intervjuns teoretiska ramverk. Detta gjordes i syfte att ge respondenterna möjlighet att förbereda sig inför inbokad intervju. Som en del i initiala kontakten redogörs även anonymitet som princip för respondenten.

Studien omfattar sex respondenter som återfinns i tabell 3.2. Vid genomförandet av en kvalitativ studie kan det vara svårt att veta vad som är ett tillräckligt antal respondenter (Bryman, 2012). Bryman (2012) understryker vikten av att tillse tillräckligt insamlad data, men fortfarande samla in data till en kvantitet som gör att den kan analyseras utförligt. För den genomförda studien skedde bokning av intervjuer löpande vilket gjorde att vi kunde sluta tillföra respondenter när vi ansåg insamlade data vara tillräcklig.

Tabell 3.2: Genomförda intervjuer

Respondent	Roll	Intervjudatum	Längd av intervju (Minut:sekund)
R1	Systemutvecklare, team lead	20-04-2026	22:48
R2	Systemutvecklare, backend	20-04-2026	19:49
R3	Systemutvecklare, tech lead, full stack	21-04-2026	32:19
R4	Senior mjukvaruutvecklare	21-04-2026	19:08
R5	Security risk och compliance officer	22-04-2026	26:19
R6	Product manager och cyber security coordinator	23-04-2026	15:40

3.3.3 Intervjuguide

Till skillnad från strukturerade intervjuer är intervjuguiden för en semistrukturerad intervju ett vägledande hjälpmedel för att hålla intervjuerna på rätt teman för studien (Bryman, 2012). Frågorna behöver inte ske i en strikt ordning och ska ge utrymme för omformulering och individuella tolkningar (Bryman, 2012). Bryman (2012) understryker att frågor som inte är med i guiden fortfarande kan ställas och appliceras om intervjuaren upptäcker något intressant utifrån respondentens svar. Vid utformandet av intervjuguiden är det viktigt att följa forskningsområdet och ifrågasätta vad för data som krävs för att besvara studiens forskningsfråga (Bryman, 2012). Vidare bör även respondentens perception av intervjuaren belysas då ålder, kön, etnicitet och upplevd professionalism påverkar hur respondenten besvarar frågor (Oates et al. 2022).

Utöver att skapa en intervjuguide som följer studiens huvudämnen är det viktigt att ställa frågor i en ordning som är logisk och skapar ett bra flöde i intervjun (Bryman, 2012). Intervjuguiden är skapad utifrån undersökningsramverket. Intervjuguiden refererar sedan tillbaka till de litteraturavsnitt som berör ämnet i syfte att ge läsaren en tydlig röd tråd i utformandet. Inledningsvis undersöks respondentens yrkesroll för att etablera en kontext. Därefter fokuseras samtalet på förändrade arbetsprocesser i IT-praktikerns vardag. Konsekvenserna av GAI medför sedan olika handlingsmöjligheter som sedan exploreras. Slutligen fokuserar intervjuguiden på samspelet mellan IT-praktiker och GAI och tillåter respondenten att addera övriga reflektioner rörande ämnet. Således styrks nyttjandet av induktiva inslag i en deduktiv ansats.

Vid formulering av frågor har stor vikt lagts på att använda ett språk som är begripligt för respondenten samt undvikit ledande frågor som kan vrida svaret (Bryman, 2012). Då terminologin kan variera mellan olika organisationer samt utvecklingsteam har förtydliganden ibland krävts för att upprätthålla en värdefull diskussion. Nedan återfinns intervjuguiden som har nyttjats som hjälpmedel för den genomförda studien (Se Tabell 3.3).

Tabell 3.3: Intervjuguide

Huvudområde	Intervjufråga	Följdfråga	Litteraturreferens
Inledning	Vad är din yrkesroll och vad är din anknytning till DevSecOps?	I hur stor del av ditt dagliga arbete använder du generativ artificiell intelligens?	1.1 Bakgrund, 1.5 Avgränsningar, 3.3.2 Urval av respondenter;
Transformation av arbetsprocesser	Vilka säkerhetsaktiviteter jobbar du med?	Om du tänker på dina huvudsakliga säkerhetsaktiviteter. Vilka har varit automatiserade och icke-automatiserade?	2.2.2 Definition DevSecOps (Shift-left), 2.3.1 Kategorisering av säkerhetsaktiviteter i DevSecOps, 2.3.2 Klassificeringsmodell av säkerhetsaktiviteter;
	I vilka säkerhetsaktiviteter har du märkt förändring som konsekvens av generativ artificiell intelligens?		2.3.3 AI inom DevSecOps;
	Hur upplever du att generativ artificiell intelligens har förändrat din yrkesroll idag?	Vilken del av ditt yrke har förändrats mest respektive minst som konsekvens av detta?	2.3.3 AI inom DevSecOps;
	Upplever du att ditt arbete utförs snabbare med hjälp av generativ artificiell intelligens?		2.3.4 Hastighet kontra säkerhetskvalitet;
Affordance-teorin	Vad kan generativ artificiell intelligens erbjuda dig idag som inte funnits tidigare?		2.4.1 Funktionella handlingsmöjligheter;

	Vilka nya möjligheter eller hinder upplever du att generativ artificiell intelligens har skapat för dig rörande ditt säkerhetsarbete?		2.4.2 Aktualisering av handlingsmöjligheter;
	Upplever du att du litar olika mycket på generativ artificiell intelligens för redan automatiserade säkerhetsaktiviteter jämfört med en expertuppgift?		2.4.1 Funktionella handlingsmöjligheter, 2.4.2 Aktualisering av handlingsmöjligheter;
	Finns det säkerhetsaktiviteter som generativ artificiell intelligens är kapabel till att göra där du ändå väljer bort teknologin?		2.4.2 Aktualisering av handlingsmöjligheter;
Human-in-the-loop	Upplever du att ditt arbete har förbättrats av generativ artificiell intelligens?		2.5.1 Komplementär teamprestation och asymmetri;
	Upplever du att integration av generativ artificiell intelligens har lett till att du fångar upp fler säkerhetsbrister?		2.5.1 Komplementär teamprestation och asymmetri;
	Finns det tillfällen där du känner att generativ artificiell intelligens skapar mer arbete?		2.5.2 Helhetsperspektivet och “ground truth”;
Avslutande	Vi har nu pratat om hur AI förändrar dina arbetsflöden, vilka nya möjligheter tekniken skapar och hur ert samarbete ser ut i praktiken. Är det något du skulle vilja tillägga på ämnet?		2.6 Konceptuellt ramverk;

3.3.4 Genomförande av intervjuer

Inför genomförandet av en intervju är det viktigt för intervjuaren att vara förberedd (Oates et al. 2022). Förberedelserna bör utöver intervjuguiden också innebära att samla bakgrundsinformation inom ämne, men även rörande respondenten (Oates et al. 2022). Oates et al. (2022) föreslår också att en bra intervju även bör vara beprövad på en testperson vilket utfördes för studien. Intervjuguiden testades på en bekant till en av författarna där frågor justerades något för att passa sammanhanget. Intervjuerna genomfördes digitalt via Zoom eller Google Meet där respondenten visste att tidsåtgången skulle vara ca 20–40 minuter. Innan intervjun påbörjades informerades respondenten återigen om studiens syfte, inspelning och hur datan kommer lagras och presenteras i slutgiltiga rapporten.

3.4 Dataanalys

3.4.1 Transkribering

För att genomföra en grundlig analys av vad som har sagts under intervjun är det en fördel att spela in den (Bryman, 2012). Inspelningen har flera positiva aspekter, bland annat kan man avlasta intervjuarens minne genom att möjliggöra återspelning och analys av intervjun i efterhand för att mer exakt kunna analysera och behandla det som har sagts (Bryman, 2012). Det finns flera olika varianter av inspelning, och för denna studie har ljudinspelning nyttjats. En ljudinspelning gör även att andra forskare kan kontrollera och analysera vad som har sagts under intervjun via appendix (Oates et.al, 2022). Transkribering av intervjuer är tidkrävande (Oates et.al, 2022), men med en ljudinspelning kan man använda sig av ”voice to text” hjälpmedel för transkribering vilket sparar betydande mängder tid (Bryman, 2012).

I denna studie har transkribering genomförts med hjälp av verktyget Sunet Scribe som baseras på AI-modellen Whisper. För att säkerställa efterlevnad av etiska riktlinjer, GDPR och tillse anonymitet av respondenterna har tjänsten nyttjats via Sunets slutna infrastruktur vilket garanterar att datan inte exponeras för externa molntjänster. Efter den automatiserade transkriberingen har materialet kontrollerats manuellt genom att författarna lyssnat igenom ljudupptagningarna. Detta gjordes för att säkerställa korrekthet samt för att korrigera eventuella felaktigheter eller missförstånd som uppstått i den automatiserade processen.

3.4.2 Deduktiv tematisk analys

Oates et al. (2022) menar att kvalitativa undersökningar ofta kritiseras för att sakna information om dess dataanalys. En anledning till detta kan vara att kvalitativ dataanalys inte har några strikta regler eller tillvägagångssätt som fungerar på ett generellt plan (Oates et al. 2022). Innan datan är redo att analyseras krävs det ett förberedande arbete där insamlade data fördelas i teman och mönster relevanta för studiens forskningsfråga och huvudområden (Oates et al. 2022). Detta tillvägagångssätt benämns tematisk analys och innebär just att svar från intervjuerna kategoriseras i olika teman (Oates et al. 2022). Ett tema kan ha ursprung från olika källor såsom från insamlade data och litteratur, men det kan även härstamma ur ämnen som sammankopplas med forskningsfrågan (Bryman, 2012). För den genomförda studien har teman ursprung i insamlad litteratur, vilket refereras till som en deduktiv ansats (Oates et al. 2022). Oates et al. (2022) belyser vikten av att inte fastna för mycket i en utvald teori och litteraturens teman då de finns en risk att missa teman som återfinns i datan från intervjuerna. Således har även denna studie induktiva inslag vilket betyder att teman i efterhand kan ha anpassats utifrån respondenternas svar (Oates et al. 2022). Detta i syfte att skapa en studie som är sammanhängande och följer en röd tråd mellan insamlad litteratur och intervjusvaren.

3.4.3 Kodning av insamlad data

Den deduktiva tematiska analysen inleddes med en kodning baserat på de teman som etablerats i studiens undersökningsramverk (se Tabell 2.3). För att säkerställa en hög grad av reliabilitet genomfördes kodningen i samråd mellan författarna manuellt där allt transkriberat material har genomgåtts. Genom att tolka meningsenheterna tillsammans skapades en gemensam förståelse för hur respondenters svar relaterar till studiens teoretiska struktur. Under processen sker en kontinuerlig diskussion kring varje enskild kodning och i de fall

oenighet uppstod rörande tolkning hanterades detta genom att återgå till litteraturavsnittet som kodningen refererat till och justera därefter. Detta säkerställer att kodningen förblir teoretiskt förankrad. Nedan återfinns teman och respektive förkortning (se Tabell 3.4).

Tabell 3.4: Kodade teman och förkortningar

Teman	Förkortning
Säkerhetsaktiviteter	SÄK
AI inom DevSecOps	AI-DSO
Hastighet kontra säkerhetskvalitet	H-SK
Funktionella handlingsmöjligheter	FHM
Aktualisering av handlingsmöjligheter	AHM
Komplementär teamprestation och asymmetri	KTA
Helhetsperspektiv och ”ground truth”	H-GT
Generell information	GEN
Irrelevant information	IRR

I syfte att konkretisera hur kodning av det transkriberade materialet har genomförts presenteras ett exempel nedan på hur ett sammanhängande citat har kodats (se Tabell 3.5). För att ytterligare konkretisera visar Tabell 3.6 en nedbrytning av meningsenheter och hur varje enskild meningsenhet har analyserats och motiverats en specifik tematisk koppling. Ett svar kan således innehålla flera analytiska dimensioner.

Tabell 3.5: Exempel på kodning

Respondentens svar	Tematisk koppling
“Det blir beroende på systemet man arbetar med. Jag arbetar med ganska interna system, så det är inte så mycket utåtriktat. Den säkerhet jag hanterar handlar om föråldrade eller felaktiga dependencies, eller om vi råkar ha något utåtriktat tillgängligt som inte borde vara det, exempelvis access till en databas. Om man skulle generera grunden för en databasuppsättning med AI skulle jag personligen vara väldigt noga med att kontrollera att allt är korrekt låst. Jag skulle inte lita på att AI:n sköter det åt mig ännu tror jag.”	SÄK, FHM, AHM

Tabell 3.6: Kodning av meningsenheter

Meningsenhet	Tematisk koppling	Motivering
“Det blir beroende på systemet man arbetar med. Jag arbetar med ganska interna system, så det är inte så mycket utåtriktat. Den säkerhet jag hanterar handlar om föråldrade eller felaktiga dependencies, eller om vi råkar ha något utåtriktat tillgängligt som inte borde vara det, exempelvis access till en databas.”	SÄK	Meningsenheten identifierar och beskriver de specifika säkerhetsaktiviteter som utgår från respondentens nuvarande yrkeskontext.

