



AKADEMIN FÖR TEKNIK OCH MILJÖ
Avdelningen för industriell ekonomi, industridesign och maskinteknik

Implementering av DevOps

En fallstudie på ett IT-konsultföretag

Lilly Carlsson
Jeanette Langsager

2021

Examensarbete, Grundnivå (kandidatexamen), 15 hp
Industriell ekonomi
Ekonomiingenjörsprogrammet

Handledare: Carina Dahlberg
Examinator: Roland Hellberg

Förord

Detta examensarbete är ett avslutande moment av högskoleingenjörsprogrammet med inriktning Industriell ekonomi på Högskolan i Gävle. Det har varit ett utvecklande, lärorikt och spännande arbete.

Ett stort tack ska riktas till fallföretaget och den personal som vi varit i kontakt med. Vi uppskattar den positivitet vi bemöts av och det goda samarbetet som har funnits. Vi vill också tacka vår examinator Roland Hellberg som utmanat oss att utveckla våra kunskaper. Ett särskilt stort tack till vår handledare Carina Dahlberg vars engagemang, glada humör och flexibilitet har förenklat samt förbättrat detta arbete. Slutligen tackar vi varandra för ett utmärkt samarbete där vi uppskattat och stöttat varandra.

Abstract

Within the field of project management different methods and agile approaches have emerged. However, dividing silo-structures have created frustration within software development. DevOps has surfaced as a way to address this and instead connect different functions such as development and *operations*. DevOps affects teams but there is no standardized approach for its implementation. Therefore, more research around the implementation process of DevOps is needed. Also, investigating what the accompanying effects could be, is of interest. The purpose of this thesis is to study the implementation of DevOps within the IT consultant industry and what impact it has on a team.

This thesis was executed with abductive reasoning where a case study was carried out. To fulfill the purpose and answer the research questions literature review and interviews have been performed. Collected theory and empirical material have been discussed and analyzed in order to formulate conclusions and recommendations.

One formulated conclusion is that each implementation of DevOps needs to follow its own adapted execution where the aim and essence are clarified. Factors regarding communication, automation, responsibility, willingness to change and collaboration also impact the implementation and the effects experienced by the team. It can also be concluded that both benefits and limitations can affect the team yet, the disadvantages often appear to be contained within the start of the implementation. A practical contribution comes from recommendations for the implementation, such as having a dedicated person and education within DevOps among others.

Keywords: *DevOps, DevOps implementation, DevOps team, agile methods*

Sammanfattning

Inom projektledning har olika metoder och agila arbetssätt utvecklats. Det har dock funnits frustration över separerande silostrukturer inom mjukvaruutveckling. DevOps har utvecklats som ett sätt att bemöta detta och istället sammankoppla olika funktioner som utveckling och *operations*. DevOps har effekter på ett team men det finns inget standardiserat tillvägagångssätt för dess implementering. Det behövs därför mer forskning kring implementeringsprocessen av DevOps. Det kan även vara av intresse att undersöka vilka de medföljande effekterna kan vara. Syftet med detta arbete är studera genomförandet av DevOps implementering inom IT-konsultbranschen och på vilket sätt det påverkar ett team.

Det här arbetet har bedrivits som en abduktiv studie där en fallstudie genomförts. För att uppfylla arbetets syfte och besvara forskningsfrågorna har litteraturstudier och intervjuer genomförts. Insamlad teori och empiri har diskuterats och analyserats för att formulera slutsatser och rekommendationer.

En formulerad slutsats är att varje enskild implementering av DevOps behöver följa ett eget anpassat genomförande där syfte och innebörd tydliggörs. Faktorer inom kommunikation, automation, ansvar, förändringsvillighet och samarbete har också inverkan på implementeringen och de effekter teamet kan uppleva. Det kan även konkluderas att både fördelar och nackdelar kan påverka teamet men nackdelarna framstår ofta vara begränsade till implementeringens uppstart. Det praktiska bidraget utgörs av rekommendationer för implementeringen såsom att till exempel ha en dedikerad person och utbildning inom DevOps.

