



# Fenomenet "The ITSM Iron Triangle" – en medlare eller uppvigglare?

Kandidatuppsats i Informatik

Keränen, Robert  
Öberg, Jonathan

2024KANI28



HÖGSKOLAN  
I BORÅS

**Svensk titel:** Fenomenet ”THE ITSM IRON TRIANGLE” – en medlare eller uppvigglare?

**English title:** The phenomena “THE ITSM IRON TRIANGLE” – a mediator or an agitator?

**Utgivningsår:** 2024

**Författare:** Keränen, Robert och Öberg, Jonathan

**Handledare:** Cronholm, Stefan

**Examinator:** Rittgen, Peter

## **Abstract**

IT-organizations that manages development and change of IT-systems, both for the internal and external enterprises is bound to incorporate both error handling and deployment processes in their businesses. The viewpoint of these processes is widely differentiated by the different teams depending on which part of the IT-organization one belongs to. Developers find these processes less prioritized, tedious and of lower status while managers of IT see them as a critical and highly prioritized task. The different viewpoints derive from the work contribution of the different work groups to cocreate value to the enterprise through IT. The conflict between these groups has a long history and appears because of both parties' separate perspective and this study aims to investigate and discuss how enterprises should approach this conflict with the weaknesses of their business. The study presents a model which aims at helping the enterprise with the investigation into areas of improvement for the business related to terminology and strategies needed to override the conflict at hand.

**Keywords:** IT-service management, ITIL, System development, System management, ITSM Iron Triangle, Incident management, Problem management, Change enablement, Conflict

## Sammanfattning

IT-organisationen som hanterar utveckling och förändring av IT-system för intern eller extern verksamhet kommer oundvikligen att behöva arbeta inom felhanterings- och åtgärdsrealiseringsprocesser. Synen på dessa processer skiljer sig markant åt beroende på vilken del av IT-organisationen man tillhör. Utvecklarna ser det som en mindre prioriterad och arbetsam arbetsuppgift med lägre status medan förvaltare ser det som en kritisk uppgift med hög prioritet. Skillnaden i syn grundas i vad de olika arbetsgrupperna anser vara sitt bidrag till att samskapa värde för verksamheten genom IT. Konflikten som har varit bestående under lång tid inom området uppstår till följd av gruppernas olika perspektiv och det undersöks och diskuteras i denna studie hur verksamheter bör påbörja arbetet med att lokalisera sina svagheter inom hanteringen av denna långtgående konflikt. Studien presenterar en modell som kan ligga till grund för det undersökande arbetet verksamheter står inför där terminologi och strategier presenteras för att upptäcka utvecklingsområden för verksamheterna.

**Nyckelord:** IT-tjänsthantering, ITIL, Systemutveckling, Systemhantering, ITSM triangeln, Incidenthantering, Problemantering, Förändringsledning, Konflikt

# Innehåll

1	Inledning.....	- 1 -
1.1	Bakgrund .....	- 1 -
1.2	Forskningsöversikt .....	- 3 -
1.3	Problemdiskussion.....	- 4 -
1.4	Problemformulering .....	- 5 -
1.5	Syfte och forskningsfråga.....	- 5 -
1.6	Målgrupp för arbetet.....	- 6 -
2	Metod .....	- 7 -
2.1	Forskningsstrategi .....	- 7 -
2.2	Forskningsmetod .....	- 8 -
2.3	Urval av respondenter .....	- 9 -
2.4	Datansamling.....	- 9 -
2.5	Dataanalys .....	- 10 -
2.6	Etiskt ställningstagande.....	- 11 -
2.7	Metodreflektion .....	- 11 -
3	Teori .....	- 13 -
3.1	Terminologi.....	- 13 -
3.2	Strategi.....	- 13 -
3.2.1	ITIL .....	- 14 -
3.2.2	DevOps.....	- 14 -
3.3	Kommunikation.....	- 15 -
3.4	Konflikt .....	- 15 -
4	Resultat.....	- 17 -
4.1	Incident.....	- 17 -
4.1.1	Terminologi.....	- 17 -
4.1.2	Strategi .....	- 17 -
4.1.3	Kommunikation.....	- 17 -
4.1.4	Konflikt .....	- 18 -
4.2	Problem .....	- 18 -
4.2.1	Terminologi.....	- 18 -
4.2.2	Strategi .....	- 18 -
4.2.3	Kommunikation.....	- 20 -
4.2.4	Konflikt .....	- 20 -
4.3	Change.....	- 20 -
4.3.1	Terminologi.....	- 20 -
4.3.2	Strategi .....	- 20 -
4.3.3	Kommunikation.....	- 21 -
4.3.4	Konflikt .....	- 21 -
5	Analys & Diskussion.....	- 22 -
5.1	Terminologi.....	- 22 -
5.2	Strategi.....	- 23 -
5.3	Kommunikation.....	- 24 -
5.4	Konflikt .....	- 24 -
5.5	Sammanfattning av analys.....	- 25 -
6	Slutsatser .....	- 27 -
7	Referenser.....	- 28 -
8	Bilagor.....	- 31 -
8.1	Bilaga 1: Arbetsblad för sökord .....	- 31 -
8.2	Bilaga 2: Intervjuguide .....	- 32 -

8.3	Bilaga 3: Respondentförteckning .....	- 34 -
8.4	Bilaga 4: Begreppslista.....	- 35 -
8.5	Bilaga 5: Generativ AI-Intyg.....	- 36 -

# 1 Inledning

Detta kapitel syftar till att introducera dig som läsare till forskningsområdet samt uppsatsens syfte och problemformulering. Detta för att ge dig en klar bild över området som det ser ut idag och förtydliga ämnet som kommer att undersökas.

## 1.1 Bakgrund

Inom IT-organisationer finns det ofta två läger som var för sig tillhandahåller tjänster som förser kunder med IT-lösningar. På ena sidan har vi IT-utvecklarna (utvecklare) vilka har till uppgift att ta fram mjukvarulösningar, testa dem och få systemet leveransklart. Utvecklarna har kunskap om olika programmeringsspråk och förståelse för teknologin som stödjer systemet i sin leverans. På andra sidan har vi IT-driften (driften) som efter det att mjukvarulösningen är färdig tar över för att förvalta systemet så att det fortsätter att leverera det tänkta värdet som kunden efterfrågar. Även om de båda sidorna tidigare uppfattats utgöra vattentäta skott mellan varandra så finns det flera faktorer som får dem att kunna dra nytta av varandras kunskap och gå in på varandras område. Till exempel kan utvecklarna dra nytta av information om hur systemet är driftsatt och används för att utveckla mer optimala lösningar för de teknologier som används i driften. Likväl behöver driften kunskap och förståelse för hur systemet är utvecklat för att lättare kunna utföra underhåll eller förse utvecklarna med rätt förutsättningar för att rätta till eventuella brister i systemet. (Walker 2022)

Enligt Agutter (2020) är området IT-service management (ITSM) ett samlingsbegrepp för de processer som har till syfte att optimera och effektivisera utveckling och drift i verksamheter och bör idag vara en del av varje organisations arbetsprocesser. Information Technology Infrastructure Library (ITIL) är det ramverk som globalt blivit erkänt som best practice för att organisera verksamheters arbete inom de många ITSM-processerna (ibid.). I denna uppsats går vi djupare in på verksamhetsprocesserna inom felhantering och åtgärdsrealisering. De processområden som inom ITIL berör dessa områden benämns: Incident management, Problem management och Change enablement (ibid.).

I och med att utveckling och drift traditionellt har bedrivits som två separata undergrupper inom IT-organisationer så har tvärfunktionellt samarbete varit problematiskt (Swanson & Beath 1989; Gregory, Henfridsson & Ruch 2018). Drift har dessutom alltid setts som den minst prestigefyllda och mindre värdefulla uppgiften (Swansons & Beath 1989). Samarbetsproblemen har blivit än mer märkbara då utveckling idag fokuserar på snabb utveckling av ny funktionalitet i systemen (Ravichandran, Taylor & Waterhouse, 2016) och driften arbetar för att underhålla ett stabilt system med så lite förändring som möjligt (Gregory, Henfridsson & Ruch 2018). Vid arbete med drift uppstår ofta frågan ”what changed” (Ravichandran, Taylor & Waterhouse, 2016, s.121) när något gått fel. Denna frågeställning syftar vanligen på en produktionssatt systemförändring i verksamhetens IT-system och denna formulering kommer med en underliggande mening om att ställa utvecklarna ansvariga för den aktuella kapacitetsförlusten som upplevs i systemet (Ravichandran, Taylor & Waterhouse 2016). En kapacitetsförlust i denna kontext kan vara till exempel minskad möjlighet att ta emot trafik på sin hemsida, minskad processorkraft för beräkning och bearbetning av data i systemet eller liknande negativ systempåverkan. Detta ger upphov till en möjlig dissonans mellan grupperna då de saknar gemensamma mål, processer, kunskaper och tillvägagångssätt för att bemöta situationen (ibid.). I verksamheter finns vanligen ett upplevt behov från utvecklarsidan av IT-organisationen att arbetet behöver ligga i framkant för att möta nya affärsmässiga utmaningar och därför upplever ett behov av att snabbt kunna genomföra förändringar i befintliga system. I arbetet möter utvecklarna motstånd från driftssidan när idéer på förändring

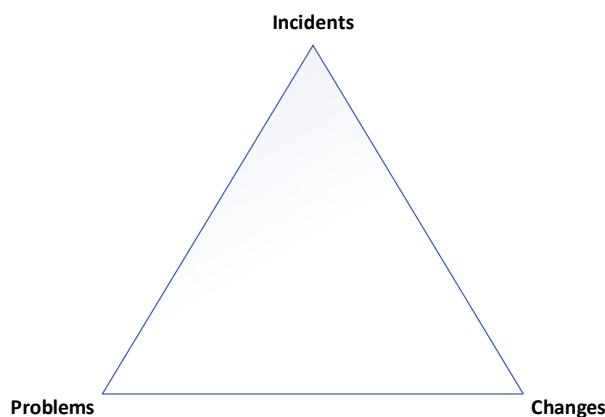
lyfts fram. Driftens primära uppgift utsätts för påtryckningar då deras primära uppgift innefattar att bland annat supportera användare, underhålla data och säkerställa systemsäkerheten. Det uppstår till följd av detta lätt en konflikt mellan grupperna då de har varierande intressegrund i frågan. Dock har dessa två arbetsgrupper ett gemensamt ansvar för helheten kring området, utvecklarna har fokus på att accelerera utveckling för att skapa ett marknadsövertag medan driftens vill skydda verksamheten ifrån skadlig förändring (McCarthy, Herger, Khan & Belgodere 2015). McLean (2012) skriver även i boken *The ITSM Iron Triangle* (ITSM Triangeln) om detta och i ett utdrag ur boken målas ett exempel upp kring hur en systemförändring relaterar till nya utmaningar för en verksamhet.

*I slammed my hand flat on the table and told the room, "Settle down. We need to get to work. I need an update on what we know about what happened to take the business off the network. Who's got some updates?"*

*Nicola, leader of the firewall team, was chugging a can of energy drink and had two more on the table in front of him. I remembered that he'd been putting in a number of approved changes last night, so he probably hadn't been to sleep yet. (McLean 2012, Kap.: Into the fire)*

Idag är Development (Utveckling) and Operations (drift) (DevOps) ett erkänt arbetssätt som försöker överbrygga gapet mellan dessa två grupper. Gemensamt för de båda grupperna är deras syfte att leverera ett system som är av god kvalitet och uppfyller kundens krav (Walker 2022). DevOps bidrar till att motverka konflikten mellan IT-organisationers utvecklar- och driftsteam är emellertid något otydligt (Jabbari, Ali, Petersen och Tanveer 2016). Frågan som uppstår är; Vilka verktyg har DevOps faktiskt bidragit med för att motverka motsättningarna?

Denna uppsats har för avsikt att genomföra en studie inom området ITSM runt konflikten som kan uppstå inom fenomenet ITSM Triangeln (Figur 1) med fokus riktat på aspekterna Incident, Problem och Change samt de inbördes relationerna mellan dessa aspekter. Anledningen till att valet föll på ITSM-triangeln som en modell för att identifiera kommunikationsbrister är att den omfattar de tre vanligaste processerna inom ITSM. ITSM Triangeln är ett fenomen som har observerats inom kontexten för ramverket ITIL och triangeln kan användas för att beskriva relationerna mellan aspekterna och deras handlingsmönster samt interaktion med varandra (McLean 2012). Relationen mellan dessa aspekter kan skapa rundgång i systemet. När det har uppstått incidenter och orsaken har blivit känd genom en problemutredning föranleder det en förändring av systemet genom en change för att åtgärda och bygga bort problematiken från systemet. Detta kan i sin tur leda till nya incidenter som även dessa behöver hanteras i nya problem (Agutter 2020).



Figur 1: The ITSM iron triangle (McLean 2012)

Ytterligare ett perspektiv som belyses av triangeln är att kända problem kan hanteras genom att göra ett aktivt val i att inom organisationen acceptera problemet och tills vidare hantera de incidenter som följer av beslutet. Problemhanteringen bidrar då med temporära lösningar fram tills dess att en lämplig permanent lösning går att implementera (McLean 2012). Kopplingarna inom ITSM triangeln (McLean 2012) har beskrivits i ITIL-ramverket och finns definierade i delarna Incident management, Problem management samt Change enablement (Agutter 2020).