"Om man skulle generera grunden för en databasuppsättning med AI skulle jag personligen vara väldigt noga med att kontrollera att allt är korrekt låst."	FHM	Här uttrycker respondenten en uppfattad funktionell handlingsmöjlighet. En insikt om vad tekniken kan bidra med.
"Jag skulle inte lita på att AI:n sköter det åt mig ännu tror jag."	AHM	Denna del av citatet kodas till aktualisering av handlingsmöjligheter då den belyser ett hinder som påverkar aktualiserandet av en handling.

3.5 Forskningskvalitet

I en kvalitativ forskningsstudie är reliabilitet och validitet kriterier som kan användas för att motivera och etablera en hög forskningskvalitet (Bryman, 2012). Inom den kvalitativa forskningen kan forskningskvalitet vara svårare att analysera då resultaten som studien medför påverkas av många olika faktorer (Bryman, 2012) och därför introduceras ytterligare kriterier för att etablera en hög transparens.

3.5.1 Reliabilitet

God reliabilitet innebär en studie som kan skapa konsekventa resultat, även vid replikerande av genomförandet (Bryman, 2012). Bryman (2012) understryker att reliabilitet är ett begrepp som ofta associeras mer med kvantitativ forskning då begreppet kan styrka kvalitén på utförda mätningar gjorda i studien. Den genomförda studien är interpretativ där målet inte är replikerbarhet utan resultat som uppfattas trovärdiga ska grundas i transparens. Replikerbarhet är mer associerat med den kvantitativa forskningsmetoden (Oates et al. 2022). Inom interpretativismen ser forskare annorlunda på replikerbarhet då resultatet inte är en faktisk sanning, utan en social konstruktion (Oates et al. 2022). För den genomförda studien har reliabilitetsaspekten fokuserats på genom konsekventa intervjuer och dokumentation genom forskningsprocessen. Alla de intervjuer som har genomförts har nyttjat samma intervjuguide som baseras på samma undersökningsramverk och litteraturmaterial i enlighet med en deduktiv ansats. Respondenterna har delgivits samma information före genomförandet och således beror inte skiljaktigheter i resultatet på informationsasymmetri. De inspelade intervjuerna har även transkriberats, kodats och analyserats på ett konsekvent sätt. Bryman (2012) benämner redogörelsen ovan som extern reliabilitet som skiljer sig från intern reliabilitet. Intern reliabilitet berör hur de olika författarna av studien tolkar de svar som uppstår vid intervjuerna (Bryman, 2012). Hög intern reliabilitet innebär således ett konsensus i uppfattningen av svar. För den genomförda studien har detta inneburit att resultat och diskussion skrivits efter diskussion om olika uppfattningar mellan författarna.

3.5.2 Validitet

Validitet berör forskningskvalitet genom studiens integritet i huruvida slutsatserna som görs är välunderbyggda (Bryman, 2012). Validitet likt reliabilitet kan redogöras intern och externt (Oates et al. 2022). Intern validitet berör hur korrekta och verklighetstrogna resultaten för en

studie är (Oates et al. 2022). I en interpretativ ansats ser man inte ett korrekt svar från respondenten, utan man ser olika uppfattningar av individers verklighet begrundat i respondentens sociala omgivning (Oates et al, 2022). Inom den kvalitativa forskningsmetoden ses intern validitet som hur väl de teoretiska aspekterna samspelar med forskarnas observationer (Bryman, 2012). Den insamlade datan för studien har kopplats till det teoretiska klassificeringsramverket och undersökningsramverk vilket stärker kopplingen mellan observation och litteratur.

Extern validitet berör huruvida observationerna för studien kan generaliseras utanför forskningskontexten där respondenternas generella täckning belyses (Oates et al. 2022). Inom interpretativismen ser man respondenten som unik och accepterar att svar från andra respondenter troligtvis inte kommer matcha studiens observationer (Oates et al. 2022). Begreppet objektivitet lyfts även av Oates et al (2022) i den interpretativa ansatsen vilket innebär att genomförda studien kommer att påverkas av forskaren då undersökning sker genom social interaktion. För den genomförda studien bör extern validitet bedömas av läsarens nytta av studiens applicerat i dennes egen kontext.

3.5.3 Tillförlitlighet, bekräftelsebarhet, pålitlighet, trovärdighet och överförbarhet

Ovan beskrivet så bedöms forskningskvalitet traditionellt från begreppen reliabilitet och validitet (Bryman, 2012). Oates et al (2022) menar dock att kvalitén av interpretativa studier kräver andra kriterier. Redogjort ovan finner ofta interpretativa studier problematik i att besluta forskningskvalitet baserat på mätbara dimensioner vilket grundar sig i interpretativismens syn på verkligheten som en konstrukt av individers sociala omgivning (Oates et al. 2022). För den genomförda studien blir det problematiskt att analysera IT-praktikers upplevelser som en objektiv sanning. Som komplement till traditionella begrepp för forskningskvalitet introducerar Lincoln och Guba (1985 citerat i Oates et al. 2022) en sammansättning kriterier som är mer anpassade för tolkning av forskningskvalitet i interpretativa ansatser. Kriterierna är: tillförlitlighet, bekräftelsebarhet, pålitlighet, trovärdighet och överförbarhet.

Tillförlitlighet berör hur mycket man kan lita på den genomförda forskningen och dess resultat (Oates et al. 2022). Bekräftelsebarhet innebär en reflektion av studiens resultat i förhållande till insamlad data och respondenternas erfarenheter (Oates et al. 2022). För att säkerställa hög bekräftelsebarhet har studien nyttjat ett konceptuellt ramverk i kombination med en ”laddering”-teknik vid intervjuerna (Schultze & Avital, 2010). I resultatdelen redogörs även direkta citat från respondenterna vilket bekräftar härkomst ur insamlade data.

Pålitlighet berör hur väl en studies process är dokumenterad och hur lätt det är för andra att följa studiens process från insamlad litteratur till insamlade data och slutligen dataanalys (Oates et al. 2022). Studien har ett utförligt metodavsnitt där varje steg från initiala tankar, urval av respondenter och genomförandet redovisas öppet. Trovärdigheten av studien belyser hur väl forskningsobjektet, alltså IT-praktikerns upplevelser är korrekt identifierade och beskrivna i resultatet (Oates et al. 2022). Detta styrks även av ”laddering”-metoden vid genomförandet av intervjuerna för att säkerställa och undersöka respondentens svar mer djupgående.

Slutligen, överförbarhet som innebär i vilken grad studiens resultat kan ses generellt relevanta i andra sammanhang eller för andra organisationer (Oates et al. 2022). Interpretativa studier tenderar till att bekräfta överförbarhet genom att innefatta detaljerade beskrivningar av

begreppstolkningar och litteraturmaterial (Oates et al. 2022). För denna studie är litteraturgenomgången extensiv och förklarar begrepp och teorier på ett sätt som för studien kan vara något extensivt, men med syftet att låta läsaren avgöra dess överförbarhet för sin egen situation och kontext.

3.6 Forskningsetik

Behandlingen av de individer som deltar i eller påverkas av studien utgör en av de mest centrala aspekterna av forskningsetiken (Oates et al. 2022). Samtliga deltagare har i enlighet med Oates et al. (2022) behandlats med ärlighet, respekt och hänsyn till deras integritet. Studiens etiska förhållningssätt vilar på grundläggande rättigheter introducerade av Oates et al. (2022). De är följande: rätten att avstå deltagande, rätten att när som helst dra sig ur, rätten till informerat samtycke samt rätten till anonymitet och konfidentialitet.

Deltagandet har genomgående varit frivilligt och ett informerat samtycke har säkrats genom att respondenterna delgivits en utförlig beskrivning av studiens syfte, det förväntade bidraget och hur deras data behandlas. Samtycke för ljudupptagning har även bekräftats muntligt vid varje intervjutillfälle innan inspelning påbörjats. För att upprätthålla anonymitet har både enskilda respondenter och deras respektive organisationer exkluderats och i de fall specifika företagsnamn eller andra identifierbara aspekter nämnts har även dessa exkluderats vid transkriberingen.

Konfidentialitet har upprätthållits genom inspelning och transkribering lokalt med verktyget Sunet Scribe. Detta har säkerställt att ingen känslig information exponerats för externa molntjänster. Endast anonymiserade data har behandlats i molntjänster i enlighet med GDPR och data har endast samlats in om den ansetts nödvändig för studien.

3.7 Metodreflektion

I syfte att bidra med en hög grad av transparens redogörs en kritisk reflektion av metodavsnittet. En central styrka i forskningsdesignen återfinns i den teoretiska sammankopplingen mellan den interpretativa ansatsen och litteraturstudien, där generativ AI konsekvent behandlas som en relationell handlingspotential snarare än en statisk uppsättning tekniska funktioner. Detta förhållningssätt har möjliggjort en djupare förståelse för IT-praktikernas meningsskapande, vilket förstärks av att studiens metod låter den empiriska kontexten belysas genom väletablerad litteratur. Samtidigt som valda metoder bedöms vara ändamålsenliga för studiens syfte, finns det problemområden som bör belysas.

Ett av dessa områden rör urvalet av respondenter via författarnas professionella kontaktnät. Att enbart genomföra ett bekvämlighetsurval riskerar enligt Oates et al. (2022) att göra en studie partisk och svår att försvara vetenskapligt. För att minimera denna risk har bekvämlighetsaspekten kombinerats med ett kriteriebaserat urval. Vidare har en professionell distans upprätthållits genom det interna kravet på att ingen tidigare personlig relation fick finnas mellan forskare och respondent vilket vi anser gör urvalet skäligt för studiens omfattning.

Ytterligare en aspekt som kräver reflektion är den begreppsliga förvirring som uppstod under intervjuerna rörande termen ”säkerhetsaktiviteter”. Medan författarna definierade begreppet som ett samlingsnamn för aktiviteter kopplade till IT-praktikerns säkerhetsansvar, visade det sig att tolkningen inte var enhetlig bland respondenterna. I efterhand kan konstateras att en mer standardiserad förklaring av begreppet borde ha tillhandahållits samtliga informanter före intervjutillfället för att säkerställa en gemensam utgångspunkt.

4. Empiri

4.1 Transformation av arbetsprocesser

4.1.1 Säkerhetsaktiviteter

I planeringsfasen beskriver respondenterna hur säkerhetsaktiviteter tar sin början genom insamling av krav och riskidentifiering. Respondent 1 förklarar att organisationen ställer specifika säkerhetskrav som teamet måste förhålla sig till, vilket innefattar krav på monitorering och larmuppsättning (R1, 8). Dessa krav relateras till övergripande aktiviteter som “compliance” och “threat intelligence” (R5, 18) och ansvarsfördelningen innebär att teamen själva kan ansvara för applikationsspecifika lösningar som databaser och kryptering (R3, 2). För att identifiera potentiella hot används strukturerade metoder såsom TARA (Threat Avoid Reduce Accept) där teamet utvärderar applikationernas säkerhet genom att besvara ett omfattande antal frågor rörande externa hot (R1, 52). En respondent betonar i detta sammanhang vikten av "shift-left", där säkerhetsaspekter hanteras så tidigt som möjligt genom samarbete redan i designfasen (R5, 20).

När processen övergår i mjukvaruproduktion (create) och verifiering (verify) skiftar fokus mot den faktiska kodens integritet och granskningsmetoder. En respondent beskriver hur arbetet innefattar att säkerställa att osäker kod inte skrivs och att mjukvarudependencies hålls uppdaterade (R4, 14), medan respondent 2 (22) poängterar vikten av att kontrollera att interna system inte oavsiktligt exponeras utåt. För att skydda känslig data i denna fas används tekniska lösningar som AWS Secrets Manager för att säkerställa att lösenord och credentials förvaras säkert och inte exponeras (R1, 28; R1, 30). Vidare identifieras kodgranskning (code review) som en fundamental aktivitet som hanteras via allt från löpande chattbaserade prioriteringar (R1, 40) till formella checkpoints för varje pull request (R4, 16). Respondent 3 lyfter även fram vikten av att granska designdokument i ett mänskligt läsbart format för att möjliggöra gemensamma beslutstagande kring arkitektur (R3, 40).

> *“Det är på lite olika nivåer. I flödet från kod till leverans har vi tre huvudmekanismer för att säkerställa att det vi levererar håller både buggfri kvalitet och en säkerhetskvalitet, vilket på ett sätt kan gå lite hand i hand. Vi arbetar med pull requests där allt granskas. Vi har landat i att mänsklig code review görs vid behov. Vissa av våra applikationer är kritiska eftersom vi hanterar mycket pengar, och då görs ofta en mänsklig granskning. I våra pipelines har vi annars automatiserad statisk kodanalys och testning. Som ett sista steg görs allt via en manuell release; ingenting går ut utan att en människa har godkänt det.”* (R3, 20)

I de avslutande stegen av verifieringsfasen dominerar automatiserade säkerhetsmekanismer, vilka ofta beskrivs som processer som varit etablerade långt innan introduktionen av generativ AI (R4, 20). Respondent 3 beskriver hur statisk kodanalys och automatiserad testning fungerar som centrala filter i pipelinen för att säkerställa säkerhetskvalitet (R3, 20; R3, 24). Detta kompletteras med skanningsverktyg för att identifiera kända CVE-er (Common Vulnerabilities and Exposures) i kodbasen (R4, 14). Respondent 6 poängterar att en av de viktigaste aktiviteterna är prioritering och klassificering av dessa sårbarheter när datamängden

är omfattande (R6, 22). Samtidigt som automation ses som nödvändig för att hantera den operativa bördan (R5, 24), kvarstår manuella element i slutet av kedjan. Respondent 2 betonar att mänsklig interaktion fortfarande krävs vid felsökning av komplexa problem (R2, 12) och respondent 3 förtydligar att kritiska applikationer ofta kräver en manuell release som ett sista godkännande steg innan produktion (R3, 20).