Nyckelord: *DevOps, DevOps implementering, DevOps team, agila metoder*

Begreppslista

Agil – Från engelskans *agile* som betyder snabb, lätt och smidig. Ordet kan användas för att beskriva ett projekts genomförande.

DevOps – Begreppet är en förkortning av *development* och *operations*.

Development & operations – *Development* och *operations* är två olika funktioner inom en organisation. *Development* är systemutveckling, själva kodskrivandet för att lösa en kunds problem. Det som sedan följer med distribution, installation och övervakning av funktionalitet hör till *operations*. I detta arbete används både ordet *development* och dess översättning som är utveckling, medan *operations* endast anges i sin ursprungliga form.

Inkrementell – Kommer från engelskans *incremental*, vilket betyder med successiva tillägg.

Iterativ – Innebär att iteration sker, vilket betyder att något upprepas.

Jira – Programvaran Jira kan användas för problemspårning och projektledning inom agila utvecklingsteam.

Kanban-tavlor – Kanban-tavlor är ett visualiseringsverktyg inom Lean. Uppgifter delas upp i tre fält efter vad som ska göras, vad som är pågående och vad som har gjorts.

Lean – Lean är en metod som handlar om effektivisering genom att minska slöserier, implementera ständiga förbättringar och respekt för individen.

Outsourcing – Ett begrepp som innebär att projekt eller uppgifter utförs utanför den egna organisationen.

Silostrukturer – Begreppet syftar till när ett team eller en grupp inte delar information, mål, prioriteringar, verktyg och processer med andra team eller grupper.

Tvärfunktionella team – Ett tvärfunktionellt team består av teammedlemmar från olika funktioner med olika expertisområden som tillsammans arbetar mot ett gemensamt mål.

T-shape – Begreppet hänvisar till att personer lär sig kunskaper utöver sin kärnkompetens. T:ets lodräta streck symboliserar personens expertiskunskap, eller kärnkompetens, och det vågräta strecket representerar att personen även har viss kompetens inom andra områden.

Innehållsförteckning

Förord	i
Abstract	ii
Sammanfattning	iii
Begreppslista	iv
Figur- och Tabellförteckning	1
1. Introduktion	1
1.1 Bakgrund.....	1
1.2 Syfte	2
1.3 Frågeställningar.....	2
1.4 Avgränsning	2
2. Metod	3
2.1 Vetenskapligt angreppssätt.....	3
2.2 Fallstudie.....	4
2.2.1 Fallföretaget.....	5
2.3 Primär- och sekundärdata	5
2.3.1 Litteraturstudie	5
2.3.2 Intervjuer.....	6
2.4 Studiens kvalitet	8
2.4.1 Reliabilitet	8
2.4.2 Validitet	8
2.4.3 Generaliserbarhet	9
2.5 Etiska perspektiv	10
2.6 Hållbarhetsperspektiv.....	10
3. Teori	12
3.1 Teoretisk bakgrund	12
3.1.1 Projekt	12
3.1.2 Agila manifestet	12
3.1.3 Agila metoder.....	13
3.1.4 Scaled Agile Framework.....	14
3.2 DevOps	15
3.2.1 DevOps framväxt.....	15
3.2.2 DevOps innehåll	16
3.2.3 Hur DevOps utvidgar agil arbetsmetodik.....	16
3.3 Implementering.....	17
3.3.1 Tillvägagångssätt	17
3.4 Implementeringsfaktorer.....	19
3.4.1 Kommunikationsfaktorer	19