## 1.2 Forskningsöversikt

Genom en initial litteraturgranskning av vetenskapliga artiklar har noterats att indikationer förekommer på att de aspekter som innefattas inom ITSM triangeln faktiskt är applicerbara över flertalet applikationer och organisationer. I Tabell 1 nedan sammanställs termerna som använts i ett antal olika publikationer och dessa termer kategoriseras utifrån ITIL ramverkets standardbenämning av termerna. Det framkommer med tydlighet att det förekommer en mängd olika begrepp för samma aspekter inom fenomenet ITSM triangeln i litteraturen. Det kan ligga till grund för kommunikationsbrister inom IT-organisationer. I en artikel beskriver Eaddy, et al. (2008) processen för att hitta och åtgärda ett fel i ett mjukvarusystem enligt följande: Vanligtvis skapas en buggrapport upp för att fånga in aktuella problem i produktionsmiljön för applikationer. Dessa buggrapporter analyseras sedan för att en utvecklare eventuellt skall påbörja ett felsökningsarbete i källkoden för att lokalisera det bekymmer som är orsaken till denna bugg. När felet är identifierat och rättningen är utvecklad produktionsstöts rättningen till slutanvändarna och felet rapporteras som löst. Även Wang och Cao (2021) skriver om samma tillvägagångssätt fast använder sig i stället av orden fel, utredning och åtgärd. Där primärkällorna skapar upp en felrapport som sedan utreds av utvecklarna och åtgärdas via förändringar i applikationen. Även termer som kapacitetsförlust och släpp förekommer i litteraturen (Waller, Ehmke & Hasselbring 2015).

Tabell 1: Olika artiklars benämningar på ITSM triangelns delar

ITIL termer	Incident	Problem	Change
Eaddy et al. (2008)	Bugg	Bekymmer	Rättning
McLean (2012)	Incident	Problem	Change
Waller, Ehmke & Hasselbring (2015)	Kapacitetsförlust	Problem	Släpp
Wang & Cao (2021)	Fel	Utredning	Åtgärd

En definition av ITSM ges av Axelos (2019) på engelska och kan översättas på följande sätt: En uppsättning av verksamhetsförmågor som möjliggör värde för kunderna i form av tjänster. ITSM omfattar processområden och förmågor som möjliggör för en verksamhets IT-tjänster att samskapa värde tillsammans med verksamheten samt leverera värde externt. Det finns ett allmänt känt sätt att utföra dessa processer för att hålla en god kvalitet på sitt arbete och detta kallas för best practice inom det valda området (Cronholm & Persson 2016; Agutter 2020).

ITIL-ramverket är en best practice som ursprungligen tagits fram av praktiker inom brittiska staten, genom att sammanställa de arbetssätt inom ITSM som resulterat i att IT-baserade tjänster och IT-projekt uppnått god kvalitet. ITIL är ingen standard och styr inte verksamheter i hur de skall arbeta med ITSM utan skall ses som ett ramverk vilket ger stöd i att utveckla verksamhetens egna ITSM-processer (Agutter 2020).

Ett försök att samordna ITSM-processer och möjliggöra samarbete mellan utvecklare och drift tar sig uttryck i form av utvecklingsmetodik DevOps. Ett generellt bekymmer beträffande DevOps är att det inte finns en allmänt vedertagen definition av vad DevOps innebär (Beetz & Harrer 2022). Jabbari et al. (2016) utförde en litteraturgranskning i ett försök att finna om det

inom teorin fanns en gemensam definition av DevOps och fann då att åtta centrala komponenter var återkommande inom de definitioner som återfanns i de studerade artiklarna. Dessa komponenter kan sammanfattas till att konceptet DevOps innefattar utvecklare och drift och anses som ett paradigm eller metod för att främja samarbete och kommunikation för att skapa effektiva team samt överbrygga gapet mellan utvecklare och drift. Vidare anges DevOps som ett modernt sätt att hantera mjukvaruutveckling där hänsyn tas till det ömsesidiga beroendet mellan utveckling och drift för att uppnå konvergens. Författarna finner det återkommande att DevOps uppges omfatta arbetssätt och principer som möjliggör att utvecklingsarbete och produktionsättning av detsamma automatiseras i den omfattning det är möjligt för att minska tiden mellan släpp av ny programkod till produktionsmiljö. Generellt ligger det ett fokus på arbetets kontinuitet och förutsägbarhet i utförande, kvalitetssäkrande och ständigt förbättringsarbete (ibid.). Jabbari et al. (2016) kommer med ett förslag på definition av DevOps efter att ha utfört sin studie och den lyder;

*"DevOps is a development methodology aimed at bridging the gap between Development (Dev) and Operations, emphasizing communication and collaboration, continuous integration, quality assurance and delivery with automated deployment utilizing a set of development practices." (Jabbari et al. 2016, s. 6)*

### 1.3 Problemdiskussion

Rigoröst arbete till trots så uppstår defekter i mjukvarusystem. Att identifiera dessa defekter och reducera deras påverkan för slutanvändarna är av hög prioritet för mjukvaruutvecklare (Eaddy, et al. 2008). I ITIL-ramverket presenteras bland annat arbetssätt för att hantera processområdena inom ITSM triangeln. Dessa processområden beskriver hur organisationer strukturerat bör arbeta med den felsökning- och åtgärdsprocess som ITSM Triangel belyser (Agutter 2020).

Det har framkommit i litteraturen som presenterats i kapitel 1.2 (*Forskningsöversikt*) att det inom IT-sektorn saknas en gemensam terminologi för att benämna och uttrycka sig kring de begrepp som förenas med incident management, problem management och change enablement. Detta kan i sin tur leda till kommunikationsbrister inom IT-organisationen. I kombination med att utvecklare och driften har olika strategier, det vill säga olika planer alternativt tillvägagångssätt (Mintzberg 1987), för att uppnå sina mål för att tillhandahålla mjukvara till kunden kan detta bidra till den konflikten som ofta förekommer mellan utveckling och drift. I Tabell 1 presenteras ett urval av olika benämningar för dessa tre processområden. Detta föranleder ett behov av att se över hur vanligt förekommande de relationer som ITSM triangel illustrerat är för att förstå om olika organisationer upplever liknande utmaningar. Men utan ett gemensamt språkbruk så kan det vara svårt att överbrygga kommunikationsbrister i verksamheterna.

Detta gjorde det intressant att utreda om en IT-organisation som har valt att försöka överbrygga kommunikationsbristerna genom att införa DevOps upplever att implementationen har lyckats lösa den konflikt som är i fokus för studien. Det som DevOps försöker åtgärda genom att svetsa samman organisationen är just kommunikationssvårigheter. Samt att verksamheten genom gemensamma strategier och en gemensam arbetsyta för utvecklare och drift gör det möjligt att förenkla samarbetet inom det dagliga arbetet mellan utveckling och drift.

## 1.4 Problemformulering

Att konflikter uppstår i kommunikationen mellan utvecklare och drift av IT-system är idag ett faktum i flera verksamheter. Med så skilda perspektiv och målsättningar med sitt arbete kan det anses oundvikligt att dessa två grupper ofta hamnar i konflikt med varandra. Men då båda grupperna i slutändan har ett gemensamt helhetsansvar för verksamheten finns det en poäng i att försöka överbrygga deras intressen och få grupperna att samexistera och samskapa värde för verksamheten. Det finns ett tydligt gemensamt intresse och ansvar för verksamhetens fortlevnad och utveckling som skulle kunna dra nytta av ett bredare perspektiv.

Problemet som ska studeras formuleras enligt följande ”Bristen på gemensamma begrepp och strategier leder till kommunikationsproblem som ligger till grund för konflikten mellan utvecklare och drift”. Studiens bidrag till området blir därför att utifrån ITSM triangeln försöka hitta gemensamma begrepp respektive strategier som skulle kunna överbrygga den konflikt som idag kan upplevas inom IT-organisationer.

## 1.5 Syfte och forskningsfråga

Studien har försökt identifiera de områden som ligger till grund för konflikten som kan uppstå i och med felhantering och åtgärdsrealisering inom IT-system i verksamheten. Dessa konfliktområden är i stort behov av ökad samsyn, en gemensam terminologi samt likvärdiga strategier för att möjliggöra detta. ITIL har lyckats belysa området och är generellt accepterat inom driftsorganisationer runt om i världen. Men då utvecklarens perspektiv på området är i behov av snabbfotad förändring och ser mest dokumentation som ett onödigt hinder för sitt arbete kan ITIL-ramverket anses alltför tungarbetat för dagens agila utvecklingsprocesser.

Därför har DevOps tagits fram från utvecklarperspektivet i ett försök att överbrygga konflikten och skapa synergi i IT-organisationen mellan utvecklare och driften. Studien ämnar undersöka om implementationen av DevOps faktiskt har haft den effekt som avses, nämligen att föra dessa två roller närmare varandra.

Uppsatsens syfte är att utreda huruvida olika IT-organisationers tillämpning och uttryck av aspekter inom fenomenet ITSM triangeln påverkar den konflikt som existerar mellan utvecklare och drift. Detta för att utreda om fenomenets förekomst går att påvisa inom dessa grupper. En effekt av undersökningen blir också att belysa om skillnader i strategier kan antas ligga till grund för konflikten mellan utvecklings- och driftsorganisationer vid arbetet inom detta fenomen.

Forskningsfrågan syftar till att skapa en preskriptiv modell för hur arbetet med aspekterna inom ITSM triangeln kan struktureras för att identifiera brister inom organisationens arbete runt konflikten som beskrivits ovan. De underordnade frågorna som har drivit arbetet framåt är formulerade i punkterna under forskningsfrågan och den första frågan har till syfte att fungera som ingång till att förtydliga och definiera förutsättningarna för den efterföljande frågan.

- Hur bör modellen över fenomenet ITSM triangeln utformas för att främja IT-organisationens arbete med konflikten mellan utvecklare och drift?
  - Hur skiljer sig terminologin åt inom olika IT-organisationer när de kommunicerar om existerande aspekter inom ITSM Triangeln?
  - Hur bidrar eventuella skillnader i språkbruk och val av strategi till konflikten mellan utveckling och drift?

## **1.6 Målgrupp för arbetet**

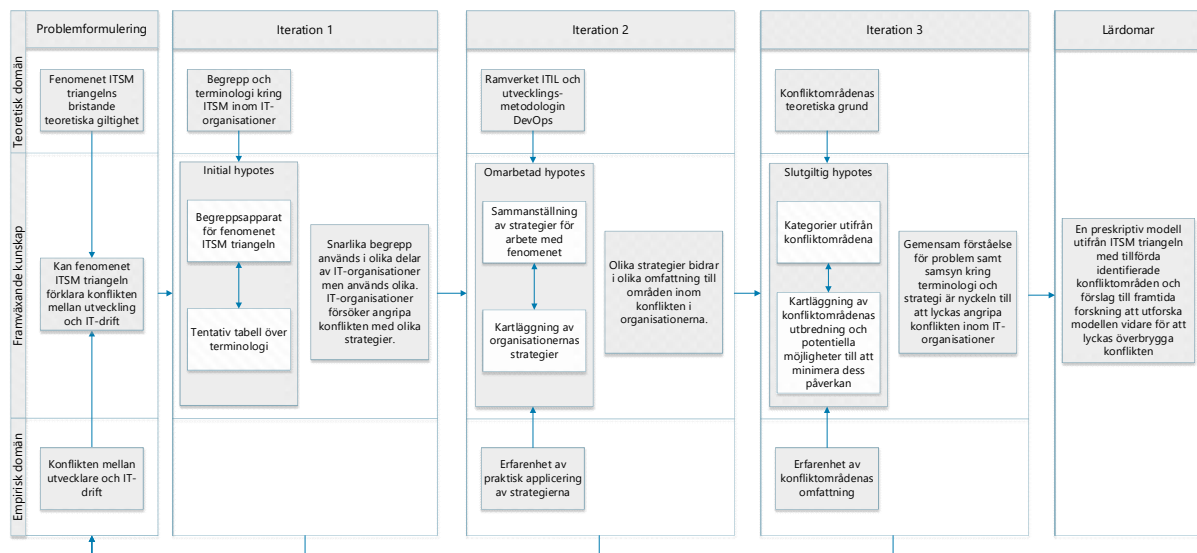
Målgruppen för denna uppsats är personer som är yrkesverksamma inom systemutveckling, IT-förvaltning eller innehar en ledande befattning inom IT-organisationer. Detta då syftet med uppsatsen är att tillhandahålla ett underlag för att påbörja arbetet med att överbrygga kommunikationsbrister mellan utveckling och drift. Även andra studerande inom IT-området, lärare och forskare med syftet att fördjupa kunskapen inom forskningsområdet kan hämta inspiration från denna studie. Övriga intressenter för studien är personer med intresse av områdena ITSM och IT i allmänhet, detta då konflikten inom området kan anses vara av allmänt intresse.

## 2 Metod

I detta kapitel lyfts vilka metodval och forskningsstrategiska beslut som tagits för att genomföra studien. Detta för att ge en tydlig bild av hur studien har genomförts för att efterföljande forskning skall få möjlighet att återskapa metoden med ett nytt urval för att validera resultatet och utreda generaliserbarheten i slutsatserna som framkommer i studien.