4.1.2 AI inom DevSecOps

Användningen av generativ AI inom DevSecOps beskrivs av respondenterna som en nyligen integrerad del av arbetsflödet, om än med varierande mognadsgrad mellan olika organisationer. Respondent 1 förklarar att införandet i större organisationer kan vara känsligt, men att verktyg som Amazon Q redan används som stöd vid kodgranskning och för att generera lösningsförslag (R1, 10; R1, 12). Denna dagliga interaktion bekräftas av respondent 3 och respondent 4, som uppger att tekniken används i princip hela tiden för allt från planering i Jira till kodning i utvecklingsmiljöer (R3, 4; R3, 8; R4, 4). Respondent 2 beskriver en snabb utveckling där AI har gått från att vara ett sökverktyg till att bli en integrerad del av hur man snabbar på tidskrävande processer (R2, 6). Denna effektivisering lyfts även fram av respondent 6 som ser en fördubbling av arbetsprestandan i vissa fall, särskilt gällande testning och dokumentation medan respondent 4 betonar hur AI möjliggjort en snabbare startsträcka vid introduktion i nya, komplexa kodbaser (R6, 18; R4, 12).

När det gäller integrationen i säkerhetsprocesser och hantering av sårbarheter råder delade meningar om teknikens nuvarande förmåga. Respondent 3 menar att AI aktivt bidrar till att både identifiera och åtgärda fler säkerhetsbrister än tidigare, medan respondent 1 och respondent 5 inte har observerat någon markant skillnad i antalet fångade brister ännu (R3, 34; R1, 48; R5, 32). Respondent 2 ger dock ett konkret exempel där AI användes för att framgångsrikt verifiera skydd mot SQL-injections i ett specifikt ramverk (R2, 38). Vidare beskrivs en metodologisk användning där AI assisterar i att sätta upp statiska processer och skapa anpassade regler för kodanalys snarare än att helt ersätta de existerande statiska verktygen (R3, 24; R3, 32).

> *"Yes, absolut. Och också fixa fler brister. Ofta visste vi om brister tidigare, framför allt om vi pratar säkerhetshål eller potentiella brister i kod, där vi kände till dem men inte hade tid att fixa dem. AI kan hjälpa oss att både hitta fler problem och definitivt att täppa till de problem som finns."* (R3, 34)

Framtidsutsikterna för AI inom DevSecOps pekar mot en mer omfattande automatisering där autonoma agenter och avancerade beslutsstöd förväntas spela en central roll i säkerhetsarbetet. Respondent 1 pekar på användningen av agenter för repetitiva uppgifter och automatiserad support, och respondent 6 ser en potential i framtida modeller som kan genomföra penetrationstester för att ligga steget före angripare (R1, 60; R6, 16; R6, 32). Dokumentationsarbetet identifieras av flera som ett område med stor potential där AI kan skapa mer detaljerad och uppdaterad information (R2, 46; R3, 38; R5, 12). Samtidigt framhåller respondent 5 att användningen ofta är personberoende och begränsas av att tillgängliga modeller inte alltid är anpassade för specifik defensiv säkerhet eller känslig data (R5, 14; R5, 26). För att AI ska kunna leverera fullt ut i en DevSecOps-kontext betonar respondent 4 och respondent 5 behovet av robusta automatiserade test- och verifieringsprocesser som kan hantera den ökade mängd kod som tekniken genererar (R4, 34; R5, 38).

4.1.3 Hastighet kontra säkerhetskvalitet

Respondenterna beskriver en motstridig bild av hur generativ AI påverkar balansen mellan arbetstempo och säkerhetskvalitet. Å ena sidan betonas en markant ökning av effektivitet och produktivitet då en respondent skiljer på att vara ”snabbare” och att vara ”mer effektiv”, där AI bidrar till att lösa problem som tidigare krävde assistans från kollegor (R1, 26). Respondent 2 instämmer i att tekniken i slutändan är mer effektiv trots att den ibland skapar merarbete och respondent 6 beskriver utvecklingen som en digital transformation som sker i en språngartad takt (”leapfrog pace”) (R2, 42; R6, 10). Även inom administration och dokumentation upplevs effektivitetsvinster, där en respondent lyfter fram att det blivit avsevärt enklare att leverera en mer detaljerad dokumentation (R3, 10).

> *“Tidigare skrev jag väldigt mycket kod själv och var mycket mer insatt i koden och förstod mer av den. Man itererade mer kring själva koden man sedan lägger upp för review. Nu spottar en AI ur sig jättemycket kod och jag upplever inte att jag har samma process kring att granska allting som genereras.”* (R4, 10)

Samtidigt uttrycks en omfattande oro för hur den ökade hastigheten och mängden genererad kod påverkar granskningsprocessen och den slutgiltiga kvaliteten. En respondent identifierar den största utmaningen i att AI möjliggör produktion av tusentals rader kod per dag, vilket gör det omöjligt för teamet att genomföra meningsfulla kodgranskningar (“code reviews”) (R3, 14; R3, 40). Denna problematik bekräftas av en annan respondent som menar att den ökade produktionstakten medför säkerhetsimplikationer då granskningen inte blir lika noggrann som när koden tidigare skrevs manuellt (R4, 10; R4, 30). Vidare beskriver respondent 5 hur ett högt flöde av kod från enskilda utvecklare riskerar att förvandla de validerande medarbetarna till kritiska flaskhalsar vilket i en manuell kontext leder till att användningen av AI snarare upplevs som en förlust än en vinst för organisationen (R5, 36).

Riskerna med att prioritera hastighet framför kontroll konkretiseras genom exempel på bristande tillförlitlighet och ansvar. Respondent 1 påpekar risken med att lita blint på att AI:n täckt alla säkerhetsfall, särskilt då resultatet är helt beroende av den individuella promptens utformning (R1, 18). Respondent 4 belyser hinder i form av otydlig ansvarsfördelning gällande rättighetsaspekter samt merarbete orsakat av hallucinationer där AI föreslår icke-existerande verktyg eller inkompatibla lösningar (R4, 10; R4, 32). Respondent 6 belyser att den snabba tekniska utvecklingen ännu inte matchas av tillräckliga säkerhetskontroller eller ”guardrails”, vilket skapar en risk för att produkternas säkerhetsprofil kan äventyras (R6, 10). Respondent 4 summerar läget med att skillnaden i vem som genererar koden är liten, men att AI gör bristerna i de automatiserade test- och verifikationsprocesserna mycket tydligare på grund av den extrema hastigheten (R4, 34).

4.2 Affordance-teorin

4.2.1 Funktionella handlingsmöjligheter

Respondenterna beskriver hur generativ AI uppfattas erbjuda en rad funktionella handlingsmöjligheter som sträcker sig från strategiskt beslutsstöd till teknisk exekvering. Respondent 1 framhåller att tekniken främst underlättar vid vägval, såsom valet mellan olika datastrukturer och databasmodeller, snarare än att enbart förändra själva kodskrivandet (R1,

14; R1, 16). Denna förmåga att agera som ett kognitivt stöd vid komplexa bedömningar delas av respondent 4, som beskriver hur AI möjliggör en snabbare förståelse för dataflöden i mikrotjänstarkitekturer vilket underlättar onboarding i nya kodbaser (R4, 12). Respondent 3 poängterar i sammanhanget att handlingsmöjligheterna är som störst i planeringsfasen, där AI kan hjälpa till att lägga upp en plan för det arkitektoniska arbetet (R3, 28).

När det gäller specifika säkerhetsrelaterade uppgifter identifieras möjligheten att hantera stora datamängder och automatisera rutinmoment som central. Respondent 6 beskriver hur AI kan användas för att filtrera och prioritera i omfattande listor med sårbarheter vilket avsevärt reducerar tiden för en initial bedömning (R6, 22). Liknande möjligheter nämns av respondent 2 gällande kontroll av biblioteksversioner och framför allt dokumentationsarbete, vilket beskrivs som en av de mest tidskrävande men viktiga aktiviteterna (R2, 34; R2, 46). Respondent 3 lyfter fram en mer avancerad handlingsmöjlighet i form av att låta AI bygga skräddarsydda regler för statisk kodanalys (custom linters), vilket skapar förutsättningar för att upprätthålla en enhetlig kodbas även när stora mängder kod genereras automatiskt (R3, 24; R3, 32).

> *"Dock finns en problematik: jag är senior och gör en personlig code review av koden AI:n genererar. Jag har en tydlig idé om hur koden bör se ut, vilket gör granskningen relativt enkel. För juniora utvecklare eller de som är nya i ett programmeringsspråk är det svårare. Där kan AI nästan leda till att man levererar mindre av hög kvalitet, vilket medför mer tid för code reviews och ändringar fram och tillbaka."* (R3, 16)

Framtida handlingsmöjligheter diskuteras i termer av autonoma agenter och systemövergripande analyser. Respondent 1 ser en potential i att använda AI-agenter för att automatisera repetitiva supportärenden, exempelvis genom att kontrollera systemkonfigurationer (R1, 60). På ett mer strategiskt plan föreslår respondent 5 att AI kan ge en mer heltäckande bild av en organisations säkerhetsprofil (security posture) genom att analysera korrelationer mellan kvalitetsproblem och incidenter över stora datamängder (R5, 8; R5, 22). Respondent 6 pekar även på handlingsmöjligheter inom offensiv säkerhet där framtida modeller förväntas kunna genomföra simulerade attacker och möjligheten att integrera tidsseriedata för realtidsanalys av systemprestanda (R6, 16; R6, 36). Samtidigt understryker flera respondenter att dessa möjligheter kräver en kompetent användare. Respondent 2 och respondent 3 betonar att handlingsmöjligheten att generera komplexa uppsättningar, såsom databasstrukturer, kräver en viss kompetens hos människan för att kunna verifiera att resultatet är korrekt låst och håller hög kvalitet (R2, 22; R3, 16).

4.2.2 Aktualisering av handlingsmöjligheter

Aktualiseringen av handlingsmöjligheter genom generativ AI tar sig uttryck i en ökad yrkesmässig självständighet där tekniken fungerar som ett kognitivt beslutsstöd vid komplexa bedömningar. Respondent 1 förklarar att behovet av att rådfråga kollegor eller arkitekter vid specifika strategiska vägval vid val av databasmodeller har minskat i takt med att AI-verktyget har börjat användas för att utvärdera olika alternativ (R1, 26). Aktualiseringen av dessa möjligheter framhålls dock som beroende av individens eget initiativ. Respondent 2 menar att graden av integration i det dagliga arbetet styrs av hur mycket utvecklaren själv väljer att nyttja verktyget (R2, 6). För respondent 3 är en förutsättning för att framgångsrikt aktualisera tekniken att användaren investerar tid i den initiala planeringsfasen genom att

tillhandahålla en gedigen kontext och plan kan resultatet från AI:n i slutändan betraktas som mer tillförlitligt (R3, 28).

I den praktiska tillämpningen framträder aktualiseringen som en utpräglad iterativ process där användaren ständigt verifierar AI:s utdata. Respondent 2 beskriver hur AI används för att generera en första grund som sedan bearbetas genom flera vändor av diskussion och optimering, en metodik som i vissa avseenden har ersatt tidigare processer för självgranskning och manuella tester (R2, 14; R2, 44). Denna form av dialogbaserad användning aktualiseras även vid identifiering av specifika säkerhetsrisker. Respondent 1 och respondent 2 redogör för hur de använder tekniken för att bolla problemställningar kring exempelvis SQL-injections för att få snabba verifieringar mot gällande dokumentation och ramverk (R1, 32; R2, 38). Respondent 4 tillägger att tekniken har aktualiserats som ett värdefullt stöd för att få en initial sammanfattning och granskning av pull requests vilket underlättar teamets gemensamma granskningsarbete (R4, 18).