3.4.2 Automationsfaktorer	19
3.4.3 Ansvarsfaktorer	19
3.4.4 Förändringsvillighet.....	19
3.4.5 Samarbetsfaktorer	20
3.5 Påverkan på team	20
3.5.1 Fördelar	20
3.5.2 Nackdelar	21
3.6 Sammanfattning teori.....	21
4. Empiri	24
4.1 Nulägesbeskrivning	24
4.2 Implementering.....	24
4.2.1 Information innan implementering	24
4.2.2 Tillvägagångssätt	25
4.2.3 Förbättring av implementeringsprocessen	26
4.3 Implementeringsfaktorer.....	27
4.3.1 Kommunikation	27
4.3.2 Ansvar	27
4.3.3 Förändringsvillighet.....	28
4.3.4 Samarbete	28
4.4 Påverkan på team	28
4.4.1 Fördelar	28
4.4.2 Nackdelar	30
4.4.3 Teamets förslag på förbättringar	31
5. Diskussion och analys	32
5.1 Implementering.....	32
5.1.1 Tillvägagångssätt	32
5.1.2 Förbättring av implementeringsprocessen.....	34
5.2 Implementeringsfaktorer	34
5.2.1 Kommunikation	34
5.2.2 Ansvar	35
5.2.3 Förändringsvillighet.....	35
5.2.4 Samarbete	36
5.3 Påverkan på team	36
5.3.1 Fördelar	36
5.3.2 Nackdelar	37
5.3.3 Teamets förslag på förbättringar	38
6. Slutsats	39
6.1 Teoretiskt bidrag.....	39
6.1.1 Hur kan implementeringen av DevOps genomföras?.....	39
6.1.2 Vilken påverkan har DevOps på ett team?	39
6.2 Praktiskt bidrag.....	39
6.3 Arbetets begränsningar	40
6.4 Framtida forskning.....	40
Referenser	41
Bilagor	1

<i>Bilaga 1 - Missivbrev</i>	<i>1</i>
<i>Bilaga 2 – Intervjufrågor personal på fallföretaget</i>	<i>2</i>
<i>Bilaga 3 – Intervjufrågor kundrepresentant</i>	<i>3</i>

Figur- och Tabellförteckning

Figur 1 - Studiens tillvägagångssätt. Källa: Egen	4
Figur 2 – Teamstruktur. Källa: Egen.....	7
Figur 3 - Scrumbaserad agil standardmodell. Källa: (Neve et al., 2017)	14
Figur 4 - SAFe Continuous Delivery Pipeline. Källa: (Scaled Agile Inc, 2021b)	15
Figur 5 - Agile to Devops transformation. Källa: (Gupta et al., 2019)	17
Figur 6 - DevOps Pyramid. Källa: (Muñoz & Rodríguez, 2021).....	18
Figur 7 - Struktur över teoriavsnitt. Källa: Egen	22
Figur 8 - Systemutvecklarens uppfattning av implementeringen. Källa: Egen.....	26
Figur 9 - Tolkning av fallföretagets implementeringsprocess. Källa: Egen.....	33
Tabell 1 - Informanternas befattning, arbetsgivare och antal år hos arbetsgivare. Källa: Egen.	7
Tabell 2 - Fördelar från DevOps implementeringen som nämndes under intervjuerna. Källa: Egen.....	29
Tabell 3 - Nackdelar från DevOps implementeringen som nämndes under intervjuerna. Källa: Egen.....	31

1. Introduktion

Följande text innehåller bakgrund, syfte, frågeställningar och avgränsning.

1.1 Bakgrund

Projekt är tillfälliga och nyttjar nödvändiga resurser för att lösa problem eller uppnå mål. Utföranden av projekt kan se väldigt olika ut och kan kräva olika förhållningssätt. De första projektledningsmodellerna som utvecklades var så kallade stage-gate modeller. Med planering, kontroll och målfokus kan stage-gate modeller vara lämpliga för målstyrda projekt där målet är fastställt på förhand. För mer målsökande projekt vars mål inte är lika klart bestämt i början behövs mer flexibilitet, vilket lett till de agila metoderna (Hallin & Karrbom Gustavsson, 2019).

Agila projektmetoder handlar om flexibilitet och inkrementellt genomförande. Till skillnad från de traditionella metoderna, fastställs inte projektplanen i förväg vid agilt arbete. Istället kan den justeras under projektets gång med hög grad av involvering från kunden. Fler organisationer implementerar agila metoder för att uppnå konkurrensfördelar i form av differentiering (Wiedemann, 2018). Agilt arbete är lämpligt då mål och krav är mindre tydliga, vilket kan vara en anledning till dess popularitet i dagens oförutsägbara marknad (Tonnquist, 2018).