### 2.1 Forskningsstrategi

Uppsatsen har syftat till att undersöka ITSM Triangeln som fenomen, utan direkt koppling till ITIL, genom en explorativ små N-studie med en abduktiv ansats. Med en abduktiv ansats växlar studien mellan teoretiskt informationssökande och empirisk datainhämtning samtidigt som arbetet med de olika inriktningarna har genomförts med ett neutralt förhållningssätt sett till varandra. Den abduktiva ansatsen tillät också studien att förändra riktning när mer data framkom allteftersom arbetet fortlöpte (Cronholm, Göbel & Cao 2023), se Figur 2 för en visuell representation av arbetet. Enligt Jacobsen & Andersson (2017) skall en små N-studie användas för studier där ett begränsat antal respondenter tillfrågas från olika kontext om information rörande ett och samma fenomen. Detta passar uppsatsens frågeställning väl då den haft till syfte att utreda orsakerna till konflikten genom att använda ITSM triangeln som deduktiv analysmodell för att utreda förekomsten av konflikten inom olika kontexter och utreda hur olika organisationer uttrycker sig om och uppfattar fenomenets olika delar. Det beskrivs även hur man bör förhålla sig till tidigare forskning i sina studier och det framkommer att ett induktivt arbetssätt lämpar sig väl för insamling av empiriska data till forskningsfrågan då den data som efterfrågats i sin karaktär bör vara beskrivande (ibid.). Denna mix av induktiva och deduktiva arbetsmetoder har lett fram till att en abduktiv ansats har varit lämplig i arbetet med forskningsområdet då studien har utgått från ett etablerat fenomen som utifrån studiens perspektiv i stort är utforskat (Cronholm, Göbel & Cao 2023).



Figur 2: Den abduktiva processen modifierad efter Cronholm, Göbel och Cao (2023)

Empirin har inhämtats från olika verksamheter som på något sätt bedriver utveckling eller förvaltning av IT-system som en av sina verksamhetsprocesser. Då det i forskningsöversikten inte har gått att identifiera någon vetenskaplig källa som har undersökt detta fenomen ur det här perspektiv tidigare var det av intresse att genomföra en explorativ inriktad studie med syftet att undersöka huruvida fenomenet förekommer eller inte i dessa verksamheter. En explorativ inriktning på studien innebär att det förutsättningslöst undersökts huruvida studiens hypotes går att påvisa i den empiri som samlats in (Jacobsen & Andersson 2017).

## 2.2 Forskningsmetod

I arbetet med studien har teoretisk information samlats in genom en översiktlig strukturerad litteraturgenomgång. Detta innebär att information samlats in med hjälp av att ta fram relevanta ämnesord (Bilaga 1) för att söka vetenskapliga forskningsartiklar och konferenspapper ur databaser som tillhandahålls av Högskolan i Borås bibliotek. Databaserna som använts är följande: ACM digital library, DiVA Portal, EBSCO eBook Collection, IEEEExplore, JSTOR Journals, ProQuest, R Discovery, ScienceDirect, Taylor & Francis Online. En artikel tillhandahölls av Högskolan i Borås bibliotek då denna inte fanns att tillgå i någon online-databas tillgänglig för studenter. Det framsökta materialet har sedan använts för att finna fler relevanta ämnesord att söka vidare på. Alla de funna ämnesorden har använts i olika kombinationer för att söka fram artiklar och konferenspapper som sedan genom ett antal urvalskriterier, som listas i Tabell 2, har bedömts lämpliga och täckande för studiens ändamål. Värdering av materialet har sedan utförts genom en genomläsning av varje artikels titel, nyckelord, sammanfattning och slutsats (Säfssten & Gustavsson 2019). Den inhämtade materialet har sedan använts till att skapa en förståelse för forskningsområdet, ge en bild av problemet, ge en struktur till arbetet genom skapandet av kategorier och underlag för intervjufrågor samt analysera resultatet av den empiriska datainsamlingen.

Tabell 2: Studiens urvalssteg för den översiktliga strukturerade litteraturgenomgången.

Urvalssteg	Urvalsdel	Urvalsmetod
1	Nyckelord	Finns nyckelbegrepp för studien med i nyckelorden för artikeln.
2	Titel	Säger titeln något om huruvida artikeln innehåller ett för studien relevant ämne för att vidare läsa sammanfattningen.
3	Sammanfattning	Är sammanfattningen intressant och påvisar den ett innehåll som kan tänkas bidra till studien.
4	Slutsats	Har författarna till artikeln kommit fram till bärande kunskap som är relevant för studien.

Empiriska data har samlats in genom kvalitativa intervjuer och har genomförts enligt strukturen i kapitel 2.4 (*Datainsamling*). Dessa intervjuer genomfördes med ett semi-strukturerat tillvägagångssätt (Säfssten & Gustavsson 2019) och sammanställdes sedan genom en innehållsanalys av det transkriberade materialet. En blandning av konventionella och riktade innehållsanalyser har lämpat sig väl för studien då den konventionella innehållsanalysen gav möjligheter till att analysera empirin med en induktivt ansats och på så sätt bidra till nya insikter från den kvalitativa data som framkommit i studien detta trots att tidigare forskning inom ämnet är begränsad (ibid.). Samtidigt som en riktad innehållsanalys ger studien möjlighet att utgå ifrån begrepp, terminologi, aspekter och kategorier vilka identifieras med stöd i redan befintlig teori (ibid.). Då analysen i studien har haft en utgångspunkt i fenomenet ITSM tringeln och detta fenomen redan har sin terminologi definierad från ITIL, se kolumnrubrikerna i Tabell 1, blev terminologin underlag för de initiala kategorierna som används i analysdelen av studien. Dessa redan definierade termer används som kategorier i analysen. Det föll sig också naturligt att analysera det transkriberade materialet utifrån dess teoretiska begrepp (ibid.). Den abduktiva ansatsen i studien öppnade upp möjligheten för att komplettera de redan identifierade aspekterna med ny kunskap som ger studien ett större djup beträffande terminologin, kommunikationen, strategierna och konflikten som studerats i forskningsfrågan och i detta arbete har stegen som tagits i detta abduktiva arbetssätt dokumenterats i Figur 3.

## 2.3 Urval av respondenter

Respondenterna valdes ut genom ett ändamålsstyrt urval baserat på att efterhöra med verksamhetsaktiva inom IT-sektorn och andra kringliggande verksamheter för att få tag i personer som skulle passa in under urvalskriterierna som beskrivs nedan. Detta då Jacobsen och Andersson (2017) beskriver denna metod som bäst lämpad för kvalitativa studier när god insikt om vilken typ av respondenter som är av intresse för studien. Dessa personer har sedan kontaktats via mejl för att efterhöra eventuell lämplighet för att medverka i studien. När lämpligheten bedömts och respondenten visat sig vara intressant för studien har tid för intervju bokats in. Intervjuerna genomfördes sedan enligt strukturen i kapitel 2.4 (*Datainsamling*), på distans med hjälp av kommunikationsverktygen Zoom alternativt Teams för att på så sätt få tillgång till funktioner såsom inspelning av ljud och video samtidigt som respondenternas geografiska plats inte varit ett hinder för studiens genomförande.

Urvalsprocessen för respondenter baserades på tre kriterier som alla syftat till att precisera respondentens förutsättningar till att ha insikt i hur det undersökta fenomenet tar sig i uttryck och benämns inom verksamheten. Det första kriteriet var att respondenten skulle vara verksam inom en IT-organisation vars syfte är att antingen förvalta eller utveckla IT-system för den interna verksamheten eller externa intressenter. Andra kriteriet var att respondenten skulle inneha en roll inom verksamheten som överensstämmer med de generella rollerna Team Leader, Scrum Master eller förvaltningsledare då dessa roller ansågs inneha den nivå av verksamhetsinsikt som syftet med studien krävde. Tredje kriteriet var att personen skulle vara verksam inom Sverige och arbeta för eller ha arbetat för en organisation aktiv inom Sveriges gränser.

Med Team Leader avses en person som har till uppgift att leda och fördela arbetet i en grupp av individer vars arbetsuppgifter är att förvalta eller utveckla ett IT-system (Kristen 2022). Scrum Master avser en person som jobbar med metodarbete och metodstyrning i en grupp som utvecklar IT-system i en verksamhet som arbetar enligt arbetsmetodikens Scrum (Tonnquist 2024). En förvaltningsledare är en person som leder arbetet med underhåll och vidareutveckling av redan produktionssatt mjukvara i en organisation som har eller inte har egen systemutveckling (The SFIA Foundation u.å.).

## 2.4 Datainsamling

För studien genomfördes sex semi-strukturerade intervjuer med respondenter vilka föll inom ramen för studiens, ändamålsmässigt identifierade urval enligt processen i kapitel 2.3 (*Urval av respondenter*). Då arbetet genomförts abduktivt och områdeskunskapen ökat i takt med att nya empiriska data samlats in så har intervjuguiden uppdaterats vilket har lett fram till ökad kvalitet i frågeställningen för intervjuerna. Under studiens genomförande så har kompletterande frågor mejlats ut till respondenter som inte svarat på den tillagda frågeställningen. Datainsamlingen har syftat till att hämta in data kring terminologi och begrepp som kan härröras till fenomenet ITSM triangeln. Spridningen av kontexten för de olika respondenterna hade till syfte att uppnå ett bredare perspektiv av fenomenet än vad som framkommit i tidigare forskning. Syftet med att genomföra semi-strukturerade intervjuer har varit att låta respondenterna med egna ord och utefter egen erfarenhet återge en så, för respondenten verklighetstrogen bild av fenomenets företeelse som möjligt. Se Bilaga 2 för att se hur intervjuerna har strukturerats. Nedan i Tabell 3 följer en sammanställning över vilken litteratur som legat till grund för utformningen av respektive intervjufråga i intervjuguiden och i vilket syfte som intervjufrågan ställts i undersökningen. Frågorna IF1- IF5 har till syfte att besvara studiens forskningsfrågor medan IF6 har till syfte att bidra till nya insikter och höja studiens kvalitet avseende forskningsområdet.

Tabell 3: Tabell över studiens intervjufrågor, tillhörande referenslitteratur och syftet med intervjufrågan (IF1-IF6 refererar till intervjufrågans referensnyckel i Bilaga 2)

Fråga	Litteratur	Syfte
IF1	Orta och Ruiz (2019), Lubis, Annisyah och Winiyanti (2020) samt Walker (2022)	Att undersöka hur verksamheten uttrycker sig kring och uppfattar processen med att gå från upptäckt avvikande funktion i applikationen till att påbörja arbetet med att ta fram en lösning.
IF2	Lubis, Annisyah och Winiyanti (2020) samt Walker (2022)	Att undersöka hur verksamheten uttrycker sig kring och uppfattar processen med att gå från identifierad lösning till att lösningen är produktionsfatt i produktionsmiljön.
IF3	Ruiz, Moreno, Dorronsoro och Rodriguez (2018) samt Walker (2022)	Att undersöka om verksamheten har någon uppföljning kring att den förändring som gjorts i applikationen gett önskad effekt.
IF4	Jabbari et al. (2016), Walker (2022) samt Beetz och Harrer (2022)	Att undersöka om verksamheten särskiljer utvecklare och förvaltare i olika grupper eller om de har tvärfunktionella team.
IF5	Swanson och Beath (1989), Gregory, Henfridsson och Ruch (2018) samt McLean (2020)	Att undersöka om verksamheten upplever konflikt mellan utveckling och drift och hur de arbetar för att överbrygga denna.
IF6	Cronholm, Göbel och Cao (2023)	Möjliggöra för den abduktiva ansatsen, där det är önskvärt att ge varje intervjukandidat möjligheten till att bidra till utveckling av intervjuguiden om de anser att något saknas eller är felaktigt.

## 2.5 Dataanalys

Intervjuerna har genomförts digitalt via Zoom eller Teams och har därefter transkriberats med hjälp utav MS-Words inbyggda funktion för transkribering. Den automatiska transkriberingen kontrollerades sedan manuellt och det bearbetade och korrekta materialet analyserades med hjälp utav en blandning av en konventionell och en riktad innehållsanalys (Säfssten & Gustavsson 2019). I denna analys kategoriserades den data som framkommit för att fånga begrepp och strategier som var relevanta för forskningsfrågan och kunde härledas till sin motsvarighet inom ITIL. Dessa begrepp och strategier har brutits ned och jämförts med aspekter inom ITSM triangeln för att se om det a) finns en korrelation mellan empirin och teori eller b) om empirin kan tillföra ny kunskap till teorier och modeller. Utifrån denna analys har sedan slutsatser kunnat dras om huruvida det förekommer skillnader respektive samsyn kring aspekterna inom ITSM triangeln.

Studien inleddes med att ta del av forskning och teori nära relaterad till området ITSM för att bygga upp kunskap för att analysera studiens teoretiska domän. Till stöd för studiens arbete låg också författarnas tidigare erfarenheter av den konflikt som observerats på arbetsplatser i vilka utvecklare och drift kämpar om rätten till företräde för sin tolkning av hur arbete med systemapplikationer och produktionsmiljö ska hanteras. När den teoretiska domänen hade upprättats och möte de empiriska erfarenheterna gick studien in i första iterationen av arbetet som visas i Figur 2.

I den första iterationen upptäcktes fynd i form av att IT-organisationer använder till synes lika begrepp och terminologi, det identifieras dock att det finns mindre skillnader i hur dessa begrepp faktiskt används eller vad de åsyftar inom varje enskild verksamhet. Denna upptäckt

resulterade i att data sammanställts för att bringa klarhet i skillnaderna. I arbetet framåt upptäcktes även att verksamheter använder sig utav olika strategier i försök att överbrygga kommunikativa missförstånd som leder fram till att konflikten tillåts eskalera.