> *"Där vi står idag är det en fråga om förtroende, inte kostnad. Om säkerhetspersonen inte upplever att man kan lita på verktyget kommer den inte använda det. [...] Förtroende är huvudfaktorn här gällande exempelvis threat modeling. Kan du lita på verktyget och det inte kostar enorma mängder, så hade man självklart använt det, för det är troligtvis mycket mer kreativt än vad du är."* (R5, 30)

Trots de identifierade möjligheterna finns det hinder som gör att man medvetet avstår från att aktualisera vissa handlingsmöjligheter, främst på grund av frågor rörande tillit och teknikens tekniska begränsningar. Flera respondenter poängterar att de drar en gräns vid att låta AI hantera känslig data eller skapa färdiga lösningar för kritiska säkerhetsfunktioner utan omfattande mänsklig kontroll (R1, 32; R2, 22; R5, 14; R5, 22). Respondent 6 betonar att fullt automatiserade AI-lösningar inte aktualiseras i deras verksamhet då tekniken är probabilistisk och icke-deterministisk, vilket anses oförenligt med branschens krav på konsistens (R6, 26; R6, 34). Respondent 3 liknar situationen vid piloter i en cockpit, även om den tekniska potentialen för full automation existerar krävs en mänsklig närvaro för att upprätthålla förtroendet gentemot både interna användare och externa myndigheter (R3, 26). Samtidigt argumenterar respondent 5 för att en framtida fullständig aktualisering kräver ett skifte i förtroende, där man accepterar att en korrekt instruerad automation är mer konsekvent och gör färre misstag än en människa vid repetitiva uppgifter (R5, 30; R5, 38).

4.3 Human-in-the-loop

4.3.1 Komplementär teamprestation och asymmetri

Samspelet mellan människa och generativ AI beskrivs av respondenterna som en asymmetrisk relation där parternas olika styrkor utnyttjas för att höja teamets totala prestation. Respondent 3 belyser hur yrkesrollen genomgår ett skifte från att utföra manuell kodning till att fungera som en planerare och granskare (R3, 36). Respondent 2 nämner ett iterativt arbetssätt där AI används för att generera en grund som sedan diskuteras och kontrolleras flera gånger (R2, 44). Denna ökade självständighet gör att utvecklare i mindre utsträckning behöver vänta på upptagna kollegor eller arkitekter för strategiska vägval (R1, 26). Samtidigt understryker

respondent 6 vikten av att alltid ha en människa med i loopen för att styra och kontrollera de automatiserade delarna av processen (R6, 24).

> *"Mitt personliga jobb har förändrats väldigt mycket från att få en idé om hur jag löser någonting och sen gå in och knacka kod, skriva tester och de bitarna, till att det nu egentligen är mer att jag får en idé i huvudet, skriver ner den i en plan med hjälp av AI:n så att AI:n är med på vad det är jag faktiskt vill få ut, och att det blir tydligt på pränt."* (R3, 36)

Asymmetrin blir särskilt tydlig i fördelningen mellan teknisk exekvering och komplexa designbeslut. Respondent 1 förklarar att AI är effektiv på att hantera smådetaljer och repetitiva uppgifter, såsom syntax eller specifika kodattribut, vilket frigör utrymme för människan att fokusera på designbeslut och domänkontext (R1, 36; R1, 42). Denna arbetsfördelning ses även inom säkerhetsarbetet där AI används för att prioritera bland hundratals sårbarheter vilket reducerar tiden för den mänskliga bedömningen (R6, 22). Respondent 5 betraktar tekniken som ett stöd som kan avlasta människan vid monotona uppgifter där den mänskliga faktorn ofta brister (R5, 38) medan respondent 4 ser ett värde i att låta tekniken göra en initial granskning av kod för att hitta saker som människan lätt missar (R4, 18). Det finns dock en medvetenhet om riskerna med detta samspel. Respondent 6 lyfter fram behovet av utbildning och ökad mänsklig medvetenhet för att förstå hur AI kan trigga säkerhetsproblem om den används på fel sätt (R6, 16; R6, 34).

4.3.2 Helhetsperspektiv och "ground truth"

Intervjumaterialet belyser en genomgående uppfattning om att generativ AI är skicklig på tekniska detaljer men saknar förmågan att förstå projektövergripande helheter och domänspecifika sammanhang. Respondent 1 förklarar att även om tekniken kan identifiera specifika kodattribut, saknar den kännedom om domänen och kan därför inte förutse om ett team faktiskt får rätt information i en produktionsmiljö (R1, 34; R1, 36; R1, 54). Denna brist på domänkontext gör att mänsklig granskning förblir nödvändig för att bedöma om logiken är rimlig i det stora hela. Respondent 2 fyller i att AI-modellerna ofta genererar svar baserat på vad som är mest vanligt förekommande i träningsdatan vilket innebär att tekniken ofta brister på detaljnivå eller inom specifika områden som kräver djupare kontextuell förståelse (R2, 18; R2, 26). Liknande erfarenheter delas av respondent 4 som menar att tekniken tenderar att missa små men avgörande detaljer i den stora kontexten vilket kan leda till dolda säkerhetsproblem (R4, 30).

Behovet av människan som den slutgiltiga referenspunkten, eller en "ground truth", framstår som centralt för att säkerställa att det tekniken genererar är korrekt och tillförlitligt. Respondent 3 bekräftar att människan alltid fungerar som ett filter i slutet av processen och agerar som den nödvändiga referenspunkten genom att granska och finjustera AI:ns exekvering för att säkerställa att resultatet överensstämmer med helhetsplanen (R3, 22; R3, 36). Respondent 6 betonar att man aldrig bör acceptera utdata utan validering mot betrodda källor då tekniken i grunden är en probabilistisk modell snarare än en garant för sanning (R6, 28; R6, 34). Denna nödvändiga verifieringsprocess beskrivs av respondent 2 som ett iterativt arbete där man ständigt får kontrollera att resultatet faktiskt blev som man tänkt sig, vilket i vissa fall har ersatt tidigare manuella testprocesser (R2, 14; R2, 44).

> *"When it comes to generative AI for chats, for example, I pretty much never accept the output until I validate it. When I receive an output, I would like to*

> *know which are the trusted sources and where it comes from to avoid hallucinations. That's the human-in-the-loop part that I was discussing before. Unfortunately, the percentage of hallucinations in AI models is still quite big. Depending on what kind of queries you're asking, if some of them are very critical, you need to ensure that you have the right sources. Sources in general are one of the most important things within the AI model."* (R6, 28)

Möjligheten att få relevanta svar från tekniken beskrivs vara direkt proportionell mot mängden kontext och data användaren tillhandahåller. Respondent 3 poängterar att en detaljerad plan och omfattande kontext är grundpelaren för att AI:ns respons ska bli tillförlitlig men noterar samtidigt att modeller har tekniska begränsningar i hur mycket information de kan hantera samtidigt (R3, 28; R3, 30; R3, 32). För respondent 4 är kvaliteten på AI:ns förståelse även beroende av vilket programmeringsspråk som används då språk med mindre träningsdata resulterar i en sämre förmåga att hantera komplicerade sammanhang jämfört med mer utbredda språk (R4, 28). Respondent 5 lyfter ytterligare en dimension genom att betona behovet av specifika och slutna AI-miljöer för att kunna mata in känslig sårbarhetsdata utan risk för informationsläckage då en generell AI saknar den specifika kontext som krävs för att göra meningsfulla analyser av en organisations faktiska hotlandskap (R5, 16). Slutligen konstaterar respondent 1 att även om AI är skicklig på tekniska detaljer förblir systemdesign subjektivt och präglat av tycke och smak vilket gör mänsklig expertbedömning oumbärlig för helhetsperspektivet (R1, 44).

5. Diskussion

I detta kapitel diskuteras studiens empiriska fynd i relation till undersökningsramverket, konceptuella ramverket och tidigare forskning. Diskussionen är strukturerad utifrån de tre teoretiska perspektiven som ligger till grund för studien. Först behandlas Affordance-teorin där funktionella handlingsmöjligheter och deras aktualisering analyseras för att förstå hur IT-praktiker uppfattar och realiserar generativ AI:s potential. Därefter diskuteras human-in-the-loop med fokus på komplementär teamprestation samt praktikerns roll som referenspunkt i säkerhetsarbetet. Slutligen prövas dessa insikter mot klassificeringsmodellen för att belysa hur upplevelsen av generativ AI skiftar beroende på var i DevSecOps-cykeln tekniken implementeras. Kapitlet avslutas med en redogörelse för studiens begränsningar.

5.1 Affordance-teorin

5.1.1 Funktionella handlingsmöjligheter

Det mest framträdande i respondenternas svar är inte bredden i de uppfattade handlingsmöjligheterna utan den tydliga skillnaden i upplevd potential som korrelerar med erfarenhetsnivå. Seniora IT-praktiker upplever en handlingspotential som sträcker sig från strategiskt beslutsstöd vid arkitekturval till avancerade säkerhetsapplikationer som skräddarsydda regler för statisk kodanalys (R1, 14; R1, 16; R3, 24; R3, 32). Respondent 3 beskriver att handlingsmöjligheterna upplevs som störst i planeringsfasen (R3, 28), medan respondent 6 upplever störst värde i förmågan att filtrera och prioritera bland hundratals sårbarheter (R6, 22). Respondent 2 upplever däremot markant mer begränsade möjligheter och drar en tydlig gräns vid komplexa databasstrukturer utan mänsklig kontroll (R2, 22). Det är alltså inte tekniken som varierar mellan respondenterna, utan upplevelsen av vad den erbjuder.

Markus och Silver (2008) definierar funktionella handlingsmöjligheter som en strikt relation mellan ett tekniskt objekt och en specifik användargrupp snarare än som en inneboende egenskap hos teknologin. Studiens empiri ger ett tydligt stöd för detta då upplevelsen av vad generativ AI erbjuder formas av vem som möter den. Valbøs (2021) rekommendation att uteslutande betrakta affordances som en relationell handlingspotential, snarare än Normans (1988) citerat i Valbø (2021) tolkning om inbyggda tekniska funktioner, visar sig analytiskt produktiv i just denna kontext. Om handlingsmöjligheterna betraktades som tekniska egenskaper hos systemet skulle den systematiska variationen i upplevelse sakna förklaring. Det är precisionen i det relationella perspektivet som gör det möjligt att förstå varför samma verktyg upplevs som kraftfullt av en senior IT-praktiker och begränsat av en junior.

Gibsons (1979) citerat i Valbø (2021) grundläggande princip tillför en ytterligare analytisk dimension. En affordance existerar oberoende av om den uppfattas eller inte, men för att den ska få praktisk betydelse måste den uppfattas och aktualiseras av aktören. Respondenterna beskriver konsekvent framtida handlingsmöjligheter som exempelvis autonoma agenter, offensiva säkerhetstester och analyser av organisationens övergripande säkerhetsprofil (R1, 60; R5, 8; R5, 22; R6, 16; R6, 36). Gemensamt är att dessa möjligheter uppfattas men ännu inte har aktualiserats i praktiken. Teknikens handlingspotential existerar, men potentialens upplevda omfång och riktning dikteras av relationen mellan praktikern och systemet. Det är

denna upplevda potential och inte tekniken i sig som formar hur respondenterna orienterar sig mot framtida möjligheter. Markus och Silver (2008) betonar att funktionella handlingsmöjligheter uteslutande rör potentiell användning och att denna potential utgör ett nödvändigt villkor för, men ingen garanti för, det faktiska användandet. En distinktion som empirin illustrerar med tydlighet.

Studien argumenterar för att detta fynd har en betydelse som sträcker sig bortom själva kompetensasymmetrin. Generativ AI är inget neutralt produktivitetsverktyg som lyfter alla på samma sätt. Det fungerar snarare som en hävstång för befintlig kompetens och förstärker de färdigheter som redan finns på plats. Respondent 3 formulerar detta explicit genom att beskriva hur granskningen av AI-genererad kod upplevs som hanterbar för en senior praktiker med en tydlig idé om hur koden bör se ut, medan samma situation för en junior praktiker riskerar att leda till leverans av lägre kvalitet och merarbete (R3, 16).

Perry et al. (2023) visar att utvecklare med AI-stöd ofta producerar mindre säker kod samtidigt som de uppfattar den som säkrare, vilket skapar en falsk trygghet. Studiens empiri bekräftar denna paradox i en DevSecOps-kontext då upplevelsen av handlingsmöjligheter för den juniora praktikern riskerar att bli missvisande i förhållande till det faktiska resultatet. Handlingsmöjligheten uppfattas och används i praktiken, men eftersom förmågan att granska AI-svaren saknas blir utfallet i slutändan skadligt för säkerheten. Genom Markus och Silvers (2008) teoretiska lins kan detta fenomen förklaras med att problemet inte handlar om tillgången till tekniken utan om avsaknaden av den kompetensbas som krävs för att en meningsfull handlingsmöjlighet ska kunna uppstå i relationen.

Markus och Silver (2008) introducerar konceptet symboliska uttryck för att fånga hur användare tolkar teknikens kommunikativa möjligheter och underliggande intentioner. I studien framträder en tydlig bild av att generativ AI inte upplevs kommunicera en intention av deterministisk sanning. Snarare signalerar systemet genomgående sin egen fallbarhet. Respondent 6 beskriver en påtaglig frekvens av hallucinationer och betonar att tekniken i grunden är probabilistisk, vilket gör att utdata aldrig kan accepteras utan validering (R6, 26; R6, 28; R6, 34). Vidare illustrerar systemets oförmåga att förstå domänspecifika sammanhang (R2, 18; R2, 26) ett symboliskt uttryck av kontextuellt beroende. Eftersom tekniken kontinuerligt kommunicerar behovet av guidning tolkas dess syfte inte som en autonom agent, utan som ett iterativt diskussionsverktyg (R2, 14; R2, 44). Det är denna symboliska dimension av systemets visade behov av en mänsklig guide som formar respondenternas tolkning av vilken roll tekniken kan tillåtas spela i kritiska DevSecOps-moment. Artefaktens symboliska uttryck av fallbarhet förmedlar tydligt att den inte är utformad för att bära ett eget säkerhetsansvar. Följaktligen kommunicerar systemet självt att dess fulla potential endast kan realiseras när det omgärdas av mänsklig kompetens, vilket avhåller IT-praktikerna från att delegera strategiska säkerhetsbeslut till AI:n.