Det finns flera olika agila metoder som bland annat inkrementell utveckling och Scrum (Tonnquist, 2018). DevOps, som är en förkortning för *development* och *operations* (Hosono & Shimomura, 2012), är ett relativt nytt fenomen inom programvaruutveckling (Wiedemann, 2018). I de flesta agila projekten inkluderas inte *operations* personal och DevOps är en nyare metodik som bemöter detta behov genom att sammankoppla utveckling och *operations* (Diel et al., 2016). DevOps är därför ett sätt att utvidga det agila arbetet och sammankoppla flera funktioner i organisation genom ett utökat samarbete. Agila metoder och DevOps modellen har under senare tid blivit alltmer attraktivt för verksamheter och forskning då det är ett sätt att underlätta anpassningen till en oförutsägbar marknad (Indriasari et al., 2020). DevOps skapar en länk mellan IT utvecklare och IT operatörer (Indriasari et al., 2020) samt utvidgar det agila perspektivet (Wiedemann, 2018).

DevOps kan beskrivas på många olika sätt men ytterligare studier behövs innan tillvägagångssättet kan standardiseras för implementeringen (Diel et al., 2016). Då det inte finns något standardutförande för implementeringen av DevOps kan dess införande försvåras (Nybom et al., 2016). För att förenkla processen krävs ytterligare kunskap och studier kring DevOps implementering. Det finns forskning som visar att förändringar sker inom team under implementeringen och dessa kan ha långvariga effekter på teamets dynamik (L. Yin & Filkov 2020). Det finns ett intresse att ta reda på hur denna påverkan kan yttras och vilka effekterna kan vara. Dessutom framhäver (Nybom et al., 2016) att erfarenheter från tidigare utförda DevOps införanden samt effekterna som upplevts kan bidra till utvecklingen av ny kunskap. Därför riktas detta arbete mot implementering av DevOps och den påverkan det kan medföra för ett team.

1.2 Syfte

Syftet med detta arbete är att undersöka hur implementeringen av DevOps kan genomföras inom IT-konsultbranschen samt hur det påverkar ett team.

1.3 Frågeställningar

- Hur kan implementeringen av DevOps genomföras?
- Vilken påverkan har DevOps på ett team?

1.4 Avgränsning

Arbetet har avgränsats till att undersöka implementeringen av DevOps hos endast ett team på ett fallföretag.

2. Metod

I detta avsnitt presenteras de metoder som har använts i studien. Ytterligare presenteras arbetets kvalitet, etiska perspektiv och hållbarhetsperspektiv.