Den andra iterationen påbörjades med att inhämta mer teoretiska fakta om strategier som kan kopplas till arbetet inom ITSM triangels olika delar. Kunskapen om dessa strategier fördjupades och tillvägagångssätt som strategierna använder för att överbrygga konflikten kartlades. Genom att fortsätta inhämtningen av empiriska data och kontrollera hur olika verksamheter arbetar med att överbrygga konflikten identifierades olika områden som ligger till grund för konflikten inom dessa strategier, vilka återfanns som intressanta att utforska vidare i nästa iteration av arbetet.

I tredje iterationen av arbetet så fördjupades uppsatsen med teori och empiri gällande de områden som ligger till grund för konflikten som identifierats under studien. Nyckelbegrepp och strategier som skulle kunna vara av intresse för att lösa konflikterna som uppstår eftersöktes hos respondenterna och i vetenskapliga texter. Detta bidrog till insikter i hur organisationer hanterar motsättningar inom det studerade fenomenet och vilket stöd som finns i teorin för att dessa tillvägagångssätt är vetenskapligt förankrade.

## **2.6 Etiskt ställningstagande**

Studien har i alla avseenden försökt att förhålla sig neutral till den konflikt som studerats och har därigenom formulerat frågorna i intervjuguiden utan att betona konflikten med avsikt att inte förnära eller kritisera någon part i intervjun. Det finns en viss risk att någon respondent i detta kan känna sig kritiserad till den analys som gjort av deras svar. Detta har dock inte varit avsikten med det neutrala förhållningssättet utan har varit en direkt konsekvens av att forskarna inte velat färga svaren i intervjusituationen. Detta etiska dilemma tas upp av Jacobsen och Andersson (2017) som menar att det kan vara nödvändigt att ibland inte vara helt öppen med varje aspekt av undersökningens syfte då detta kan påverka respondentens utsago under intervjun.

Ett ytterligare etiskt beslut har tagits om att i möjligaste mån låta respondenterna vara anonyma i sin utsago. Detta har bidragit till att studien har skalat bort all fakta om kontexten från var respondenten härrör och vilka de är då detta har ansetts ge ringa bidrag till det framtagna resultatet i studien. I ett försök att göra det omöjligt för läsaren att utrona respondenternas identiteter och vad de står för. Det blir viktigt i studien då respondenterna är så få till antalet (Jacobsen & Andersson 2017).

## **2.7 Metodreflektion**

I enlighet med gällande forskning kring metod och vetenskaplig analys har avsikten varit att göra lämpliga val av metoder för studien i syftet att belysa forskningsfrågan på ett adekvat och sanningsenligt sätt. Både Jacobsen och Andersson (2017) samt Säfsten och Gustavsson (2019) har använts som kurslitteratur i den metodkurs som författarna till denna uppsats genomgått innan uppsatsskrivandet. Det blev då naturligt att använda dessa två källor för att styrka den metod som använts i genomförandet av studien. Det har löpande diskuterats under studiens genomförande huruvida metoderna som valts lyckats i att leverera bidrag som tillför värde för de resultat som framkommit under studien. I dessa diskussioner framkom att studien bör genomföras abduktivt då detta arbetssätt tillåter en flexibel ansats mellan teori och empiri som förklaras av Cronholm, Göbel & Cao (2023) i deras konferensrapport "Abductive Design Science Research: The Interplay between Deduction and Induction". Denna källa gav också en djupare insikt i hur denna ansats skall implementeras i studien. Valet att genomföra en små-n-studie grundade sig i intresset för att utreda ett fenomen utanför den kontext som fenomenet

idag är etablerat inom. Genom att utföra en små-n-studie gavs möjligheten att se på fenomenet ur flera perspektiv, i olika kontext. Ett annat alternativ hade varit att genomföra en fallstudie av fenomenet, då hade resultatet blivit mer vinklat och saknat det bredare perspektivet som eftersökts för att på ett adekvat sätt utreda fenomenets förekomst samt den upplevda konflikten mellan systemutvecklare och drift som existerar inom IT-organisationer idag.

Det har även blivit uppenbart att valet att inte använda termer från ITIL tillsammans med det neutrala förhållningssättet på vilken konflikten presenterats i intervjuerna har gett det empiriska underlaget till studien en validitet som går att spåra i resultatet. Genom att aktivt välja att genomföra intervjuerna på detta sätt har respondenterna inte reflekterat över att deras syn på processen inte är den enda utan helt från egen erfarenhet och egna värderingar gett de ofärgade svar som eftersökts. Om intervjuguiden i stället hade utformats med en terminologi hämtad ur ITIL hade respondenterna säkerligen färgats i sina svar då nästan alla respondenter kände till ramverket och var familjära med dess terminologi. Detsamma gäller förhållningssättet till konflikten som hade kunnat visa sig i svaren om respondenten hade känt sig ifrågasatt eller stärkt av att frågan ställts med en positiv eller negativ vinkling till dennes uppfattning om ämnet.

Vid analysen som genomförts av det insamlade materialet har valet fallit på en blandning av en konventionell och en riktad innehållsanalys, detta på grund av att frågeställningen har varit av en utforskande karaktär vilket föranleder behov av att öppet kunna finna nya begrepp och termer för det redan etablerade fenomenets aspekter. Samtidigt har innehållet i svaren analyserats för att ta fram underliggande faktorer till konflikten som kunnat spåras i det empiriska materialet som framkommit. Valet att genomföra analysen genom en blandning av riktad respektive konventionell innehållsanalys grundades sig i en viss oro kring hur resultatet riskerat att i viss utsträckning ta skada av att vara bundet till utgångspunkt i exempelvis ett visst ramverk. Vår bedömning är att analysen hade försvårats om vi helt tagit avstamp i en redan befintlig teoretisk grund eftersom studiens syfte är att finna och studera nya ingångsvinklar och perspektiv på området. Att angripa och analysera materialet med ett helt öppet sinne torde därför vara det mest lämpliga tillvägagångssättet.

Oro fanns för att författarnas ringa erfarenhet av att genomföra intervjuer och innehållsanalyser skulle komma att påverka resultatet som framkom i det material som ligger till grund för analysen. Denna problematik har aktivt arbetats med tillsammans med handledaren för att minimera dessa effekter. En annan aspekt som tagits i beaktning under arbetet är den verksamheterfarenhet som en av författarna besitter av ITSM-processerna och ramverket ITIL. För att motverka att detta haft påverkan i resultat- och analysdelen så har en kontinuerlig dialog mellan författarna använts för att konfirmera att det som skrivits i arbetet faktiskt framkommit under studien och inte är taget direkt ur den verksamheterfarenhet som besitts av författarna.

## 3 Teori

I detta kapitel ges en översyn av teorin som använts under studien och motiveringar till valen av kategorier som styrkts av litteraturen och som sedan använts i analysen av det empiriska materialet.

### 3.1 Terminologi

För att få en bild över vad för terminologi som kan väntas uppkomma i empirin har terminologin i artiklarna av Eaddy, et al. (2008), McLean (2012), Waller, Ehmke och Hasselbring (2015) samt Wang och Cao (2021) analyserats och listats i Tabell 1. Vidare har begreppen som förekommer i studien hämtats ur olika källor och översatts till svenska. Nedan förklaras begreppen som återkommande hittats i teorin.

Felhantering är ett samlingsbegrepp för insamling av fel-rapporter och arbetet med problemlösning som vanligen omnämns i forskningslitteratur. Att utföra aktiviteter inom felhanteringsprocess benämns bland annat som "perform the resolution tasks" (Orta & Ruiz 2019, s.9), "Bug-fixing" (Wang & Cao 2021, s.630) och incident- samt problemmanagement (Axelos 2019; Agutter 2020).

Termen åtgärdsrealisering är en översättning av flertalet engelska termer som återfinns i teorin. Termer som förekommer i litteraturen är till exempel "issue resolution" (Wang & Cao 2021, s.637), "concern's implementation" (Eaddy et al. 2008, s.513), change management (Ruiz, Moreno, Dorronsoro & Rodriguez 2018) och change enablement (Axelos 2019; Agutter 2020).

Inom ITIL används många områdesspecifika termer som var och en förklarar någon aspekt av det arbete som utförs inom ITSM triangeln. Till exempel nämner ITIL-ramverket att det är önskvärt att använda en "single point of contact" (Agutter 2020, Kap: Service Management Practices, Stycke: The Service Desk) när det kommer till incidentrapportering. Denna enda kontaktväg utgörs oftast av en servicedesk (Axelos 2019; Orta & Ruiz 2019; Agutter 2020). Andra benämningar för organisatoriska enheter som fyller denna funktion är kundtjänst (Orta & Ruiz 2018).

Root cause eller den svenska översättningen rotorsak är ett till begrepp som används inom problem management och beskriver den underliggande orsaken till att ett problem faktiskt uppstått i systemet (Axelos 2019; Agutter 2020; Lubis, Annisyah & Winiyanti 2020; Okes 2019).

DevOps tar upp termer såsom "Communication, collaboration, team working" (Jabbari et al. 2016, s.5), "Bridge the gap" (Ibid.), "Development method" (Ibid.), "Software delivery" (Ibid.), "Automated deployment" (Ibid.), "Quality assurance" (Ibid.), continuous integration (CI) och continuous deployment (CD) (Beetz & Harrer 2021) för att prata om aspekter inom felhantering och åtgärdsrealisering.

### 3.2 Strategi

I detta stycke presenteras ingående två erkända ramverk för att arbeta med felhantering och åtgärdsrealisering inom IT-organisationer. ITIL är den best practice som åtföljs av de flesta driftsorganisationerna (Agutter 2020) medan DevOps är ett ramverk som används i mer utvecklingsorienterade organisationer för att främja samverkan mellan utveckling och drift samt ta höjd för kraven inom agila arbetssätt (Wiedemann, Wiesche, Gewalt & Krcmar 2020).

### 3.2.1 ITIL

Enligt Orta och Ruiz (2019) hanteras incident management implementerat enligt ITIL-ramverket vanligen med en process som servicedesk, vilken är en del av en organisations förvaltning dit användare ges möjlighet att ta kontakt för att rapportera in att incidenter uppstår. Vid inrapportering kategoriseras incidenten beroende av typ, klassificeras med hänseende till allvarlighetsgrad och tilldelas prioritering utifrån vilken påverkan som uppstår för verksamheten. När en incident hanterats av servicedesken kan uppgiften att hantera incidenten fortskrida enligt organisationens policy avseende incidenthantering, vanligen finns det flera olika nivåer inom en organisation som hanterar incidenter. Ärendets väg eskaleras beroende på karaktär och prioriteringsgrad. När väl en åtgärd finns tillgänglig för att lösa användarens bekymmer återkopplar agenten som hanterat incidenten och bekräftar att åtgärden löst incidenten och ärendet kan stängas (ibid.).

Vid de tillfällen en incident inte kan avhjälpas omgående tillämpas processområdet problem management som avser en identifieringsprocess med syfte att helt förhindra skadan av incidenter som är återkommande, alternativt att minimera påverkan av incidenter som i nuläget inte är möjliga att helt och hållet lösa. Arbetet med problem management innefattar även att utreda orsaken till att incidenter uppstår, samt uppföljning av problemet för att finna och genom utveckling av systemet bygga bort rotorsaken till problemet (Lubis, Annisyah & Winiyanti 2020; Okes 2019).

Vidare omfattar change enablement (tidigare benämnt change management i ITIL 3) processer och aktiviteter som täcker in förändringsarbetet och konfiguration av IT-system och IT-tjänster. Vid arbete med change är en viktig aspekt som lyfts fram vikten av att utvärdera hur väl utfallet av en change föll ut. Det framgår tydligt i ITIL:s change management process att först tas en change fram, sedan genomförs den och därefter behöver den utvärderas före change-ärendet slutligen kan stängas. (Ruiz, Moreno, Dorronsoro & Rodriguez 2018)

### 3.2.2 DevOps

Att arbeta enligt DevOps syftar till att underlätta samförstånd mellan utvecklare och drift samt möjliggöra kontinuerliga släpp av systemförändringar till produktionsmiljö. Det ställs därför större krav på kommunikation, tvärfunktionella samarbeten och arbetssätt för att uppnå DevOps syfte jämfört med traditionellt systemutvecklingsarbete inom IT-organisationer. Idén som återspeglas i DevOps består i att flera olika utvecklingsteam kontinuerligt släpper nytt material (CD), uppdateringar eller fixar så nära direkt mot produktionsmiljön som möjligt (CI). Detta benämns som CI/CD av Wright och Druta (2014). Humble och Farley (2010) menar på att DevOps borde utgöra en naturlig del av systemutvecklingsarbete vilket de uttrycker på följande sätt:

*”It is worth emphasizing that the first release of an application is just the first stage in its life. All applications evolve, and more releases will follow. It is important that your delivery process also evolves with it.” (Humble & Farley 2010, Kap.: The Problem of Delivering Software Stycke: Continuous Improvement)*

Inom DevOps läggs betydande vikt vid att team ska samlas för att gemensamt utvärdera arbetsinsatsen som utförts. Denna ceremoni kallas för retrospective och har till syfte att så väl utveckla processen som att ta med lärdom in i det framtida arbetet som att faktiskt utvärdera den aktuella arbetsinsatsen. Vanligen uppmuntras det till att personer från hela organisationen

som ingått i arbetsprocessen deltar under retrospective ceremonin, till syfte att undvika att information och kunskap blir fast på ett ställe inom organisationen, det handlar om att sprida kunskapen. (Humble & Farley 2010)

Det finns ett antal principer som kan sägas vara tillämpbara och gällande inom DevOps. Dessa principer bygger på en tanke om att underlätta att snabbare ge och ta emot feedback och att reducera risker inom utvecklingsarbete. Flera av principerna återfinns inom den agila arbetsmetodiken. Principerna innefattar att arbeta iterativt, vilket är en kärnprincip inom agilt arbete och DevOps. Att arbeta inkrementellt, som blir en direkt konsekvens av att arbeta i iterationer och tanken är att hela tiden utveckla arbetet till det bättre med varje cykel. Att arbeta med kontinuitet, vilket i sammanhanget syftar på att arbetet utförs likvärdigt genom processen, att följa rutiner och överenskomna arbetsmetoder. Att automatisera arbetsuppgifter som är repetitiva och återkommande där så är möjligt för att undvika fel beroende av mänskliga faktorn. Att vara självsvandande vilket innebär att DevOps-team ska ges möjlighet att agera självständigt och på så sätt minska tidsåtgång, ett team kan själva sätta upp en testmiljö i stället för att invänta att ett annat team sätter upp testmiljö åt dem till exempel. Att samarbeta, vilket bland annat omfattar vikten av att förbättra kommunikation och fokusera på personlig kontakt mellan teamen och slutligen holism, att team ska vara organiserade på ett sådant sätt att det möjliggör självständig utveckling, testning och tillgängliggörande av värde för kunden. Sammantaget kan DevOps sägas bygga på dessa sju grundläggande principer dock är det inte nödvändigt att alltid följa samtliga principer eftersom DevOps praktiseras genom metoder. Det bör dock tilläggas att genom att inte vara tydligt definierat i sin utformning, så är DevOps principerna endast att se som rekommendationer och inte ett regelverk för hur metodologin utformas. (Beetz & Harrer 2022)

### **3.3 Kommunikation**

Humble och Farley (2010) betonar vikten av kommunikation i form av kunskapsspridning. De belyser vikten av kommunikation genom hela processen då de talar om att det är nödvändigt att ha en förståelse för att applikationer först produktionssetts och sedan kommer att utvecklas under hela sin livscykel. Detta gör kunskapsspridning till en kritisk framgångsfaktor för att lyckas med felhantering och åtgärdsrealisering i ett IT-system (Humble & Farley 2010).