5.1.2 Aktualisering av handlingsmöjligheter

Diskussionen landar här i att den avgörande frågan inte är om generativ AI erbjuder handlingsmöjligheter utan under vilka kognitiva och kontextuella villkor dessa faktiskt realiseras. Enligt Strong et al. (2014) utgör en uppfattad handlingsmöjlighet endast en potentiell funktion vars struktur träder fram först under aktualiseringen. För att tekniken ska realiseras i praktiken krävs att aktören uppfattar ett förväntat utfall som ett omedelbart och konkret resultat vilket är användbart för att nå övergripande mål (Strong et al. 2014). Det är denna upplevda konsekvens och nytta snarare än teknikens kapacitet i sig som styr om och

hur handlingsmöjligheten aktualiseras. Empirin illustrerar detta tydligt då aktualiseringen beskrivs som en sökande process fylld av verifiering och ifrågasättande snarare än att uppgiften helt lämnas över till systemet. Respondent 2 beskriver hur AI används för att generera en grund som sedan bearbetas genom flera vändor av diskussion och optimering, en process som i vissa avseenden har ersatt tidigare manuella testprocesser (R2, 14; R2, 44). Respondent 1 och respondent 2 beskriver hur de aktivt bollar säkerhetsproblem mot tekniken för snabba verifieringar mot gällande dokumentation och ramverk (R1, 32; R2, 38). Aktualiseringen är genomgående en individuell förhandling om tillit där den förväntade konsekvensen i form av ett tillförlitligt och kontextuellt förankrat svar ständigt vägs mot den upplevda osäkerheten i AI:ns utdata.

Studien hävdar att det främsta hindret för en fullständig aktualisering är AI-systemets bristande förklarbarhet. Respondent 5 beskriver hur tilliten till verktyget, snarare än kostnaden, är den avgörande faktorn för om tekniken används i säkerhetskritiska sammanhang som hotmodellering (R5, 30). Bauer et al. (2023) visar att viljan att följa och aktualisera ett AI-systems råd är starkt beroende av om systemets utdata stämmer överens med praktikerns existerande mentala modeller och att en motstridighet skapar kognitiv dissonans som leder till att handlingsmöjligheten förblir icke aktualiserad. Respondent 6 beskriver hur fullt automatiserade AI-lösningar aktivt väljs bort eftersom tekniken upplevs som probabilistisk och icke-deterministisk, vilket är oförenligt med branschens krav på konsistens (R6, 26; R6, 34). Respondent 3 använder cockpit-metaforen för att illustrera att full automation visserligen är tekniskt möjlig men att en människa vid spakarna förblir nödvändig för att garantera förklarbarhet och upprätthålla förtroendet (R3, 26). Studien hävdar att detta inte handlar om ett ogrundat motstånd mot ny teknik, utan är en logisk reaktion på AI-systemets brist på förklarbarhet. Fu et al. (2025) poängterar att utan begripliga förklaringar får experter svårt att lita på besluten eller rätta till fel. Detta gör det i sin tur svårt att använda sin egen erfarenhet för att styra verktyget på ett bra sätt. Avsaknaden av tydliga förklaringar i verktygen är alltså inget sidospår, utan själva kärnan i förklaringen till varför det är så svårt att omsätta tekniken i praktiskt arbete.

Bauer et al. (2023) belyser vidare att erfarna experter har en stark tendens till bekräftelsebias, där de tar till sig AI:s råd när dessa bekräftar deras egna teorier men aktivt ignorerar förklaringar som utmanar etablerad kunskap. Empirin visar tydligt hur användarnas erfarenhet påverkar deras sätt att arbeta med tekniken. De seniora respondenterna beskriver ett selektivt bruk där de utnyttjar AI:ns styrkor för rutinmoment men behåller en aktiv skepsis vid säkerhetskritiska bedömningar. Detta står i kontrast till oron för att juniora praktiker ska acceptera AI:ns svar okritiskt då de ännu saknar den djupa yrkeskunskap som krävs för att kunna granska resultatet (R3, 16). Det är alltså inte ett enhetligt förhållande till AI-systemets utdata som empirin avslöjar, utan ett spektrum av tillit baserat på erfarenhet där bekräftelsebias och okritisk acceptans fungerar som varandras motpoler.

Von Zahn et al. (2025) tillför en nyansering av denna bild då författarna förklarar att mötet med ett motstridigt AI-svar inte enbart behöver leda till kognitiv dissonans. När AI synliggör en diskrepans mellan systemets logik och användarens resonemang kan detta trigga en förbättrad metakognitiv kalibrering där praktikern omvärderar sin övertro på den egna kompetensen och därigenom faktiskt ökar sin benägenhet att delegera till systemet. Kalibreringen sker dock utan stöd av explicita XAI-funktioner i respondenternas verktyg, vilket innebär att processen är beroende av individuell erfarenhet och yrkesmässigt omdöme snarare än av systemets design. Konsekvensen är att kalibreringen sker ojämnt och personberoende i enlighet med en respondents observation om att nyttjandegraden varierar kraftigt mellan individer inom samma organisation (R5, 26). Det metakognitiva perspektivet

förklarar därmed varför vissa praktiker med erfarenheten som grund lyckas kalibrera sin tillit och aktualisera tekniken effektivt medan andra förblir antingen för skeptiska eller för okritiska i sin användning.

Dessa iakttagelser pekar sammantaget mot en slutsats som flyttar fokus från individens kognitiva förmåga till de bakomliggande strukturella förutsättningarna. Bedoya et al. (2024) konstaterar att AI-verktygens effektivitet i DevSecOps är direkt beroende av en underliggande mänsklig kompetens inom cybersäkerhet och att organisationens mognadsgrad är en förutsättning för en framgångsrik integration. Studiens empiri bekräftar detta men preciserar det ytterligare då aktualiseringen av handlingsmöjligheter inte enbart är beroende av individens kompetens utan av tillgången till förklarbarhet som ett organisatoriskt designval. Den ökade självständigheten i strategiska vägval hos respondent 1 (R1, 26) och kravet på gedigen kontext för tillförlitliga svar från respondent 3 (R3, 28) illustrerar att aktualiseringen kräver att praktikern aktivt kompenserar för det som systemet inte erbjuder. Även Fu et al. (2025) understryker att DevSecOps förblir djupt beroende av mänskliga experter och att XAI är ett kritiskt krav för att detta samspel ska fungera.

Sammantaget pekar dessa iakttagelser på att frånvaron av XAI i dagens AI-verktyg utgör ett påtagligt tekniskt underskott inom DevSecOps, vilket tvingar fram en förskjutning i hur förklarbarhet uppnås. Studien argumenterar för att IT-praktiker kompenserar för denna svarta låda genom att själva konstruera förklarbarhet och tillit via erfarenhetsbaserad kontextualisering och iterativ verifiering (R2, 14; R3, 28). I brist på transparent systemdesign är det därmed praktikerns kontinuerliga interaktion och metakognitiva kalibrering (von Zahn et al. 2025) som fyller gapet. Utifrån ett affordance-perspektiv (Strong et al. 2014) innebär detta att handlingsmöjligheten att fatta säkra och tillförlitliga beslut inte är en inbyggd egenskap i verktyget, utan aktualiseras uteslutande genom praktikerns aktiva, iterativa användande. Det är således i den mänskliga praktiken, snarare än i den tekniska designen, som tilliten slutgiltigt förankras.

5.2 Human-in-the-loop

5.2.1 Komplementär teamprestation och asymmetri

Respondenterna enas om att samspelet med generativ AI inte innebär att mänsklig expertis ersätts utan att den ompositioneras. Exempelvis beskriver en respondent att arbetet gått från att själv producera kod på egen hand till att snarare planera och vägleda AI:n så att den förstår idén och vad man faktiskt vill få ut (R3, 36).

Det centrala är inte enbart att arbetsuppgifterna förändras, utan hur denna förändring upplevs av IT-praktikern. Hemmer et al. (2025) betonar att effektiviteten i human-in-the-loop-metoder bygger på en informations- och förmågeasymmetri mellan människa och system och att asymmetrierna kan resultera i mer träffsäkra gemensamma beslut än vad vardera parten uppnår på egen hand. Studiens empiri bekräftar att detta asymmetriska samspel inte bara är effektivt i teknisk mening utan att det upplevs som meningsfullt och stimulerande av IT-praktikerna. Respondent 1 förklarar att AI:ns förmåga att hantera smådetaljer och repetitiva uppgifter frigör utrymme för människan att fokusera på designbeslut och domänkontext (R1, 36; R1, 42). Respondent 5 beskriver tekniken som ett stöd som avlastar vid monotona uppgifter där den mänskliga faktorn ofta brister (R5, 38). Denna upplevelse av avlastning från

rutinarbete leder till att IT-praktikern kan ägna mer kognitiv energi åt de uppgifter som upplevs som meningsfulla. Alenezi och Akour (2025) berör denna dimension utan att explicit benämna den. Författarna belyser att när IT-praktikern övergår från att manuellt producera kod till att strategiskt granska och styra AI-genererade förslag upplever denne en förskjutning mot mer kvalificerade och intellektuellt engagerande arbetsmoment, vilket skapar förutsättningar för ökad arbetstillfredsställelse. Studiens empiri ger konkret stöd för detta då respondent 4 beskriver värdet i att låta tekniken göra en initial granskning av kod för att fånga saker som människan lätt missar (R4, 18). Respondent 6 understryker vikten av att alltid ha en människa i loopen för att styra och kontrollera de automatiserade delarna av processen (R6, 24). Värdet ligger inte i att tekniken tar över utan i att den möjliggör att människan kan lägga mer av sin energi på de stora besluten och det strategiska arbetet.

Hemmer et al. (2025) finner i sina experimentella tester att samarbetet leder till färre beslutsfel jämfört med isolerat arbete och att praktikerna behåller sitt förtroende för tekniken trots att de observerar systemets brister. Studiens empiri ger ett tydligt stöd för detta då respondenterna konsekvent väljer att integrera AI i uppgifter där tekniken tillför värde, exempelvis vid initial sårbarhetsfiltrering och hantering av repetitiva smådetaljer, men utövar samtidigt en aktiv skepsis i situationer med hög säkerhetskritikalitet. Denna selektiva tillit innebär inte ett förlorat förtroende för tekniken utan baseras på en kalibrerad professionell bedömning, vilket är en nyans som Hemmer et al. (2025) inte explicit behandlar. Fu et al. (2025) beskriver integreringen av AI-verktyg i DevSecOps uttryckligen som en människoassisterad process och Bedoya et al. (2024) poängterar att avancerade säkerhetsprocesser förblir djupt beroende av mänskliga experter. Studiens empiri bekräftar och förstärker dessa slutsatser genom att visa att den komplementära teamprestationen inte är en teknisk egenskap hos systemet utan en social konstruktion som kräver att IT-praktikern aktivt och medvetet ompositionerar sig.

Studien argumenterar för att detta fynd har en kritisk implikation för hur organisationer bör förstå AI-integration. Hemmer et al. (2025) avråder från att se AI som ett verktyg för total automatisering och menar att man istället måste bevara den unika mänskliga kunskapen. Detta rimmar väl med hur respondent 3 beskriver sin nya roll som planerare och granskare snarare än kodare (R3, 36). Samtidigt understryker respondent 6 att det krävs en människa som styr de automatiserade delarna för att undvika att tekniken triggar nya säkerhetsproblem (R6, 24; R6, 34). Studien menar att glappet mellan vad tekniken kan utföra rent tekniskt och det ansvar som människan måste ta för säkerheten är en central utmaning som varken Hemmer et al. (2025) eller Fu et al. (2025) adresserar fullt ut. Genom att repetitiva och monotona säkerhetsuppgifter istället delegeras bort (R5, 38) blir det tydligt att AI:s främsta påverkan ligger i hur den omformar upplevelsen av yrkesrollen. IT-praktikerna upplever en direkt ökning av arbetstillfredsställelsen (Alenezi & Akour, 2025) eftersom human-in-the-loop-konfigurationen tvingar fram en intellektuell ompositionering då IT-praktikern går från att vara en reaktiv utförare till en proaktiv och strategisk vägledare (R1, 36; R3, 36). Detta visar sammantaget att generativ AI i DevSecOps inte bara förändrar säkerhetsarbetets hastighet, utan skapar en positiv arbetsupplevelse genom att frigöra kognitivt utrymme för domänexpertis när det implementeras för att komplettera IT-praktikerns och GAI:s asymmetri.