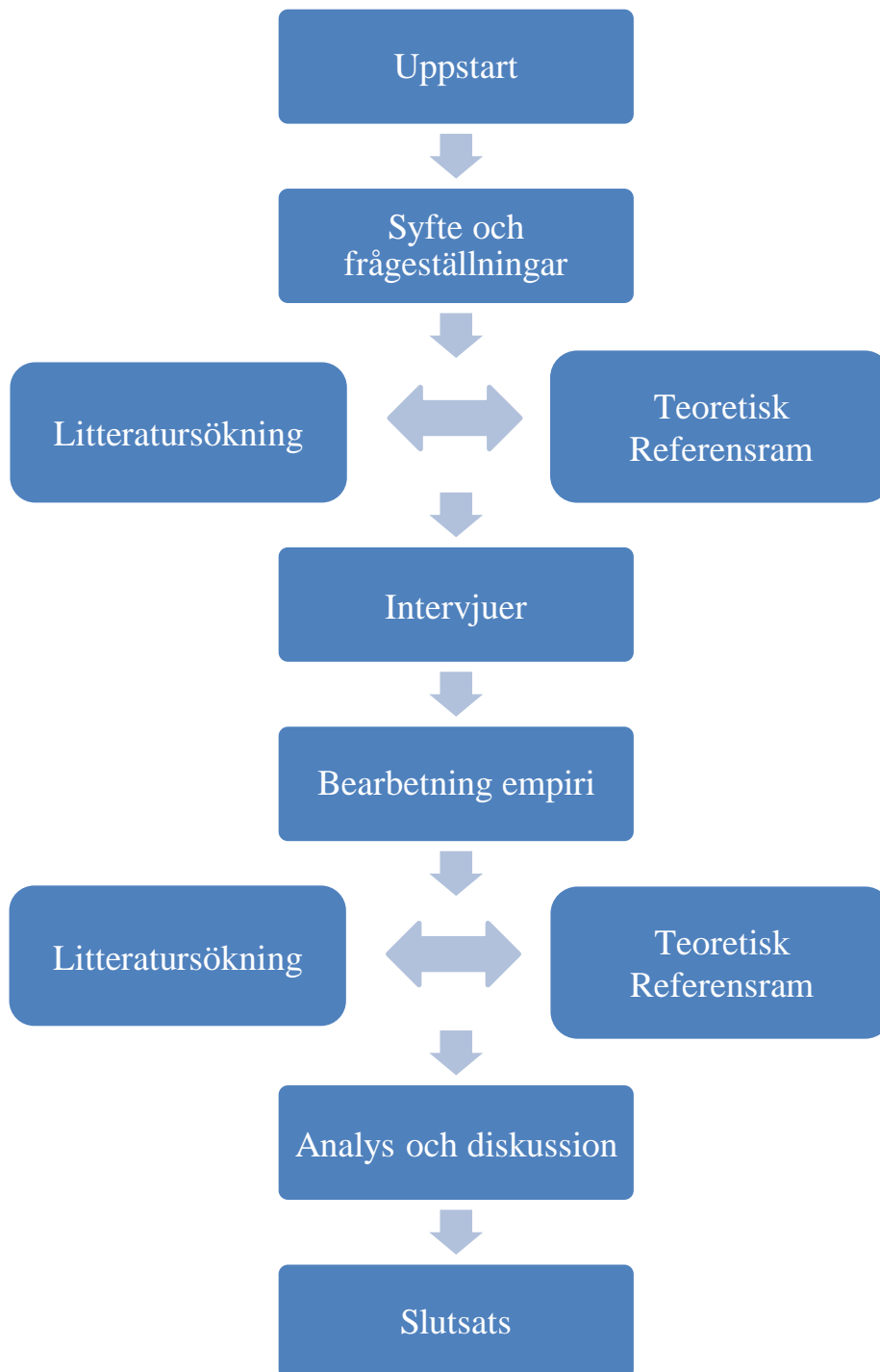
2.1 Vetenskapligt angreppssätt

För att undersöka relationen mellan teori och empiri kan tre förhållningssätt antas: deduktion, induktion samt abduktion. Deduktivt arbete utgår från bevis, allmänna principer samt befintlig teori och undersöker enskilda fall utifrån dessa. Teorin finns redan innan och styr studiens upplägg. Induktivt arbete däremot är mer upptäckande och behöver inte ha redan fastställd teori från början. Istället kan forskningsobjektet undersökas och teorin utformas utifrån den information som då insamlas. Abduktion kan ses som en kombination av deduktion och induktion. Arbetet kan då till en början spegla ett mer induktivt perspektiv där en föreslagen teoretisk struktur formuleras utifrån det särskilda fallet. Därefter övergår arbetet till att bli mer deduktivt då hypotesen testas på fler fall och utvecklar mer generellitet (Patel & Davidson, 2011). Det abduktiva förhållningssätt innebär att man växlar mellan framtagen teori och empiri för att utöka förståelsen inom det studerade ämnet (Blomkvist & Hallin, 2014).

Inhämtning av empiri kan vara antingen kvalitativt eller kvantitativt. Den kvantitativa undersökningsmetoden använder sig av hårda data såsom siffror eller annat som är mätbart, och presenteras ofta i form av tabeller eller diagram. Den kvalitativa undersökningsmetoden fokuserar istället på mjuk data som ofta måste tolkas i syfte att förstå hur ett fenomen påverkar de personer som upplever den (Blomkvist & Hallin, 2014).

Denna studie har ett abduktivt förhållningssätt. Inledningsvis var arbetet upptäckande och induktivt då teori samlades in och ett problemområde fastställdes. Detta utforskades sedan vidare genom ytterligare litteratur och intervjuer på ett mer deduktivt sätt. Denna pendling mellan induktion och deduktion framhäver arbetets abduktiva förhållningssätt. Inhämtningen av empirin har skett enligt en kvalitativ metod genom intervjuer. Detta innebär att stort fokus har legat på hur empirin tolkas och förstås. Den kvalitativa metoden är lämplig för denna fallstudie eftersom den behandlar mänskliga tankar och upplevelser som inte kan kvantifieras.