Inom DevOps återfinns en feedbackloop som benämns ”communication and collaboration” i definitionen med detta menas att det skall finnas en kontinuerlig dialog mellan utveckling och drift under hela livscykeln (Jabbari et al. 2016). Denna feedback är kritisk för att minska risken för och upptäckten av missförstånd och felaktiga antaganden i systemutveckling (Ravichandran, Taylor & Waterhouse 2016).

### **3.4 Konflikt**

Artiklar från författarna Swanson & Beath (1989), McLean (2012) Ravichandran, Taylor & Waterhouse (2016), Gregory, Henfridsson och Ruch (2018) samt Walker (2022) har studerats i syfte att skapa en tydligare bild av, samt beskriva den konflikt som återfinns mellan systemutvecklare och IT-förvaltare. Konflikten kan antas vara ett naturligt fenomen då parterna ansvarar för samma område men har olika ingångsvinklar på hur utveckling och drift inom området bör hanteras.

Man har länge kunnat se klara tendenser på att arbetet med drift har ansetts vara en mindre prestigefylld uppgift. Detta har lett till att personer som arbetar med drifts uppgifter har uppfattats som mindre viktiga än utvecklarna. Detta i sin tur har ökat spänningarna mellan grupperna (Swanson & Beath 1989).

Med ökade krav på agilt arbete och snabbfotad utveckling av ny funktionalitet pressas utvecklare att producera nya funktioner och förbättra prestandan på IT-system i allt högre takt (Ravichandran, Taylor & Waterhouse, 2016). Detta medan drift har en lång tradition av att försöka minimera förändringar i systemet för att kunna garantera en smidig produktionsmiljö med så få avbrott som möjligt (Gregory, Henfridsson & Ruch 2018). Detta ger upphov till en konflikt mellan utvecklare och drift då målet för de olika grupperna skiljer sig åt i så stor utsträckning (Wiedemann et al. 2020).

Okänd förändring av IT-system är en av orsakerna till att konflikt uppstår inom IT-organisationer, detta är ett fenomen som vanligen visar sig vid arbete med molnbaserade applikationer. I moderna och välutrustade IT-organisationer finns det idag analysverktyg som används för att eftersöka förändringar och anomalier i IT-system, dock tar dessa verktyg ofta relativt mycket tid i anspråk innan en färdig och tillförlitlig analys finns att tillgå. Det vill säga, att innan det finns fakta som kan härleda att en förändring faktiskt lett fram till att en incident eller ett problem uppstått i systemet, samt kopplingen mellan förändringen och den upplevda kapacitetsförlusten. (Ravichandran, Taylor & Waterhouse, 2016)

## 4 Resultat

Kapitlet presenterar det resultat som framkommit vid datainsamlingen under studiens genomförande. Strukturen på kapitlet följer ITSM triangelns aspekter och de kategorier som enligt studien ligger till grund för konflikten mellan utveckling och drift. Denna uppdelning skapar en god översikt över vad som hämtats ut ur intervjuerna.

### 4.1 Incident

Som nämnts i avsnitt 3.2.1 (*ITIL*) beskrivs incident management som den process inom ITIL vilken är det initierande stadiet i felhanteringsprocessen hos en IT-organisation. Processen omfattar bland annat inrapportering, kategorisering av typ, samt klassificering utifrån felets allvarlighetsgrad (Agutter 2020; Orta & Ruiz 2019). Nedan sammanställs resultatet av den empiriska undersökningen som genomförts under studien.

#### 4.1.1 Terminologi

När det kommer till den aspekt som påbörjar arbetet med felhantering så benämns det bland annat som felrapportering. Andra begrepp som används är ärenden från slutanvändaren samt defekt i systemet. En respondent pratar om felhantering och använder då begreppet problem i kontexten av att en användare som upptäckt ett fel rapporterar in det som ett problem.

Respondenterna har olika termer som de använder för aspekten incident under intervjuerna och en sammanställning av terminologin har gjorts i Tabell 4. Detta förtydligar och bidrar till insikten om att det saknas en gemensam terminologi för att uttrycka denna typ av företeelse inom felhanteringsprocessen mellan olika IT-organisationer.

#### 4.1.2 Strategi

Samtliga respondenter är enig om att denna process initieras från en ensam källa inom verksamheten där felrapporter tas emot av någon form av kundtjänst eller supportfunktion. Respondenterna uppvisar även en gemensam bild över att det inom felhanteringsens initialskede skapas upp en artefakt utifrån den initierande rapporten. Denna artefakt kategoriseras och prioriteras utefter allvarlighetsgrad och ligger sedan till grund för eller uppdaterar en registrerad bugg i utvecklingsteamens backlogg. Detta kan dock i vissa fall undvikas om driften har möjligheten att med enklare medel konfigurera eller ändra i systemet på något sätt för att åtgärda det rapporterade felet.

#### 4.1.3 Kommunikation

Det talas om en problematik vilken upplevs av utvecklingsteamerna i att de inrapporterade incidenterna påvisar en bristande kunskap om IT-systemen och därav inte är formulerade så att utvecklarna förstår vad som faktiskt behöver åtgärdas. Vanligen rapporteras incidenten i flera steg för att komma utvecklaren till handa.

Vid inträffandet av allvarliga incidenter inom en av verksamheterna efterfrågas generellt feedback på om huruvida incidenterna slutat uppkomma eller inte längre går att återskapa i produktionsmiljön efter det att lösningen är släppt till produktionsmiljön. Detta går också i linje med vad en annan respondent uppger om att feedback alltid tas upp från slutanvändaren antingen innan eller när en lösning har körts ut i produktion. Vidare påtalas dock att det vid mindre buggrättningar inte efterfrågas någon sådan feedback då man i stället förlitar sig på att testerna som gjorts innan lösningen släpps till produktionsmiljön är tillförlitliga nog.

#### 4.1.4 Konflikt

Den mest noterbara skillnaden mellan respondenterna är huruvida de är intresserade av slutanvändarens feedback beträffande förändringens resultat i att åtgärda den incident som påbörjade felhanteringsarbetet. Några respondenter säger sig vara intresserad av samtliga förändringars påverkan på ursprungsincidenten medan andra inte nämner detta alls.

## 4.2 Problem

Som nämnts i avsnitt 3.2.1 (*ITIL*) beskrivs problem management som en identifieringsprocess med syfte att klargöra felets omfattning, utreda felorsaken och arbeta fram en hållbar lösning (Lubis, Annisyah & Winiyanti 2020). Nedan sammanställs resultatet av den empiriska undersökningen som genomförts under studien.

### 4.2.1 Terminologi

En intressant iakttagelse kring arbetet med att utreda och åtgärda problem är att det endast är en utav respondenterna som säger att de inom verksamheten arbetar med termer som nuläge kontra förväntat läge, samt lägger stor vikt vid att det finns en tydlig samsyn kring dessa begrepp inom IT-organisationen. Övriga respondenter använder sig av betydligt vagare formuleringar eller omnämner överhuvudtaget inte den här typen av konsensusbild i sammanhanget. De flesta respondenter kan ändå anses vara inne på området lägesbild, genom att de talar om en kontinuerlig dialog, alltså att det kontinuerligt hålls en aktiv dialog mellan utvecklare och driftsteam. Terminologin över hur man uttrycker sig om lägesbilden verkar i dessa fall variera mellan verksamheterna. Vissa använder sig av en strategisk definition av IT-systemet som ägs av produktägaren andra går på utvecklarens tolkning av vad de identifierar som fel i IT-systemet.

När ett problem har prioriterats som kritisk benämns dessa problem som hotfixes alternativt kritiska problem. Det finns en stor samstämmighet över hur man uttrycker sig om aspekten när en incident rapporterats och arbetet med problembilden tar vid så kallas denna aspekt för en bugg.

När arbetet med att finna lösningen är på plats går återigen terminologin isär. Begrepp som används för att gå över från problem- till changeprocessen är: pull request, rätta, sprint review eller stagea.

### 4.2.2 Strategi

Felhanteringsprocessen visar sig fortsätta från registrering av felet till bland annat utredning av felorsak, det vill säga att utreda hur felet uppstod och varför, samt vikten av prioritering av felen för att separera de kritiska felen från de enklare. När ett fel uppdagats behöver den som utreder felet titta närmre på orsaken till hur problemet kunde uppstå och registrera det som en bugg. Utifrån klassificering och bedömd allvarlighetsgrad hos en bugg fortlöper arbetet därefter vidare enligt buggens prioritering och omfattning. Samtliga respondenter ser buggrättning som en förändring som behöver göras i programkoden för att åtgärda eventuella fel. Dock ser en respondent att buggrättningar kan ta sig i uttryck även som omläsning av filer eller direkt åtgärd på transaktionsdata. Vanligen rör det sig om en bugg som kan åtgärdas direkt av first- eller secondline operatörer vilket gör att buggrättningen således inte behöver processas i det ordinarie processflödet utan går vid sidan av. Bedöms buggen vara så pass allvarlig att den omgående behöver åtgärdas till följd av att den påverkar och stör produktionsmiljön, tillskrivs den en hög prioritet eftersom denna bugg behöver åtgärdas genom en så kallad hotfix. Lite överraskande ges svaret från en av respondenterna att under normala förhållanden så lyckas man åtgärda kritiska problem inom loppet av ett dygn, vilket står i kontrast mot övriga

respondenter som uppger strategier i stil med att släppa allt man har för att nå en åtgärd inom detta korta tidsspänn. Visar det sig att buggen inte är av sådan karaktär att den direkt stör produktionsmiljön kan åtgärden i stället planeras i en åtgärdsplan som läggs till i backloggen och arbetas vidare med därifrån. Det ansvariga teamet tar sig an och hanterar buggen, det vill säga tar fram ett förslag på relevant lösning som sedan testkörs. Det team som tar fram lösningen skapar och utför även tester av lösningen och lyfter in lösningen i en produktionslik miljö i denna fas av processen.

I processen att utreda buggar och initiera arbetet med att ta fram en lösning ställs frågor som; Vad är det som ska åtgärdas? För vem skapar vi lösningen? Detta är en dialog som förs med respektive produktägare som då görs delaktiga i processen alternativt förs dialogen mellan utvecklaren och den som rapporterat felet. Produktägaren har i denna del av processen som mål att se till att lösningen inte skapas utifrån en enskild användares önskemål eller behov, utan säkerställer att det är strategiskt korrekt att genomföra förändringen. Detta kan ses som en variant av strategi där det i slutet av felhanteringsprocessen utvärderas huruvida verksamheten och IT-organisationen har samma förväntade bild av hur funktionen som buggrapporterats skall fungera. Finns det inte en gemensam bild blir nästa steg i processen att identifiera ett gemensamt förväntat läge för funktionen. Detta för att skapa en samsyn kring önskat läge respektive nuläge för att klargöra vilka åtgärder som behöver tas för att återfå systemet till det önskade läget. Lösningen arbetas sedan fram i en kontinuerlig dialog mellan slutanvändaren och utvecklare tills dess att en fullgod lösning är utvecklad.

I problemutredningen uppges att orsaken till att buggen uppstår kan visa sig i att inte ligga i förvaltningens händer utan i en systemdel som inte ägs eller kan konfigureras av den egna verksamheten. Ärendet behöver därav hanteras av en extern tjänsteleverantör. Detta arbete benämner en av respondenterna samlas i en Root Cause Action (RCA), detta arbete innefattar att kontakta berörd extern part och meddela att problemet ligger på deras sida och måste hanteras därifrån. Medan en annan respondent menar på att när problem kan härledas till tredje part påverkar detta hanteringen av utredningen väldigt individuellt utifrån vilken aktören i fråga är. Utredningsprocessen påverkas av gällande service level agreement (SLA) det vill säga gemensamma förväntningar parterna emellan och bland annat hur kritisk tjänsten som tillhandahålls är för respektive part.