5.2.2 Helhetsperspektivet och ”ground truth”

Respondenterna delar en gemensam upplevelse av att generativ AI kan identifiera att en variabel bör vara "final" eller att en klass borde vara "private", men saknar insikt i om den

genererade koden faktiskt uppfyller verksamhetens mål i produktion (R1, 34; R1, 36). Respondent 2 beskriver hur AI-modellerna genererar svar baserat på vad som är mest vanligt i träningsdatan, vilket innebär en upplevd otillförlitlighet i situationer som kräver djupare kontextuell förståelse (R2, 18; R2, 26). Respondent 4 uttrycker en frustration över att tekniken tenderar att missa avgörande detaljer i den stora kontexten (R4, 30) och respondent 6 beskriver en grundläggande osäkerhet inför teknikens output eftersom den i grunden är en probabilistisk modell snarare än en garant för sanning (R6, 28; R6, 34). Det är denna upplevda osäkerhet som driver behovet av en aktiv mänsklig referenspunkt och som formar hur IT-praktikerna förhåller sig till tekniken i det dagliga arbetet.

Grønsund och Aanestad (2020) undersöker hur human-in-the-loop-konfigurationer uppstår när AI-system introduceras och observerar att det mänskliga arbetet inte elimineras utan omformas till kompletterande arbete bestående av en kontinuerlig feedbackloop. Studiens empiri bekräftar denna omformning men tillför en upplevelsedimension som författarna inte tar upp. Respondent 3 beskriver hur den mänskliga granskningen upplevs som nödvändig för att säkerställa att AI:ns exekvering överensstämmer med helhetsplanen (R3, 22; R3, 36). Respondent 6 validerar alltid AI:ns utdata mot betrodda källor (R6, 28). Respondent 2 beskriver en iterativ verifieringsprocess som i vissa avseenden har ersatt tidigare manuella testprocesser men som upplevs som mer kognitivt krävande eftersom hen nu måste bedöma riktigheten i kod som hen inte själv har skrivit (R2, 14; R2, 44). Det förstärkande arbetet är alltså inte neutralt i upplevelsetermer då det innebär en förskjutning av den kognitiva belastningen från produktion till kritisk granskning, vilket ställer nya krav på praktikerns kompetens och uppmärksamhet.

Alenezi och Akour (2025) och Chen et al. (2021) konstaterar att mjukvaruutvecklarens yrkesroll fundamentalt omformas från primär skapare av kod till operatör som övervakar och kritiskt granskar AI-genererade förslag. Studiens empiri bekräftar detta men adderar en säkerhetsspecifik dimension som litteraturen inte explicit behandlar då IT-praktikern i DevSecOps-kontexten inte enbart validerar kodens funktionella korrekthet utan dess säkerhet i en specifik domän med specifika hotaktörer (R4, 30; R5, 16). Perry et al. (2023) och Corso et al. (2024) betonar att kodassistenter ofta saknar insikt i den specifika säkerhetskontexten vilket kräver rigorös mänsklig validering för att förhindra att funktionellt korrekt kod introducerar dolda sårbarheter. Respondent 1 exemplifierar hur denna brist upplevs konkret då AI inte kan förutse om teamet faktiskt får rätt information i produktionsmiljön eftersom den saknar domänkontexten (R1, 34; R1, 36; R1, 54). Respondent 4 tillägger att upplevelsen av AI:ns kontextuella förmåga dessutom varierar beroende på vilket programmeringsspråk som används eftersom språk med mindre träningsdata resulterar i sämre förmåga att hantera komplicerade sammanhang (R4, 28). Dessa variationer i upplevd tillförlitlighet formar i sin tur hur IT-praktikerna kalibrerar sin tillit och fördelar sitt granskningsarbete.

Studien hävdar att detta fynd visar var Grønsund och Aanestads (2020) ursprungliga resonemang inte längre räcker till. Deras studie fokuserar på algoritmer inom AI-system i kontexten av dataanalys, och den "ground truth" de beskriver handlar om att tillhandahålla korrekt referensdata för att träna systemet vidare. I DevSecOps-kontexten är "ground truth" mer komplex då det inte endast handlar om att verifiera AI:ns utdata mot kända fakta utan om att tillhandahålla en mänsklig domänspecifik säkerhetsbedömning som bygger på en förståelse för hotbilden som inte går att översätta till ren träningsdata. Respondent 5 exemplifierar detta med behovet av slutna AI-miljöer för att kunna mata in känslig sårbarhetsdata utan risk för informationsläckage eftersom en generell AI saknar den specifika kontext som krävs för att göra meningsfulla analyser av en organisations faktiska hotlandskap (R5, 16). Studien argumenterar för att detta representerar en utvidgning av Grønsund och

Aanestads (2020) resonemang till en ny och mer komplex kontext och att det förstärkande arbetet i DevSecOps inte enbart handlar om att granska och justera utan om en kontinuerlig kontextualisering som kräver djup domänexpertis inom säkerhetsområdet.

Bedoya et al. (2024) konstaterar att effektiviteten av AI-verktyg i DevSecOps är starkt beroende av organisationens mognadsgrad eftersom tekniken kräver en underliggande mänsklig medvetenhet om cybersäkerhet för att kunna integreras framgångsrikt. Studiens empiri stödjer detta men preciserar det ytterligare då mognadsgraden inte upplevs som enbart organisatorisk utan som individuell och kontextspecifik. Denna personliga dimension blir tydlig både i empirin rörande teknikens handlingsmöjligheter och i frågan om hastighet kontra säkerhetskvalitet. Den seniora expertis som respondent 3 besitter gör att hens förmåga att agera referenspunkt är välutvecklad, vilket gör granskningen relativt hanterbar och upplevs som en naturlig del av arbetsflödet (R3, 16). Samtidigt visar respondent 4 hur det uppskruvade tempot och mängden kod pekar mot en upplevelse där referensfunktionen riskerar att bryta samman när volymen helt enkelt överstiger vad en människa har kapacitet att validera (R4, 10). Följaktligen visar detta att IT-praktikern upplever rollen som systemets ”ground truth” (Grønsund & Aanestad, 2020) som en krävande omställning. Det förstärkande arbetet upplevs inte som en passiv övervakningsuppgift, utan som ett ansvar (R2, 44) där praktikern måste kompensera för AI:ns brist på domänförståelse (R1, 34; R5, 16).

5.3 Transformation av arbetsprocesser

5.3.1 Hastighet kontra säkerhetskvalitet

Ett centralt empiriskt fynd som belyser nödvändigheten av ett socioteknisk perspektiv är hur generativ AI accelererar friktionen mellan hastighet och säkerhet. Perry et al. (2023) identifierar en kritisk paradox där utvecklare med tillgång till AI-kodassistenter producerar mindre säker kod men har en falsk trygghet kring dess kvalitet. Studiens empiri bekräftar denna falska trygghet men visar att praktikerna upplever paradoxen som ett strukturellt och organisatoriskt problem snarare än enbart ett kognitivt. Respondent 4 vittnar exempelvis om en förlorad insyn och kontroll. När tekniken genererar massiva kodvolymer raderas den traditionella iterativa förståelsen för kodbasen ut, vilket omöjliggör en meningsfull granskningsprocess (R4, 10).

För att förstå vidden av detta problem måste det placeras i relation till DevSecOps-kulturens framväxt. Enligt Zhou et al. (2023) etablerades DevSecOps för att motverka den traditionella DevOps-kulturens tendens att systematiskt nedprioritera säkerhetsaspekter till förmån för agilitet. Studien hävdar att integrationen av generativ AI riskerar att framkalla en återgång till detta tillstånd. När enskilda utvecklare plötsligt kan producera tusentals kodrader per dag krossas den mänskliga granskningskapaciteten (R3, 14). Det uppstår ett läge där de validerande medarbetarna ofrivilligt förvandlas till kritiska flaskhalsar, vilket gör att AI-integrationen i en manuell granskningskontext upplevs som en operativ förlust (R5, 36).

Denna obalans förvärras av teknikens inneboende begränsningar. Bedoya et al. (2024) påtalar att språkmodeller misstolkar kontext och Chen et al. (2021) visar att verktygen sprider existerande programmeringsfel som kan framstå som legitima. Empirin bekräftar detta i form av promptberoende utdata och hallucinationer som skapar merarbete (R1, 18; R4, 32). När den mänskliga granskningen fallerar på grund av volym tvingas pipelinen förlita sig blint på

automatiserade tester. Men som respondent 4 påpekar är det just här AI:ns höga hastighet exponerar bristerna även i de automatiserade verifikationsprocesserna (R4, 34) då dessa inte är kalibrerade för att fånga AI-genererade kontextfel. Sammantaget visar det här att IT-praktiker ser att generativ AI kan exponera de mänskliga bristerna i granskningsprocessen och även förstärka dem systematiskt genom att öka volymen av kod som kräver granskning utan att proportionerligt öka granskningskapaciteten.

5.3.2 Upplevelse av GAI i säkerhetsaktiviteter

Hittills har Affordance-teorin och human-in-the-loop bidragit med varsin vinkel till IT-praktikerns upplevelse med generativ AI. Det första perspektivet förklarar den individuella uppfattningen av tekniken och det andra förklarar hur implementationen uppfattas. Nu måste dessa insikter prövas mot den faktiska kontexten för säkerhetsarbetet. Klassificeringsmodellen används här som det sista analytiska verktyget. När empirin diskuteras utifrån historisk automatiseringsgrad och utvecklingsfas belyses det att upplevelsen av GAI skiftar utifrån var i DevSecOps-cykeln tekniken implementeras.

Respondenterna bekräftar denna uppdelning i praktiken. Respondent 3 beskriver hur pipelinen består av tre tydliga mekanismer där automatiserad statisk kodanalys kompletteras med mänsklig kodgranskning vid kritiska behov, samt ett slutgiltigt mänskligt godkännande (R3, 20). Vidare tillägger respondent 4 att majoriteten av de automatiserade aktiviteterna, inklusive CVE-skanning och statisk analys, var etablerade processer långt innan generativ AI introducerades (R4, 20). Detta är analytiskt viktigt då det innebär att generativ AI inte introduceras för att bygga en ny säkerhetspipeline utan implementeras i ett socioteknisk system där IT-praktikern redan har en uppfattning om var sin plats är. Införandet av generativ AI utmanar därmed gränsen mellan manuella och automatiserade säkerhetsaktiviteter och tvingar IT-praktikern att omförhandla sin uppfattade plats. När empirin analyseras genom klassificeringsmodellens faser framträder ett tydligt mönster vilka presenteras nedan.

Plan
Det är de fundamentala expertuppgifterna i Plan-fasen såsom hotmodellering och riskbedömning som praktikerna upplever förblir mest resistenta mot AI-integration och därmed kräver fortsatt mänsklig närvaro. Studien hävdar att detta fynd visar var Grønsund och Aanestads (2020) ursprungliga resonemang om mänsklig referenspunkt inte längre räcker till för att förklara upplevelsen. I DevSecOps-kontexten är ”ground truth” mer komplex då det inte endast handlar om att verifiera AI:ns utdata mot kända fakta utan om att tillhandahålla en mänsklig domänspecifik säkerhetsbedömning. Respondent 5 exemplifierar detta med behovet av slutna AI-miljöer för att kunna mata in känslig sårbarhetsdata utan risk för informationsläckage eftersom en generell AI saknar den specifika kontext som krävs för att analysera en organisations faktiska hotlandskap (R5, 16).

Create
Samtidigt aktualiseras generativ AI mycket framgångsrikt i Create-fasen som ett stöd vid säker kodning. Här upplever IT-praktikerna att den kognitiva asymmetrin mellan människa och AI-verktyg utnyttjas optimalt vilket skapar en komplementär teamprestation. Respondenterna beskriver hur kodassistenter effektivt hanterar smådetaljer och repetitiva uppgifter vilket frigör utrymme för praktikern att fokusera på övergripande designbeslut (R1, 26; R1, 36; R1, 42). Denna arbetsfördelning innebär att yrkesrollen förskjuts från att enbart producera kod till att strategiskt planera och vägleda AI:ns output (R3, 36). Praktikerna upplever att denna avlastning från rutinarbete möjliggör att kända säkerhetshål snabbare kan

åtgärdas (R3, 34) vilket skapar förutsättningar för en ökad arbetstillfredsställelse i linje med Alenezi och Akours (2025) slutsatser.

Verify
I Verify-fasen som primärt utgörs av traditionella automatiserade verktyg samt mänsklig kodgranskning upplever IT-praktikerna däremot en påtaglig friktion. Trots att aktiviteterna är av helt olika karaktär anses generativ AI för närvarande inkapabel att autonomt ersätta någon av dem. Gällande den manuella kodgranskningen bottnar detta i bristande tillit då tekniken igen saknar förståelse för den övergripande domänkontexten vilket leder till att kritiska detaljer missas (R1, 34; R4, 30; Bedoya et al. 2024; Perry et al. 2023). Gällande tester som SAST förhindras en integration av att generativ AI fundamentalt utgörs av probabilistiska modeller med risk för hallucinationer (R6, 26; R6, 28; Chen et al. 2021) vilket krockar med den deterministiska exakthet som säkerhetstester kräver.