Denna studie började med utformningen av frågeställningar och syfte med forskningsgap i åtanke. Därefter insamlades relevant teori genom en litteratursökning. Syftet med litteratursökningen var att kunna förklara relevanta begrepp, men även att samla information om vad som redan är känt inom studiens område. Studien fortsatte sedan med utformningen och utförandet av intervjuer för att inhämta empiri från fallföretaget. Efter utförandet av intervjuerna bearbetades intervju svaren för att sammanställa empirin. Den sammanställda empirin lästes igenom av fallföretaget för att säkerställa att informationen inte var felaktig. Utifrån insamlad empiri kompletterades teoriavsnittet med relevant forskning genom ytterligare litteraturstudie. Den teoretiska referensramen samt den insamlade empirin resulterade i avsnittet diskussion och analys som slutligen ledde fram till slutsatsen. Arbetsgången i denna studie kan även ses i figur 1 nedan.



Figur 1 - Studiens tillvägagångssätt. Källa: Egen

2.2 Fallstudie

Denna studie genomfördes som en fallstudie. Fallstudier innebär att mindre avgränsade grupper studeras. Under en fallstudie är målet att få så bred och täckande information som möjligt. Informationssamlingen i en fallstudie kan göras på många sätt och det är inte ovanligt att kombinera både intervjuer, enkäter och observationer (Patel & Davidson, 2011). En av styrkorna vid en fallstudie är enligt Bell (1993) att forskare tillåts att fokusera specifika problem

eller utmaningar som lätt kan försvinna i stora forskningsprojekt. En utmaning som nämns är att vid dessa ofta småskaliga fallstudier kan det vara svårt att kontrollera information vilket kan leda till en förvrängning av verkligheten.

2.2.1 Fallföretaget

Fallstudien utfördes på ett IT-konsultföretag inom digitaliseringsområdet som valt att vara anonymt. 1990 var företagets etableringsår och sedan dess har företaget växt till att nu ha cirka 2600 anställda. Organisationen finns på 15 orter i Sverige, dessutom finns de i Norge, Danmark, Finland och Tyskland. Fallföretagets vision är att med hjälp av digitalisering och innovation bidra till ett hållbart och mänskligt samhälle. Affärsidén är att uppnå digitala möjligheter och långsiktigt värde genom en kombination av strategisk förmåga, teknik och kreativitet. I detta arbete studerades ett team från organisationen som finns i Sverige där konsulter hyrs ut till kunder under en längre tid. Teamet består av konsulter från fallföretaget men även personal från kunden.

2.3 Primär- och sekundärdata

Primärdata är ny data som insamlas under studiens gång men som även anpassas efter studien med avsikt att hjälpa besvara syfte och frågeställningar. Sekundärdata däremot är data som andra forskare eller institut redan har samlat in (Olsson & Sörensen, 2003).

I denna studie användes både primär- och sekundärdata. Primärdata samlades in under intervjuer, och sekundärdata samlades in under litteraturstudien.

2.3.1 Litteraturstudie

Forskning handlar om att ställa frågor och besvara dessa, därför är det logiskt att påbörja en studie genom att fråga vad som redan finns besvarat kring ämnet (Loseke, 2013). Genom att studera litteratur uppnås kunskap om vad som redan har studerats samt vad som inte har studerats. Därutöver krävs avgränsningar vid litteratursökning för att kunna sortera och finna relevant litteratur till studien (Loseke, 2013).

Under detta arbete har en litteraturstudie utförts. De vetenskapliga artiklarna som valdes ut hittades via databasen Scopus och sökmotorn Google Scholar. Sökorden som användes under den initiala litteratursökningen var DevOps och Devops AND challenge*, där asterisken gör att alla böjningar av ordet tas med. För att hitta relevanta artiklar för denna studie avgränsades sökningen till språket engelska och ämnet business, management and accounting. Samtliga artiklars abstract lästes igenom för att urskilja vilka som hade relevans för detta arbete. De artiklar som ansågs relevanta sparades och ytterligare urval gjordes utifrån avsnittet conclusion. De artiklar som fortfarande ansågs ha relevans lästes igenom fullständigt och utifrån det gjordes ett urval av litteratur. Under arbetets gång gjordes ytterligare litteratursökningar med sökord såsom bland annat DevOps implementation och DevOps AND Effect AND Team.