Enligt agilt arbetssätt, som samtliga respondenter arbetar med, gäller att upptäckta och registrerade buggar, oavsett om de föranleds av ett ärende eller om buggen uppmärksammats via logggranskning, tas till backloggen och det är upp till den ansvarige för arbetet att planera och prioritera vad som behöver åtgärdas och i vilken ordning. Vid agilt arbete förespråkas att släpp av rättningar eller buggfixar bör göras löpande så länge det inte rör sig om några större produktionsstörande systemdelar som är drabbade. Skulle så vara fallet kan det krävas ett mer planerat och kontrollerat släpp mot produktionsmiljön. Beträffande åtgärdsrealisering uppges att teamen arbetar tvärfunktionellt både inom teamen, då det är teamet som utvecklar samt äger sin produkt och därigenom också hanterar driften utav den samma, men även externt med andra team för att identifiera vilket team som äger rotorsaken.

Efter testkörning kan åtgärder eller lösningsförslag läggas upp för att hanteras vid nästa planerade släpp till produktion. Enda gången detta förbigås är vid en hotfix som innebär att en åtgärd måste tas fram och släppas omgående för att det inte finns tid att vänta på ett schemalagt släpp. Alternativt så har teamet utvecklat automatiserade tester som kvalitetssäkrar buggrättningarna för att sedan släppa rättningarna direkt till produktion. I vissa fall kan utvecklarna ta beslut om att testa själva i testmiljön för att sedan köra ut lösningen i produktion

men dessa fall är av mindre allvarlighetsgrad och skall vara enkla att verifiera så att lösningen är korrekt av utvecklaren personligen.

### **4.2.3 Kommunikation**

Vissa respondenter nämner att man i arbetet med buggen har en kontinuerlig dialog med den som registrerat ärendet för att kontrollera om åtgärden faktiskt löser felet. Svar på åtgärds effekt kommer då snabbt åter till den part som skapat åtgärden. Detta innan buggrättningen slutligen förbereds för ett acceptanstest. Det framgår att en god förståelse för verksamheten som rapporterat in felet hjälper driftsorganisationen att bättre kommunicera vad felet är och till vilken grad det påverkar verksamheten.

### **4.2.4 Konflikt**

En utmaning som har uppmärksammat är att det förekommer kunskapsbrister om hur andra gruppers lösningar och ansvarsområden fungerar. Detta ger upphov till brister i förståelsen och kommunikation mellan teamen. Detta är en identifierad problematik som vissa verksamheter arbetar aktivt med. Bland annat genom att flytta personal mellan grupperingarna för att på så sätt överbrygga kommunikationsbristerna samt sprida kunskap om helheten till respektive team. Andra tendenser finns till att utvecklare inom vissa organisationer faktiskt kan riskera att elda på konflikten inom IT-organisationen genom användandet av så kallade "silent releases". Dessa silent releases innebär i den här kontexten att utvecklare själva anser sig kunna verifiera och acceptera åtgärder för små buggar av enklare karaktär och sedan låta dessa gå ut i produktionsmiljön utan vidare utvärdering som till exempel acceptanstester. Vid det här förfarandet har ett aktivt val gjorts om att gå förbi feedbackloopen som inom agilt arbete anses som mycket viktig.

Ytterligare en faktor som kan anses bidra till konflikten som lyfts fram är att utvecklingsteamet i många fall har bristfällig verksamhetsförståelse och i utredningen av buggen därför riskerar att missa väsentliga delar av problembilden för att rättningen skall anses fullgod och lösa användarens problem fullt ut. Det lyfts av en enstaka respondent fram att ett ytterligare område för konflikt är prioritering av vilka problem som skall åtgärdas först respektive vad som kan väntas med. Då utvecklarna behöver ta hänsyn till verksamhetens drift som helhet vid prioritering av vilka buggar som skall arbetas med i nuläget. Vidare uppges av en respondent att i ett försök att förekomma konfliktmomentet med bristfällig förståelse för varandras arbetsområden arbetas och planeras verksamhetens arbete i längre cykler där hela verksamheten gemensamt deltar i planeringsforumet.

## **4.3 Change**

Som nämnts i avsnitt 3.2.1 (*ITIL*) beskrivs change enablement som den process som omfattar förändrings- och konfigurationsarbete i IT-system (Ruiz et al. 2018). Nedan sammanställs resultatet av den empiriska undersökningen som genomförts under studien.

### **4.3.1 Terminologi**

Det finns fler begrepp som används när arbetet med en bugg är färdigställt och lösningen är redo för acceptanstest och produktionssättning, begreppen som framkommit under intervjuerna är åtgärda, släppa, release, köra ut och stagea. En viss antydning till samstämmighet råder om slutfasen av change processen då ett par respondenter benämner denna aspekt som släpp.

### **4.3.2 Strategi**

I flera verksamheter utförs acceptanstesterna av övriga IT-kollegor alternativt IT-team vilket innebär att de granskar förslaget för slutligt godkännande innan ändringarna kan släppas till

produktionsmiljön. En respondent anser i stället att det bästa sättet att hantera denna aspekt av fenomenet är genom automatiserade tester och att efter de är genomförda och godkända så släpps lösningen direkt. Ett alternativ till automatiserade tester är att verksamheten har ett team av "trackers" som försöker testa den kod som skall släppas för att hitta eventuella buggar som då skickas tillbaka till utvecklingsteamet för att arbetas vidare med.

I en alternativ process till de ovan nämnda återfinns arbetet med att acceptanstesta lösningarna med hjälp av slutanvändarna alternativt den som rapporterat felet. Detta så att den utvecklade lösningen på felet är redo att köras ut i produktion samt att den faktiskt löser användarens ursprungs fel.

#### **4.3.3 Kommunikation**

En respondent uppger att man i samband med genomförandet av en change skickar ut ett dokument som benämns "Release notes". Dessa dokument innehåller information om vilka förändringar som utförts och syftar till att ge slutanvändaren möjlighet att testa och ge feedback på den genomförda changen. Andra uppger att de vid behov samt vid inträffandet av större fel upprättar en "After action plan" för att komma fram till vad som faktiskt hänt, vad som gjorts och hur organisationen skall hantera situationen så att felet inte uppstår igen. Ett alternativt tillvägagångssätt inom vissa verksamheter är att de i stället alltid efterfrågar feedback om utfallet av produktionsatta lösningar.

#### **4.3.4 Konflikt**

Det framgår i flera verksamheter att det saknas en tydlig feedbackloop tillbaka till incidentprocessen vilket resulterar i att en konflikt uppstår då det inte blir tydligt för varken den som felrapporterat eller den som utvecklat lösning om huruvida felet faktiskt avhjälpas med den produktionsatta åtgärden eller inte.

## 5 Analys & Diskussion

Kapitlet presenterar en analys av resultatet som framkommit vid datainsamlingen under studiens genomförande och resonerar kring den teoretiska och empiriska validiteten för studiens hypotes. Uppdelningen av informationen är gjord utefter de kategorier som enligt studien ligger till grund för konflikten mellan utveckling och drift.

### 5.1 Terminologi

Utifrån resultatet som presenterats ovan sammanfattas här de benämningar som har koppling till ITSM triangeln för att ge en klar bild över hur respondenternas utsago stämmer in på fenomenets terminologi. Respondenternas individuella termer för varje del av ITSM triangeln kan utläsas i Tabell 4.

Tabell 4: Olika respondenters benämningar på ITSM triangelns delar

ITIL termer	Incident	Problem	Change
Respondent F	Defekt	Bugg	Staging
Respondent K	Fel	Problem	Implementering
Respondent L	Ärenden	Bugg	Släpp
Respondent M	Fel	Ärende	Åtgärd
Respondent R	Fel	Bugg	Släpp
Respondent S	Problem	Bugg	Release

När det kommer till begreppen som används inom aspekten incident kan det visa sig problematiskt att använda begreppet problem. Detta påstående grundar sig i att inom ITIL används begreppet incident i incident management där hanteringen av fel initieras medan begreppet problem i stället återfinns inom problem management och där avser begreppet problem en eller flera sammankopplade incidenter som hanteras och utreds (Agutter 2020). Vidare anses det också olämpligt att använda sig av begreppet ärende för att benämna en incident, det går att argumentera för att det i en felhanteringsprocess rimligen bör vara ett ärende som hanteras genom hela processen och att en incident tillhör eller ryms inom ärendet som sådant. Enligt Svenska Akademiens Ordböcker (u.å.) definieras ärende som ett mer eller mindre avgränsat uppdrag som kräver förflyttning för att utföra. Det anses därför att det rimligen är mest lämpligt att benämna aspekten incident med begreppet fel, alternativt defekt då dessa begrepp åsyftar likartad mening (Svenska Akademiens Ordböcker u.å.). Vi har valt att använda begreppet fel då denna benämning återges flest gånger under studien och har en något mildare negativ klang.

Inom aspekten problem känns det viktigt att belysa problematiken som uppstår genom att olika begrepp används lättvindigt och därigenom resulterar i en otydlighet kring vad som de facto avses och därmed är det olyckligt när begrepp som ärende även förekommer som benämning av ett problem, detta då ärende som tidigare nämnts bör åsyfta själva hanteringen inom hela felhanteringsprocessen, och inte syfta på det faktiska felet eller det enstaka problemet utan processen som överspänner båda aspekterna. Likväl är språkbruk och ordval generellt viktigt att bruka med förnuft och i samförstånd för att undvika missförstånd och eventuella konflikter. Att arbeta enligt en enhetlig struktur samt med en gemensam vokabulär är ett koncept som förordas av Black et al. (2007). Därmed ter det sig lämpligt att kringgå benämningen ärende även för problem aspekten och benämningen bugg blir då den som enligt studien skall vara gällande. Svenska Akademiens Ordböcker (u.å.) definierar ordet bugg som "(programmerings)fel i datorprogram".

Vid diskussion av vilket begrepp som vore lämpligt att använda för aspekten change så framkommer att alla benämningar har sina brister. När begreppet släpp används för att åsyfta avslutet på change processen genom produktionssättandet av en utvecklad lösning kan det uppfattas finnas en brist på intresse från utvecklarnas sida att ta ägandeskap för vad som inträffar efter det att lösningen släppts till produktion. Tolkningar av begreppet att släppa kan förstås som att lösningen lämnats över i någon annans händer och därav har man själv inte kvar något ansvar i frågan (Svenska Akademiens Ordböcker u.å.). Benämningarna release och stagea kan anses vara direkta låneord från engelskan som vid översättning tolkas olika av olika personer. Detta innebär att dessa begrepp inte heller kan rekommenderas som lämpliga benämningar för change aspekten i denna studie. Begreppet åtgärda har en underton som implicit menar att något är fel och detta behöver någon lösa. Begreppet har därför en negativ klang och bör även det undvikas. Det begrepp som identifierats som gångbart för att minska tendensen till konflikten är implementering som anspelar på att tillföra något i mer positivt ordalag (Svenska Akademiens Ordböcker u.å.). Begreppet implementering är gångbart i både fallet att implementera en ny lösning och att implementera en förändring och är varken beskyllande eller tolkningsbart.

## 5.2 Strategi

Samtliga respondenter inleder sin beskrivning av felhanteringsprocessen genom att förklara att processen initieras av slutanvändarna och registreras i en single point of contact. Handhavandet går sedan vidare i felhanteringsprocessen genom att kategoriseras och prioriteras som buggar i en backlogg hos utvecklingsteamet. Detta arbetssätt går i linje med incident management inom ITIL ramverket (Agutter 2020).

Buggar kan även uppmärksammas direkt via loggranskning som utförs av utvecklingsteamet själva, detta kan anses bryta mot ITSM triangeln då ingen incident förestår buggen men kan också ses som att ITSM triangeln inte har en definierad startpunkt utan att startpunkten är dynamisk och kan återfinnas i samtliga aspekter inom fenomenet. Detta arbetssätt bekräftas också som ett sätt att inom DevOps förekomma konflikten med driften genom att förebygga att fel upptäcks i produktionsmiljön (Wiedemann et al. 2020).

Buggarna utreds vidare för att identifiera en rotorsak och komma fram till en lösning på buggen. Lösningen utvecklas och testas sedan i åtgärdsrealiseringsprocessen utav utvecklingsteamet alternativt åtgärdas buggen utan kodmässig förändring i form av omläsning av fil eller korrigerigering av data (Okes 2019). Detta arbetssätt går i linje med problem management inom ITIL ramverket (Agutter 2020). Ur ett annat perspektiv beskrivs denna del av processen som ett försök till att skapa samsyn i vad som är önskat läge respektive nuläge samt att utredningen syftar till att identifiera åtgärder för att ta sig från nuläge till önskat läge. Flertalet respondenter uppger att hela buggutredningsprocessen ägs av en tvärfunktionell grupp som arbetar med både utveckling och drift av den komponent som identifierats orsaka buggen. Detta understryker att samtliga utvecklingsorganisationer i studien använder sig av någon form utav DevOps struktur för arbetet med denna aspekt inom ITSM triangeln. Jabbari et al. (2016) definierar arbetsstrukturen i en DevOps-organisation som en samling av tvärfunktionella team som arbetar med både utveckling och drift av IT-system.