Preproduction
Slutligen visar empirin hur denna upplevelse sträcker sig hela vägen in i Preproduction-fasen genom förväntningar på framtida handlingsmöjligheter. Enligt Gibsons (1979) citerat i Valbø (2021) teori existerar en affordance oberoende av om den aktualiseras för stunden. Detta fenomen synliggörs när respondenterna diskuterar potentialen i att framtida AI-modeller kan genomföra offensiva säkerhetstester och storskaliga analyser för att proaktivt identifiera sårbarheter (R6, 16). Även om denna handlingspotential ännu inte är fullt aktualiserad i det dagliga arbetet bevisar det hur tekniken formar IT-praktikernas framtidsvisioner för säkerhetsarbetet.

Omförhandling av den sociotekniska rollen
En skeptisk läsare skulle kunna invända att detta empiriska mönster är förväntat. Det uppfattas ofta som självklart att komplexa planeringsuppgifter i Plan-fasen kräver mänsklig expertis. Men studiens vetenskapliga poäng är en annan. Det centrala fyndet är inte att en teoretisk skiljelinje existerar utan hur IT-praktikerna tvingas omförhandla sin upplevda plats i det sociotekniska systemet i takt med att tekniken mognar. Snarare än att betrakta GAI som en ersättare av manuellt eller redan automatiserat arbete upplever IT-praktikerna tekniken i en utpräglad hybridroll. Dels uppfattas ett användningsområde av GAI för att förstärka befintlig automation, vilket illustreras av respondent 3:s metod att låta GAI skriva och förbättra regler som sedan förbättrar de traditionella statiska verktygen (R3, 24). Men GAI ses även som en förstärkare för IT-praktikern i manuella processer genom att underlätta strategiska arkitekturbeslut (R1, 14) och snabbt filtrera enorma mängder sårbarhetsdata inför mänsklig bedömning (R6, 22). Det IT-praktikerna upplever är därmed ett tydligt exempel på vad Grønsund och Aanestad (2020) definierar som förstärkande arbete. Tekniken upplevs inte eliminera människan utan istället omforma säkerhetsarbetet till en iterativ feedbackloop där IT-praktikerns huvuduppgift inte längre är att exekvera utan att agera referenspunkt och styra både den mänskliga och automatiserade kapaciteten. Denna upplevelse är även i linje med både Fu et al. (2025) och Bedoya et al. (2024) mening om att IT-praktikern är djupt involverad och nödvändig för säkerhetens framgång.

Samtidigt visar empirin att denna upplevelse ur ett helhetsperspektiv är mer fragmenterad än vad litteraturens tekniska optimism antyder. Även om tekniken avlastar individen har flera respondenter ännu inte observerat någon markant övergripande skillnad i antalet fångade säkerhetsbrister (R1, 48; R5, 32). Nyttjandet är starkt personberoende och begränsas av att generella modeller sällan är anpassade för specifik defensiv säkerhet eller känslig data (R5, 14; R5, 26). Det analytiska värdet i denna observation är att upplevelsen av generativ AI i säkerhetsarbetet just nu inte är enhetlig utan djupt kontextuell. Fu et al. (2025) understryker

att AI-integration i DevSecOps kräver domänspecifika verktyg och empirin bekräftar att denna anpassning upplevs saknas i många organisationer (R5, 14; R5, 26). Studien bidrar därmed med ett empiriskt underlag som visar att den tekniska potentialen ständigt måste balanseras mot praktikerns upplevda verklighet i det sociotekniska systemet.

5.4 Begränsningar

Föreliggande studie bidrar med insikter kring ett aktuellt område inom informationssystem men det finns begränsningar som bör belysas i syfte att tillhandahålla transparens. Studiens urval av respondenter består av sex IT-praktiker verksamma i Sverige vilket är ett begränsat antal för att kunna redogöra ett representativt resultat för ett bredare perspektiv av IT-praktiker inom DevSecOps. Även om urvalet är kriteriebaserat begränsas möjligheten att identifiera mönster som är gemensamma för bredare grupper av IT-praktiker. Respondenterna rekryterades via författarnas professionella nätverk vilket medför en risk för att urvalet är påverkat av dess relation till författarna även om det inte har funnits en direkt relation.

En ytterligare begränsning rör den begreppsliga variation som uppstod kring termen säkerhetsaktiviteter under intervjuerna. Trots att begreppet definierades av författarna visade det sig att respondenterna tolkade det på olika sätt beroende på sin yrkesroll och organisatoriska kontext. Exempelvis sker olika tolkningar av liknande teknologiska begrepp mellan en utvecklare och en mer renodlad säkerhetsroll. Denna tolkningsvarians kan ha medfört att jämförbarheten mellan respondenternas svar begränsades något och att en mer standardiserad begreppsförklaring hade kunnat ge ett mer enhetligt empiriskt underlag ifall det redogjort före samtliga intervjuer.

Det bör även noteras att studien tillämpar en interpretativ ansats med semistrukturerade intervjuer vilket innebär att resultaten speglar respondenternas subjektiva upplevelser snarare än objektivt verifierbara fakta. Kodningen av det transkriberade materialet har genomförts gemensamt av författarna vilket stärker reliabiliteten, men det medför samtidigt att perspektiv som fallit utanför det etablerade analysramverket riskerar att ha förbisetts. Valet av en deduktiv tematisk analys med induktiva inslag innebär dessutom att de teman som identifierats i litteraturgenomgången till stor del har styrt hur empirin tolkades och presenterades, vilket kan vara en begränsning när man väljer att notera induktiva inslag.

Slutligen är det värt att påpeka att generativ AI befinner sig i en period av snabb teknisk utveckling. De verktyg och erfarenheter som respondenterna beskriver vid tidpunkten för intervjuerna kan ha förändrats på kort tid vilket innebär att studiens empiriska redogörelse är tidsbegränsad.

6. Slutsats

Denna studies syfte har varit att utforska och förstå IT-praktikerns subjektiva upplevelser av hur generativ AI omformar säkerhetsaktiviteter inom ramen för DevSecOps, vilket operationaliserats i forskningsfrågan: *Hur upplever IT-praktiker generativ AI:s påverkan på säkerhetsaktiviteter inom DevSecOps?* För att besvara denna fråga har ett socioteknisk perspektiv tillämpats där tre analytiska huvudområden samspelar: Affordance-teorin, human-in-the-loop och transformation av arbetsprocesser. Dessa perspektiv har prövats mot en interpretativ kvalitativ ansats bestående av sex semistrukturerade intervjuer med IT-praktiker verksamma i Sverige.

Studien visar att generativ AI inte erbjuder en enhetlig handlingspotential för IT-praktiker utan att upplevelsen av vad tekniken erbjuder är fundamentalt relationell och kompetensberoende. Seniora praktiker upplever breda handlingsmöjligheter som sträcker sig från strategiskt beslutsstöd till avancerade säkerhetsapplikationer, medan juniora praktiker möter en snävare och mer riskfylld potential. Generativ AI fungerar därmed som en hävstång för befintlig kompetens snarare än ett neutralt produktivitetsverktyg. Aktualiseringen av dessa handlingsmöjligheter framträder som ett iterativt och individuellt förhandlingsarbete där tillit, snarare än kostnad, är den avgörande faktorn. Det primära hindret för aktualisering är inte kompetensbristen i sig utan AI-systemens bristande förklarbarhet, vilket tvingar IT-praktikerna att kompensera genom erfarenhetsbaserad kontextualisering och iterativ verifiering snarare än att kunna förlita sig på transparenta systemegenskaper.

Samspelet mellan IT-praktikern och generativ AI framträder konsekvent som en asymmetrisk men komplementär relation där yrkesrollen förskjuts från manuell kodproduktion till strategisk planering och kritisk granskning. Denna ompositionering upplevs som meningsfull snarare än hotfull, då den kognitiva avlastningen från repetitiva uppgifter skapar utrymme för intellektuellt engagemang och domänexpertis. Studien visar dock att human-in-the-loop inte är en statisk egenskap utan en förmåga vars kvalitet varierar med individens kompetens och den arbetsbelastning AI-integrationen medför. IT-praktikern fyller inte bara rollen som kontrollant utan som en obligatorisk referenspunkt som kompenserar för AI:ns brist på domänspecifik säkerhetsförståelse. Denna referensfunktion upplevs som kognitivt krävande och ställer högre krav på praktikerns kompetens och uppmärksamhet än vad litteraturen hittills har belyst.

Generativ AI accelererar säkerhetsarbetet men skapar samtidigt en ny och allvarlig friktionspunkt. Hastigheten och volymen av AI-genererad kod överstiger ofta granskningskapaciteten, vilket riskerar att underminera hela syftet med DevSecOps och driva utvecklingen tillbaka till ett läge där snabb leverans återigen prioriteras högre än säkerhet. Klassificeringsmodellen belyser hur upplevelsen varierar systematiskt beroende på fas och aktivitetens karaktär. I Plan-fasen upplevs expertuppgifterna som hotmodellering resistenta mot AI-integration eftersom de kräver en domänspecifik säkerhetsbedömning som AI saknar förmåga att tillhandahålla. I Create-fasen upplevs tekniken positivt som stöd vid säker kodning och avlastar praktikern från repetitiva detaljer. I Verify-fasen uppfattas en påtaglig friktion då teknikens probabilistiska natur kolliderar med de deterministiska krav som säkerhetstestning ställer. I Preproduction-fasen uppfattas handlingsmöjligheter som offensiva säkerhetstester men är ännu inte aktualiserade i det dagliga arbetet.

IT-praktiker upplever generativ AI:s påverkan på säkerhetsaktiviteter inom DevSecOps som djupt kontextuell, asymmetrisk och beroende av fas. Upplevelsen formas av tre samverkande faktorer. Genom Affordance-teorin synliggörs hur IT-praktikern upplever teknikens handlingspotential olika beroende på kompetens, det vill säga vad tekniken uppfattas kunna erbjuda som funktionell handlingsmöjlighet och i vilken utsträckning denna möjlighet faktiskt aktualiseras i det dagliga säkerhetsarbetet. Genom human-in-the-loop framträder hur upplevelsen formas av hur generativ AI är implementerad och hur IT-praktikern arbetar med den, dels genom komplementär teamprestation där asymmetrins styrkor utnyttjas, dels som helhetsperspektiv och "ground truth" där praktikern kompenserar för teknikens brist på domänförståelse. Slutligen visar transformation av arbetsprocesser, systematiserat genom klassificeringsmodellen, att upplevelsen varierar beroende på vilken fas i DevSecOps-cykeln GAI implementeras och om aktiviteten historiskt hanterats som en expertuppgift eller en automatiserad process. Gemensamt för alla faser är att IT-praktikern inte upplevs som ersatt utan ompositionerad. Eftersom GAI främst avlastar vardagliga arbetsuppgifter snarare än att ta över komplexa säkerhetsaktiviteter, har praktikern kunnat gå från kodproducent till referenspunkt och från utförare till strategisk vägledare.

6.1 Förslag till vidare forskning

Den genomförda studiens resultat och identifierade begränsningar öppnar för flera intressanta riktningar och tolkningar för framtida forskning. Initialt vore ett naturligt steg att replikera studien med ett större och bredare representativt urval av respondenter. Det kan vara av intresse att inkludera representanter verksamma i andra länder eller med bredare spridning av organisationer, roller och erfarenhetsnivåer. Eftersom studiens respondenter uteslutande är verksamma i Sverige är det oklart i vilken utsträckning upplevelserna av handlingsmöjligheter och human-in-the-loop dynamiken är kulturellt specifika eller mer generellt förekommande i DevSecOps-kontexter. En komparativ studie mellan till exempel nordiska och andra organisationskulturer hade kunnat belysa hur institutionella och kulturella faktorer påverkar IT-praktikerns upplevelser av GAI mer nyanserat.

Studien identifierar en skiljaktighet i uppfattning och aktualisering mellan seniora och juniora IT-praktiker. I framtida forskning vore det av stort värde att genomföra en studie som jämför liknande forskningsfråga mellan olika kompetensnivåer av IT-praktiker. En sådan studie skulle kunna besvara mer nyanserat på skiljaktigheter i kompetensnivå.

Slutligen pekar studiens fynd på att human-in-the-loop inte är en stabil egenskap utan en förmåga vars kvalitet varierar med individens kompetens och den arbetsbelastning som AI-integrationen medför. Det vore intressant att framtida forskning undersöker organisatoriskt beslutsfattande och utbildning som kan stärka IT-praktikerns förmåga att agera som en referenspunkt till tekniken. En sådan studie hade kunnat ge konkreta rekommendationer till organisationer som befinner sig i processen att integrera generativ AI i sina DevSecOps-pipelines.

Appendix 1: AI-bidragsredogörelse

Nyttjade verktyg: Google Gemini och Sunet Scribe

Grad av användning: I den genomförda studien har AI-verktyg använts som ett interaktivt stöd i syfte att upprätthålla en hög analytisk kvalitet samt säkerställa hög språklig kvalitet. Verktygen har varit till hjälp för att upprätthålla en röd tråd genom uppsatsen där Google Gemini nyttjats som en källa till kritisk återkoppling genom att skapa ”Gems” anpassade för studien med stående prompter. Sunet Scribe har nyttjats vid transkribering för att höja korrekthet och tidsbespara processen.