Den färdigutvecklade lösningen testas av utvecklingsteamet, slutanvändare eller en separat testfunktion i en enskild produktionslik miljö för att därefter bli godkänd för att implementeras i produktionsmiljön som ett sista steg i åtgärdsrealiseringsprocessen. Detta arbetssätt går i linje med change enablement inom ITIL ramverket (Agutter 2020). Dock har det framkommit att vissa typer av rättningar på buggar inte kräver kodförändring i systemet och dessa

buggrättningar skulle då kringgå aspekten change enligt ITIL:s definition. Det går dock argumentera för att dessa filinläsningar eller data korrigeringar också borde anses som implementation av förändring i produktionsmiljön då det är någon form av manipulation av produktionsmiljön som kräver verifikation och kan leda till följd effekter i produktionsmiljön (Okes 2019). När åtgärden tagits fram finns det olika vägar att realisera den i produktionsmiljön. Vid prioriterade buggar menar de flesta respondenterna att en åtgärd skall implementeras i produktionsmiljön så fort som möjligt och då kan testförfarandet i dessa fall komma att enbart utföras av utvecklingsteamet innan implementation, vilket stöds av CI/CD konceptet inom DevOps (Jabbari et al. 2016). Detta samtidigt som majoriteten av respondenterna menar att mindre åtgärder kan vänta till nästa planerade implementation i samband med övriga förändringar som planerats.

Till synes följer samtliga respondenters svar den utstakade vägen genom ITSM triangeln när de arbetar strategiskt med sin felhanterings- och åtgärdsrealiseringsprocess. De presenterar också olika strategier som är sprungna ur DevOps olika koncept för att överbrygga hinder som kan ge upphov till konflikt inom detta fenomen.

### **5.3 Kommunikation**

Av svaren att döma är det bara ett par respondenter som uppger att de i samtliga fall när de har löst ett ärende kommunicerar ut detta till slutanvändarna. En annan respondent kommunicerar endast ut information till slutanvändare i begränsad utsträckning, till exempel vid större incidenter. Endast i en verksamhet hanteras arbetet med samtliga buggar genom en kontinuerlig dialog mellan utvecklingsteamet som arbetar med buggen och slutanvändaren som registrerat ärendet. Att överhuvudtaget inte efterfråga feedback från slutanvändare leder till brister i kontrollen av huruvida en implementation av en lösning på ett identifierat och utrett problem faktiskt har avhjälpt felet som inrapporterats (Faustino et al. 2022).

Det återfinns svar som nämner att de i organisationen skickar tillbaka restpunkter från en change till problemutredningen för att de skall fortsätta arbetet med dessa buggar och rättningen på felet får skjutas upp till nästkommande implementation. Majoriteten av respondenterna understryker dock vikten av tvärfunktionell kommunikation som en kritisk framgångsfaktor för att lyckas överbrygga konflikten mellan utvecklare och drift. Detta understryks även av Weidemann et al. (2020) som en kritisk framgångsfaktor för att överbrygga kommunikationsbrister.

### **5.4 Konflikt**

I svaren från respondenterna visar det sig finnas ett visst beteende där utvecklare inte aktivt söker feedback om hur väl en implementerad lösning bidragit med förväntad funktionalitet. Vilket tar sig i uttryck genom att begreppet släpp används vid implementering av en change och det här återspeglar sig i respondenternas svar i att släppa verkar ha en underbyggd mening om att gå fri från vidare ansvar. Detta kan definitivt anses vara en katalysator till konflikten mellan utvecklare och drift då detta kan ses som ett tecken på att intresset saknas från utvecklarsidan för huruvida driftens pågående incidentarbete har fått en lösning i system eller ej.

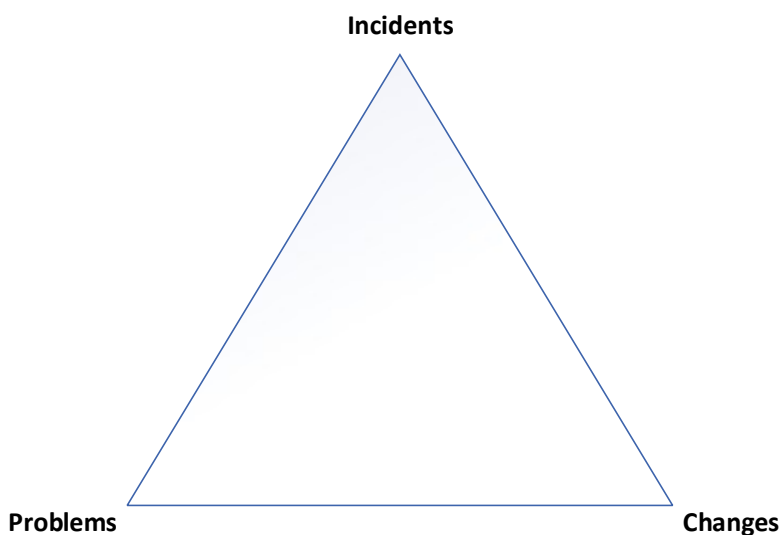
Vissa strategier som respondenterna lyft fram under intervjuerna upplevs inte direkt stödja idén om tvärfunktionell samsyn. I problemutredningsfasen av felhanteringsprocessen uppges av respondenterna att det finns en konflikt mellan utvecklare och drift, konflikten upplevs grunda sig i kunskapsluckor mellan grupperna beträffande varandras kompetensområden. Till exempel är det inte ovanligt att utvecklare inte besitter tillräckligt djup verksamhetsförståelse för att

lösningarna som tas fram ska både avhjälpa felet och samtidigt skall fungera smidigt driftsmässigt. Likaså kan drift teamen sakna en djupare systemförståelse med systemkunskap på detaljnivå vilket skulle kunna gjort det möjligt att föra en mer nyanserad dialog vid framtagning av lösningar. Det återfinns i svaren från respondenterna att man är medveten om problematiken och tillhörande konflikt, men det är endast i ett fåtal av svaren som det återges en erforderlig strategi som aktivt bemöter problematiken och försöker bidra till kunskapsspridning inom teamen och verksamheten. Samsynen mellan teamen tillika samarbetet får sig en törn när inte teamen respekterar varandras intressen i frågan och konflikten blir då ett faktum. Inom DevOps är andemeningen att utvecklingsteam och drift ska arbeta tillsammans genom kontinuerlig dialog och gemensamma arbetsrutiner samt förståelse för varandras uppgifter (Jabbari et al. 2019; Weidemann et al. 2020).

Beträffande förändringsarbete återfinns i respondentsvaren en underton av att det finns ett bristande intresse hos utvecklare att ta ägandeskapet över lösningens hela livscykel, närmare bestämt efter det att lösningen implementerats till produktionsmiljö. Det finns anledning att misstänka att användandet av strategier som inte direkt lägger någon större vikt vid samordning och att hämta in feedback från slutanvändare efter åtgärdsrealisering spår på konflikten mellan utvecklare och drift. Faustino et al. (2022) belyser vikten av att strategiskt arbeta med en feedbackloop i sin verksamhet, framför allt när verksamheten arbetar utefter DevOps koncepten.

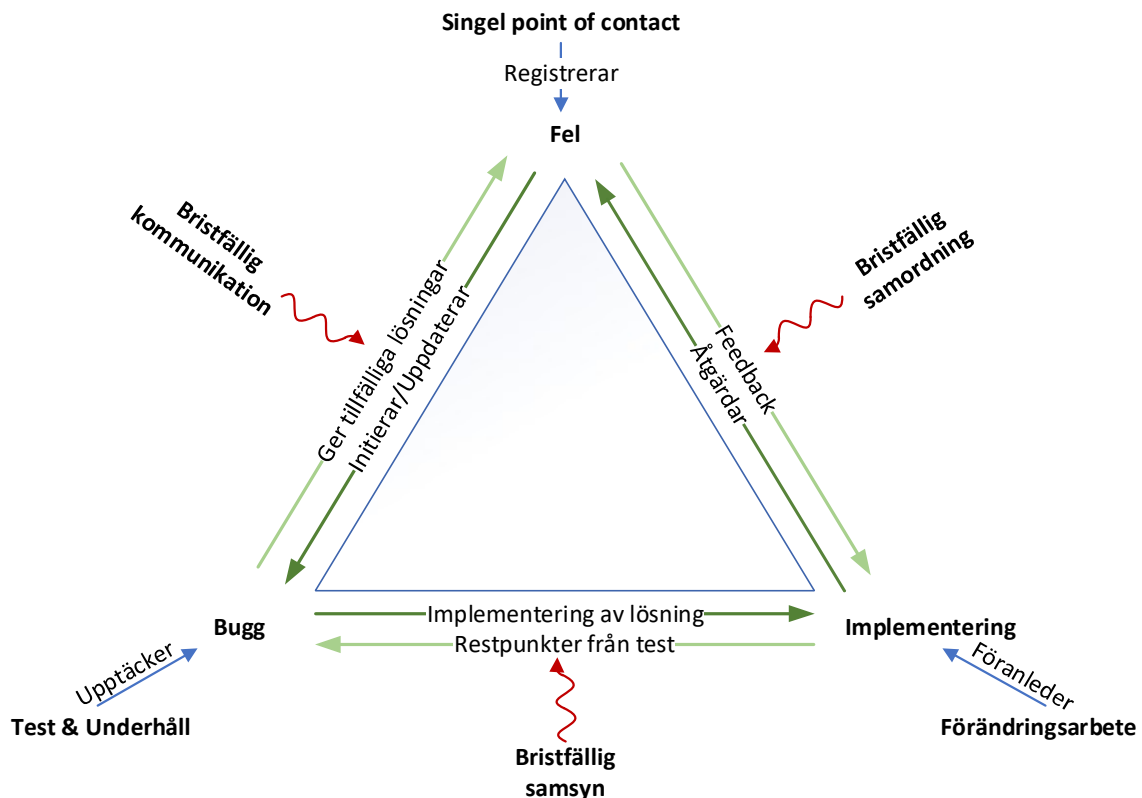
## 5.5 Sammanfattning av analys

Även om alla verksamheter har olika delar som de brister på så har vi genom sammanställning av identifierade brister modifierat den klassiska bilden av ITSM triangeln som tidigare lyfts upp i kapitel 1 (Figur 1) för att arbeta vidare med den i ett försök att belysa konfliktens omfattning i modellen.



Figur 1: The ITSM iron triangle (McLean 2012)

Denna nya modell (Figur 3) förespråkar ökad kommunikation och en gemensam terminologi för hantering av samtliga delar av det studerade fenomenet.



Figur 3: Omarbetad version av ITSM triangeln modifierad från McLean (2012)

Enligt analysen som redogjorts för ovan har den nya modellen (Figur 3) tillskrivits de nya benämningarna fel, bugg och implementering i stället för de ursprungliga ITIL-benämningarna av aspekterna. Ett tillägg till ursprungsmodellen är tre ingående stimuli som var för sig kan initiera processerna inom fenomenet. Fel registreras av en single point of contact, buggar upptäcks genom test och underhåll av systemet samt implementering av förändring föranses av förändringsarbete i IT-systemet. Ytterligare ett tillägg är de störningar som påverkar flödet mellan fenomenets aspekter. Dessa störningar grundar sig till stor del i bristen på kunskapsspridning mellan utvecklare och drift och uttrycker sig i form av följande:

- Bristfällig kommunikation mellan de som registrerar fel i systemet och de som skall utreda dem.
- Bristfällig samsyn mellan de som utreder problem och de som slutligen beslutar om att implementera lösningen
- Bristfällig samordning när det kommer till att utreda om implementeringen av lösningen faktiskt åtgärdat det upplevda felet i IT-systemet.

## 6 Slutsatser

Våra slutsatser är att det har tydligt framkommit under studien att flera av de brister som bidrar till konflikten har att göra med missförstånd eller avsaknad av en gemensam terminologi alternativt strategi för att kommunicera mellan eller inom grupperna i en IT-organisation. I Figur 3 har slutsatserna som framkommit i studien sammanställts med ITSM triangeln som grund, för att på ett visuellt sätt kunna återge en bild av hur de identifierade terminologiska skillnaderna och de dokumenterade strategierna samverkar för att konflikten mellan utvecklare och drift skall uppträda.

Dock kan studien inte presentera hur de konkreta åtgärderna skulle utformas utan lämnar detta till framtida forskning att utreda. Ett förslag till fortsatt forskning är att validera den omarbetade modellen av ITSM-triangeln genom att tillämpa den i andra studier. Dessa studier skulle kunna ta sig i uttryck av till exempel en kvantitativ studie över vilken terminologi som faktiskt är lämplig att använda och bör tillskrivas modellen. En kvalitativ undersökning kring varför DevOps kärnkoncept med kontinuerlig dialog och samverkan visat sig svårt att implementera och efterleva i IT-organisationer. Med denna fortsatta forskning skulle konflikten mellan utvecklare och drift mycket sannolikt kunna reduceras utifrån det resultat som framkommit och presenterats i denna studie.

Forskarna vill tacka alla respondenter som bidragit till studien och hoppas att ingen har tagit illa vid sig i att deras utsago använts på det sätt som gjorts i studien.

## 7 Referenser

Agutter, C. (2020). *ITIL® 4 Essentials: Your essential guide for the ITIL 4 Foundation exam and beyond*, second edition. IT Governance Publishing.

Axelos (2019). ITIL® Foundation ITIL 4 edition. <https://purplegriffon.com/downloads/resources/itil4-foundation-glossary-january-2019.pdf> [2024-04-05]

Beetz, F. & Harrer, S. (2022). GitOps: The Evolution of DevOps? *IEEE Software*, 39(4), s. 70–75. doi:10.1109/MS.2021.3119106

Black, J., Draper, C., Lococo, T., Matar, F. & Ward, C. (2007). An integration model for organizing IT service management. *IBM Systems Journal*, 46(3), s. 405–422. doi:10.1147/sj.463.0405

Cronholm, S., Göbel, H. & Cao, Lu (2023). Abductive Design Science Research: The Interplay between Deduction and Induction. I *Australasian Conference on Information Systems*, Wellington, Nya Zeeland 2023 6 december 2023, artikel 21 s. 1-9.

Cronholm, S. & Persson, L. (2016). Best Practice in IT Service Management: Experienced Strengths and Weaknesses of Using ITIL. I *International Conference on Management, Leadership & Governance*. Sankt-Petersburg, Ryssland 14–15 april 2016, artikel 7 s. 60–67.

Eaddy, M., Zimmermann, T., Sherwood, K. D., Garg, V., Murphy, G. C., Nagappan, N., & Aho, A. V. (2008). Do Crosscutting Concerns Cause Defects? *IEEE Transactions on Software Engineering*, 34(4), s. 497–515. doi:10.1109/TSE.2008.36

Faustino, J., Adriano, D., Amaro, R., Pereira, R. & Silva, M. M. (2022). DevOps benefits: A systematic literature review. *Software, Practice & Experience*, 52(9), 1905–1926. doi:10.1002/spe.3096

Gregory, R.W., Kaganer, E., Henfridsson, O. & Ruch, T.J. (2018). IT consumerization and the transformation of IT governance. *MIS Quarterly*, 42(4), s. 1225-1253. doi:10.25300/MISQ/2018/13703

Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Pearson Education, Limited.

Jabbari, R., Ali, N., Petersen, K. & Tanveer B. (2016) What is DevOps? A Systematic Mapping Study on Definitions and Practices, I *XP '16 Workshops: Proceedings of the Scientific Workshop Proceedings of XP2016* Edinburgh, Skotland 24 maj 2016, artikel 12 s. 1-11 doi:10.1145/2962695.2962707

Jacobsen, D. & Andersson, S. (2017). *Hur genomför man undersökningar?: Introduktion till samhällsvetenskapliga metoder*. Studentlitteratur AB

Kristen, V. (2022). *Teamleader: Förser teamet med riktning, instruktioner & coaching*. <https://projektleddning.se/teamleader/> [2024-03-04]

- Lubis, M., Annisyah, R. C., & Winiyanti L. L. (2020) ITSM Analysis using ITIL V3 in Service Operation in PT.Inovasi Tjaraka Buana. *IOP Conference Series: Materials Science and Engineering*, 847(1), 12077-12084. doi:10.1088/1757-899X/847/1/012077
- McCarthy, M., Herger, L., Khan, S. & Belgodere, B. (2015). Composable DevOps: Automated Ontology Based DevOps Maturity Analysis. *IEEE International Conference on Services Computing*, 600-607. doi:10.1109/SCC.2015.87
- McLean, D. (2012). *The ITSM Iron Triangle*. IT Governance Publishing
- Mintzberg, H. (1987). The Strategy Concept 1: Five Ps For Strategy. *California Management Review*, 30(1), 11–24. doi:10.2307/41165263
- Okes, D. (2019). *Root cause analysis : the core of problem solving and corrective action*. 2 uppl. ASQ Quality Press.
- Orta, E., & Ruiz, M. (2019). Met4ITIL: A process management and simulation-based method for implementing ITIL. *Computer Standards and Interfaces*, 61, 1–19. doi:10.1016/j.csi.2018.01.006
- Ravichandran, A., Taylor, K., & Waterhouse, P. (2016). *Devops For Digital Leaders : Reignite Business With a Modern Devops-Enabled Software Factory*. Apress.
- Ruiz, M., Moreno, J., Dorronsoro, B., & Rodriguez, D. (2018). Using simulation-based optimization in the context of IT service management change process. *DECISION SUPPORT SYSTEMS*, 112, 35–47. doi:10.1016/j.dss.2018.06.004
- Svenska Akademiens Ordböcker (u.å.). Svenska Akademin. <https://svenska.se/> [2024-05-15]
- Swanson, B & Beath, C (1989). Reconstructing the Systems Development Organization. *MIS Quarterly*. 13(3), s. 293–307.
- Säfsten, K., & Gustavsson, M. (2019). *Forskningsmetodik: för ingenjörer och andra problemlösare*. Studentlitteratur AB
- The SFIA Foundation (u.å.). *Asset management ASMG*. <https://sfia-online.org/en/sfia-8/skills/asset-management> [2024-03-04]
- Tonnquist B. (2024). *Projektledning*. Sanoma utbildning
- Waller, J., Ehmke, N. C. & Hasselbring, W (2015). Including Performance Benchmarks into Continuous Integration to Enable DevOps. *Software Engineering Notes*. 40(2), s. 1-4. doi:10.1145/2735399.2735416
- Walker, J. (2022). *Development vs Operations: Where Do “Dev” and “Ops” Roles Differ and Overlap?* <https://www.howtogeek.com/devops/development-vs-operations-where-do-dev-and-ops-roles-differ-and-overlap/> [2024-03-08]
- Wang, Y. & Cao, J. (2021). Well Begun Is Half Done: How First Responders Affect Issue Resolution Process in Open Source Software Development? *In Communications in Computer and Information Science*, 1330, s. 628–638. doi:10.1007/978-981-16-2540-4\_46

Wiedemann, A., Wiesche, M., Gewalt, H. & Krcmar, H. (2020). Understanding how DevOps aligns development and operations: a tripartite model of intra-IT alignment. *European Journal of Information Systems*, 29(5), s. 458-473. doi:10.1080/0960085X.2020.1782277

Wright, S. A., & Druta, D. (2014). Open source and standards: The role of open source in the dialogue between research and standardization. I *2014 IEEE Globecom Workshops (GC Wkshps*, Austin TX, USA, 8-12 december s. 650–655. doi:10.1109/GLOCOMW.2014.7063506



## 8 Bilagor

### 8.1 Bilaga 1: Arbetsblad för sökord

#### 1 Formulera din forskningsfråga.

Kan fenomenet ITSM triangeln förklara konflikten mellan utveckling och förvaltning?

#### 2 Plocka ut huvudbegreppen i forskningsfrågan. Översätt sedan dessa till engelska.

Huvudbegreppen bildar olika "block" i din sökstrategi:

<b>Svenska</b>	Huvudbegrepp 1 på svenska ITSM	Huvudbegrepp 2 på svenska Utveckling	Huvudbegrepp 3 på svenska Förvaltning	Huvudbegrepp 4 på svenska Konflikt
<b>Engelska</b>	Huvudbegrepp 1 på engelska ITSM	Huvudbegrepp 2 på engelska Development	Huvudbegrepp 3 på engelska IT Operation	Huvudbegrepp 4 på engelska Conflict

#### 3 Analysera nu huvudbegreppen närmare. Hitta synonymer och ämnesord (t ex Subject).

Huvudbegrepp 1 Information technology service management IT-service management Incident management Problem management Change enablement ITIL DevOps key concepts	Huvudbegrepp 2 Release Solution Fix Software development	Huvudbegrepp 3 IT governance Investigation "Root Cause" Concern	Huvudbegrepp 4 Cross functional alignment Operational alignment Lean IT
--	--	---	--



## 8.2 Bilaga 2: Intervjuguide

### Introduktion

Vi studerar på Högskolan i Borås till Systemvetare med inriktning mot Systemutveckling. Skriver nu en kandidatuppsats inom informatik.

Jag heter Robert Keränen och har flera års arbetslivserfarenhet av IT som supportspecialist och har jobbat i flertalet interna IT-projekt.

Jag heter Jonathan Öberg och har lång erfarenhet inom logistik och e-handel men har också jobbat deltid i interna IT-projekt.

Vår undersökning skall utreda och hämta in begrepp och förhållningssätt till felhantering och åtgärdsrealisering inom systemutveckling och IT-förvaltning.

Vi har själva av eget intresse för ämnet initierat denna studie för att undersöka och få djupare förståelse för dessa processer.

Intervjuerna kommer att spelas in och material kommer samlas in genom 4–6 intervjuer. Insamlat material kommer att sammanställas anonymiserat, med endast en koppling till din yrkestitel, av oss i syfte att framställa en kandidatuppsats som kommer att publiceras av Högskolan i Borås och finnas allmäntillgänglig via Diva-portalen.

Du kommer att omnämnas som: ”Respondent X som arbetar som [yrkestitel]”

Vi räknar med att den här intervjun ska ta cirka en timme. Vi fortsätter i alla fall inte längre än till klockan xx.

Har du några frågor innan vi börjar?

### Allmänna frågor

---

*Mall för frågeställningens upplägg:*

- *Öppen fråga till respondenten*
  - *Eventuella uppföljningsfrågor om respondenten inte svarar alls på frågan*

- 
- Vilken är din yrkestitel?
  - Hur lång erfarenhet har du av IT-relaterat arbete?
  - Vad är ert företags primära arbetsområde?
    - Utveckling/Förvaltning/Konsult



## Intervjufrågor

- IF1. När vi nämner felhantering vad tänker du på då?
- Kan du redogöra för felhanteringsprocessen hos er?
  - Beskriv hur verksamheten blir uppmärksam på att fel uppkommit i produktionsmiljön?
  - Hur vet ni vad som skall arbetas med i första hand och vad som kan vänta?
- IF2. När vi nämner åtgärdsrealisering vad tänker du på då?
- Hur går ni till väga för att rätta fel i er produktionsmiljö?
- IF3. När åtgärder lyfts in i produktionsmiljö, följer ni upp och utvärderar utfallet av åtgärden?
- IF4. Samarbetar ni tvärfunktionellt mellan systemutvecklingsteam och förvaltningsteam?
- Hur ser detta samarbete ut?
- IF5. Vad skulle du säga är den största utmaningen med att jobba tvärfunktionellt mellan systemutvecklingsteam och förvaltningsteam?
- Jobbar ni aktivt för att brygga ev. utmaningar?
- IF6. Är det avslutningsvis något du vill säga som vi inte har kommit in på angående felhantering eller åtgärdsrealisering?

---

## Avslut (tillkom efter första intervjun)

- Känner du till ramverket ITIL?
- Känner du till DevOps som arbetssätt?



## 8.3 Bilaga 3: Respondentförteckning

Respondent	Yrkestitlar	Verksam inom	Bransch- erfarenhet	Känner till ITIL	Känner till DevOps
Respondent F	Utvecklingschef, Utvecklare	Utveckling	7år	Ja	Ja
Respondent K	Product Owner, Förvaltningsledare, Functional owner, Business process developer	Utveckling och Förvaltning	15år	Ja	Ja
Respondent L	Business Analyst, Scrum Master, Agile Coach	Utveckling	40år	Ja	Ja
Respondent M	IT-tjänsteansvarig, IT-projektledare, systemförvaltare	Förvaltning	24år	Ja	Nej
Respondent R	Scrum Master, Kvalitetssäkrare, Testledare, Enhetschef	Utveckling och Förvaltning	25år	Nej	Ja
Respondent S	Utvecklingschef, Team Leader, Systemarkitekt, Utvecklare	Utveckling	10år	Ja	Ja



## 8.4 Bilaga 4: Begreppslista

Begrepp	Förklaring
Bugg	Ett systemfel som ger upphov till kapacitetsförluster i ett IT-system.
drift	De arbetsprocesser och verktyg som används för att underhålla och förändra redan produktionsatta mjukvarulösningar i en organisation.
IT-förvaltning	Den verksamhetsdel med tillhörande resurser som utför arbetsprocesserna inom drift.
Kapacitetsförluster	Händelser som ger upphov till minskad förmåga för systemet att hantera förfrågningar från användare alternativt integrationer av andra system.
Produktionsmiljö	Den del eller miljö av ett IT-system som användare arbetar inuti, det vill säga en verksamhets aktiva IT-system som hanterar produktionsmässiga data.
Rotorsak	Den faktiska anledningen till uppkomsten av en bugg i ett IT-system.
Trackers	Är en yrkesroll där man jobbar som testare av ett system där man inte har fått några förkunskaper om systemets uppbyggnad men ändå fått instruktioner om vad för funktionalitet man skall testa och sedan utför man test på systemet ad hoc för att se om lösningen är intuitiv och uppfyller kraven på funktionalitet.



### 8.5 Bilaga 5: Generativ AI-Intyg



#### Generativ AI Användningsrapport

Denna mall för Generativ AI Användningsrapport har utformats för att ge studenter inom informationsteknologi och relaterade områden ett strukturerat ramverk för att rapportera om användningen av generativ artificiell intelligens (AI) i deras uppsatser. Syftet med denna mall är att transparent kommunicera var och hur generativ AI har inkorporerats i uppsatsen, samt att belysa anledningen och utsträckningen av dess användning.

<b>Uppsatsens Titel</b>	Fenomenet "THE ITSM IRON TRIANGLE" – en medlare eller uppviglare
<b>Studentens Namn</b>	Student 1: Jonathan Öberg Student 2: Robert Keränen <del>Student 3:</del>
<b>Datum</b>	2024-05-15

Uppsatsens delar	Jag/vi har använt Generativ AI i delar av uppsatsen.		Motivation (Utsträckning samt anledning)
	Ja	Nej	
Introduktion	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Teori	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Metod och genomförande	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Resultat och analys	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Diskussion	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Slutsatser	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Övrigt	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vi har med hjälp av AI randomiserat respondenterna



# HÖGSKOLAN I BORÅS

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