Google Gemini har använts genomgående i samtliga kapitel för:

- Generering av idéer vid tankespridning.
- Feedback för att upprätthålla en röd tråd.
- Förbättrande av språk (Från enkla manuellt skrivna stycken till akademiskt språk med manuella justeringar).
- Förbättrad meningsstruktur i syfte att undvika upprepning och felstrukturerade meningar.

Exempel på prompts som nyttjats i Google Gemini Gems:

- ”Granska min nuvarande forskningsfråga: XXX. Analysera om den är för bred för en uppsatsen eller om den riskerar att bli rent deskriptiv.”
- ” Jag planerar att använda semistrukturerade intervjuer för att undersöka vårt ämne. Vad bör jag tänka på? Ge mig en checklista på vad jag måste motivera i metodkapitlet för att genomföra intervjuerna väl.”
- "Agera som en strikt examinator under slutseminariet på LUSEM. Läs igenom mitt diskussionsavsnitt och förklara sammanhanget och strukturen som en del av uppsatsen."
- ”Följande ord låter vardagligt. När jag säger ord XXX menar jag YYY. Vad finns det för akademiska alternativ till ordet”
- ”Se över meningsuppbyggnad och omformulera felaktigheter i meningsstruktur. Ditt svar ska vara anpassat till uppgiften, men ändra inget innehåll.”
- ”Poängtera stavfel i följande stycke”

Sunet Scribe har använts för att:

- Transkribera inspelade intervjuer där automatisk konvertering av ljudfiler till text nyttjats. Resultatet har därefter genomgått en manuell kontroll och korrigeringar av utfyllnadsord samt delar som bör plockas bort för att säkerställa anonymitet

Referenser

AIS. (2025). Senior Scholars List of Premier Journals, https://aisnet.org/research/seniorscholarsbasket/ [Hämtad 21 april 2026]

Alenezi, M., & Akour, M. (2025). AI-Driven Innovations in Software Engineering: A review of current practices and future directions, *Applied Sciences*, vol. 15, no. 3, 1344 https://doi.org/10.3390/app15031344

Bain & Company. (u.å.). DevSecOps, https://www.bain.com/insights/books/doing-agile-right/resources/enablers-of-agile-software-development/devsecops/ [Hämtad 28 april 2026]

Banh, L., & Strobel, G. (2023). Generative Artificial Intelligence, *Electronic Markets*, vol. 33, no. 63, https://doi.org/10.1007/s12525-023-00680-1

Bauer, K., von Zahn, M., & Hinz, O. (2023). Expl(AI)ned: The impact of explainable artificial intelligence on users' information processing, *Information Systems Research*, vol. 34, no. 4, pp.1582-1602, https://doi.org/10.1287/isre.2023.1199

Bedoya, M., Palacios, S., Díaz-López, D., Laverde, E., & Nespoli, P. (2024). Enhancing DevSecOps practice with Large Language Models and Security Chaos Engineering, *International Journal of Information Security*, vol. 23, pp.3765-3788, https://doi.org/10.1007/s10207-024-00909-w

Bryman, A. (2012). Social Research Methods, Oxford: Oxford University Press

Buchanan, B. G. (2005). A (Very) Brief History of Artificial Intelligence, *AI Magazine*, vol. 26, no. 4, pp.53-60, https://www.proquest.com/docview/208132026?accountid=12187&sourcetype=Scholarly%20Journals [Hämtad 15 april 2026]

Camacho, N, G. (2024). Unlocking the Potential of AI/ML in DevSecOps: Effective strategies and optimal practices, *Journal of Artificial Intelligence General Science*, vol. 2, no. 1, https://doi.org/10.60087/jaigs.v2i1.p89

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. de O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, R., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., & Zaremba, W. (2021). Evaluating Large Language Models Trained on Code, preprint, https://doi.org/10.48550/arXiv.2107.03374

Corso, V., Mariani, L., Micucci, D., & Riganelli, O. (2024). Generating Java Methods: An empirical assessment of four AI-based code assistants, 2024 IEEE/ACM 32nd International Conference on Program Comprehension (ICPC), pp.13-23, https://doi.org/10.1145/3643916.3644402

CrowdStrike. (2026). 2026 Global Threat Report, https://www.crowdstrike.com/en-us/global-threat-report/ [Hämtad 16 maj 2026]

Cui, K. Z., Demirer, M., Jaffe, S., Musolff, L., Peng, S., & Salz, T. (2026). The Effects of Generative AI on High-Skilled Work: Evidence from three field experiments with software developers, *Management Science*, pp.1-13, https://doi.org/10.1287/mnsc.2025.00535

Daniotti, S., Feng, X., Neffke, F., & Wachs, J. (2026). Who is Using AI to Code? Global diffusion and impact of generative AI, *Science*, vol. 391, no. 6787, pp.831-835, https://doi.org/10.1126/science.adz9311

Doshi-Velez, F., & Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning, preprint, pp.1-13, https://doi.org/10.48550/arXiv.1702.08608

Friedman, N. (2021). Introducing GitHub Copilot: your AI pair programmer, https://github.blog/news-insights/product-news/introducing-github-copilot-ai-pair-programmer/ [Hämtad 28 april 2026]

Fu, M., Pasuksmit, J., & Tantithamthavorn, C. (2025). AI for DevSecOps: A landscape and future opportunities, *ACM Transactions on Software Engineering and Methodology*, vol. 34, no. 4, pp.1-61, https://doi.org/10.1145/3712190

Gall, M. & Pigni,F. (2022). Taking DevOps mainstream: A critical review and conceptual framework, *European Journal of Information Systems*, vol. 31, no. 5, pp.548-567, https://doi.org/10.1080/0960085X.2021.1997100

Gartner. (2016). DevSecOps: How to Seamlessly Integrate Security Into DevOps [pdf], https://cdn2.hubspot.net/hubfs/1958393/White_Papers/devsecops_how_to_seamlessly__315283.pdf

Gartner. (2026). Gartner Says Worldwide AI Spending Will Total $2.5 Trillion in 2026, https://www.gartner.com/en/newsroom/press-releases/2026-1-15-gartner-says-worldwide-ai-spending-will-total-2-point-5-trillion-dollars-in-2026 [Hämtad 28 april 2026]

Grønsund, T., & Aanestad, M. (2020). Augmenting the algorithm: Emerging human-in-the-loop work configurations, *Journal of Strategic Information Systems*, vol. 29, no. 2, https://doi.org/10.1016/j.jsis.2020.101614

Haenlein, M., & Kaplan, A. (2019a). A Brief History of Artificial Intelligence: On the past, present, and future of artificial intelligence, *California Management Review*, vol. 61, no. 4, pp.5-14, https://doi.org/10.1177/0008125619864925

Haenlein, M., & Kaplan, A. (2019b). Siri, Siri, in My Hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence, *Business Horizons*, vol. 62, no. 1, pp.15-25, https://doi.org/10.1016/j.bushor.2018.08.004

Hemmer, P., Schemmer, M., Kühl, N., Vössing, M., & Satzger, G. (2025). Complementarity in human-AI collaboration: Concept, sources, and evidence, *European Journal of Information Systems*, vol. 34, no. 6, pp.979-1002, https://doi.org/10.1080/0960085X.2025.2475962

Hemon-Hildgen, A. & Rowe, F. (2022). Conceptualising and defining DevOps: A review for understanding, not a framework for practitioners, *European Journal of Information Systems*, vol. 31, no. 5, pp.568-574, https://doi.org/10.1080/0960085X.2022.2100061

Hoffmann, M., Boysel, S., Nagle, F., Peng, S., & Xu, K. (2025). Generative AI and the Nature of Work, Harvard Business School Strategy Unit Working Paper No. 25-021, CESifo Working Paper Series No. 11479, https://doi.org/10.2139/ssrn.5007084

Kalota, F. (2024). A Primer on Generative Artificial Intelligence, *Education Sciences*, vol. 14, no. 2, https://doi.org/10.3390/educsci14020172

Markus, M. L., & Silver, M. S. (2008). A Foundation for the Study of IT Effects: A new look at DeSanctis and Poole's concepts of structural features and spirit, *Journal of the Association for Information Systems*, vol. 9, no. 10, pp.609-632, https://doi.org/10.17705/1jais.00176

McKinsey & Company. (2023). Unleashing developer productivity with generative AI, https://www.mckinsey.com/capabilities/tech-and-ai/our-insights/unleashing-developer-productivity-with-generative-ai [Hämtad 28 april 2026]

Mohammed, K. I., Shanmugam, B., & El-Den, J. (2025). Evolution of DevSecOps and Its Influence on Application Security: A systematic literature review, *Technologies*, vol. 13, no. 12, https://doi.org/10.3390/technologies13120548

Mohan, V., & Othmane, L. B. (2016). SecDevOps: Is It a Marketing Buzzword? - Mapping research on security in DevOps, *2016 11th International Conference on Availability, Reliability and Security (ARES)*, https://doi.org/10.1109/ARES.2016.92

Myers, M. D., & Avison, D. (2002). Qualitative Research in Information Systems, London: SAGE

Myrbakken, H. & Colomo-Palacios, R. (2017). DevSecOps: A multivocal literature review, *Software Process Improvement and Capability Determination*, vol. 770, pp.17-29, https://doi.org/10.1007/978-3-319-67383-7_2

Oates, B. J., Griffiths, M. & McLean, R. (2022). Researching Information Systems and Computing, 2:a upl, London: SAGE

Pakalapti, N. Konidena, B, K. Mohamed, I, A. (2023). Unlocking the Power of AI/ML in DevSecOps: Strategies and best practices, *Journal of Knowledge Learning and Science Technology*, vol. 2, no. 2, https://doi.org/10.60087/jklst.vol2.n2.p188

Perry, N., Srivastava, M., Kumar, D., & Boneh, D. (2023). Do Users Write More Insecure Code with AI Assistants?, *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, https://doi.org/10.1145/3576915.3623157

Prates, L., & Pereira, R. (2024). DevSecOps practices and tools. *International Journal of Information Security,* vol. 24, no. 1, pp.1-25, https://doi.org/10.1007/s10207-024-00914-z

Rahman, A. A. U., & Williams, L. (2016). Software Security in DevOps: Synthesizing practitioners' perceptions and practices, *2016 IEEE/ACM International Workshop on Continuous Software Evolution and Delivery (CSED)*, https://doi.org/10.1145/2896941.2896946

Schultze, U., & Avital, M. (2010). Designing interviews to generate rich data for information systems research, *Information and Organization*, vol. 21, no 1, pp.1-16, https://doi.org/10.1016/j.infoandorg.2010.11.001

Sergeyuk, A., Golubev, Y., Bryksin, T., & Ahmed, I. (2025). Using AI-based Coding Assistants in Practice: State of affairs, perceptions, and ways forward, *Information and Software Technology*, vol. 178, https://doi.org/10.1016/j.infsof.2024.107610

Shahin, M., Babar, M.A., & Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A systematic review on approaches, tools, challenges and practices, *IEEE Access*, vol. 5, pp. 3909-3943, https://doi.org/10.1109/ACCESS.2017.2685629

Stack Overflow. (2025). 2025 Developer Survey, https://survey.stackoverflow.co/2025/ [Hämtad 28 april 2026]

Strong, D. M., Volkoff, O., Johnson, S. A., Pelletier, L. R., Tulu, B., Bar-On, I., Trudel, J., & Garber, L. (2014). A Theory of Organization-EHR Affordance Actualization, *Journal of the Association for Information Systems,* vol. 15, no. 2, pp. 53-85, https://doi.org/10.17705/1jais.00353

Valbø, B. (2021). The IS-Notion of Affordances: A Mapping of the Application of Affordance Theory in Information Systems Research, *Selected Papers of the IRIS, Issue Nr 12*, https://aisel.aisnet.org/iris2021/2 [Hämtad 4 maj 2026]

von Zahn, M., Liebich, L., Jussupow, E., Hinz, O., & Bauer, K. (2025). Knowing (Not) to Know: Explainable Artificial Intelligence and Human Metacognition. *Information Systems Research*, https://doi.org/10.1287/isre.2024.1431

World Economic Forum. (2026). Global Cybersecurity Outlook 2026 [pdf], https://reports.weforum.org/docs/WEF_Global_Cybersecurity_Outlook_2026.pdf

Zhao, X., Clear, T., & Lal, R. (2024). Identifying the Primary Dimensions of DevSecOps: A multi-vocal literature review, *Journal of Systems and Software*, vol. 214, https://doi.org/10.1016/j.jss.2024.112063

Zhou, X., Mao, R., Zhang, H., Dai, Q., Huang, H., Shen, H., Li, J., & Rong, G. (2023). Revisit security in the era of DevOps: An evidence-based inquiry into DevSecOps industry, *IET Software*, vol. 17, pp.435-454, https://doi.org/10.1049/sfw2.12132