2.3.2 Intervjuer

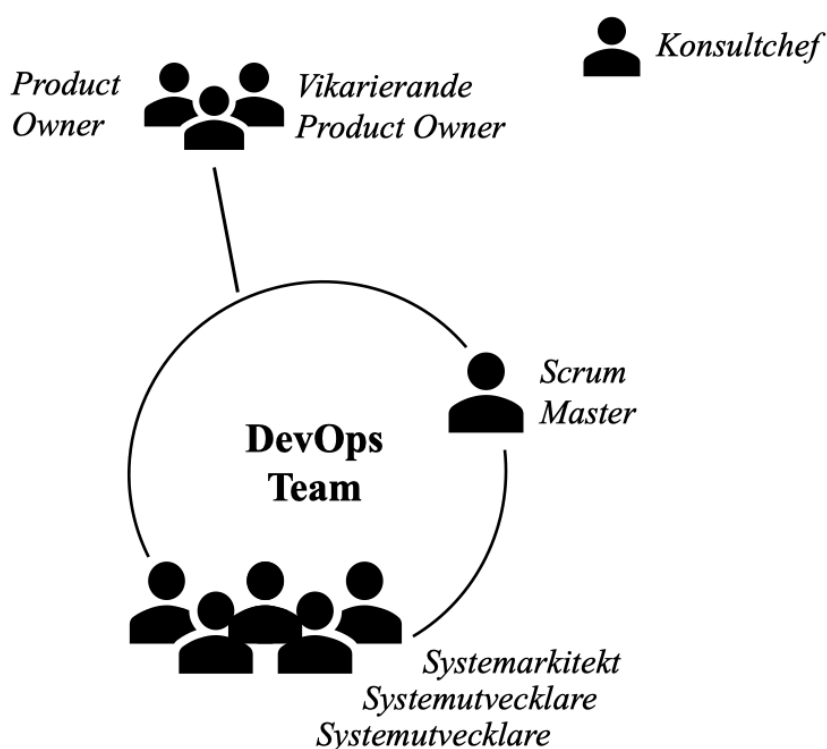
Att utföra intervjuer innebär stora fördelar och kan leda till mycket information (Bell, 1993). En intervju kan därför också handla om att avläsa attityder, känslor och föreställningar och tolkningar. Intervjuer som del av en studie kan ge en ökad förståelse och djupare kunskap. Samtidigt är det viktigt att vara öppen för överraskande och ny information (Sohlberg & Sohlberg, 2013). Det finns ostrukturerade intervjuer och strukturerade intervjuer, samt olika sammansättningar av dessa. Strukturerade intervjuer innebär att alla frågor är förbestämda och att den som intervjuar inte får avvika från dessa. Ostrukturerade intervjuer innebär att endast ämnet är förutbestämt och fungerar mer som en konversation. En klar fördel med strukturerade intervjuer är att svaren enklare kan sammanfattas och analyseras, medan ostrukturerade intervjuer är mer krävande att sammanställa och analysera. Semistrukturerade intervjuer sätter en ram för intervjun men möjliggör samtidigt informanter att fritt prata om det som ligger centralt för just den personen (Bell, 1993). Vissa frågor formuleras ofta i förhand, och kan ha en låg eller hög grad av standardisering beroende på vilken ordning frågorna ställs. Ställs alltid frågorna i samma ordning, har intervjun en hög grad av standardisering. Ställs frågorna däremot i den ordning som bäst passar in i samtalen har intervjun en låg grad av standardisering (Patel & Davidson, 2011).

I detta arbete utfördes semistrukturerade intervjuer med personal på fallföretaget samt från en av fallföretagets kunder som ingår i ett DevOps team. Semistrukturerade intervjuer innebär att vissa frågor var förutbestämda, men informanterna hade stor möjlighet att utforma svaren på egen hand. Därutöver fanns möjlighet för den som intervjuade att ställa följdfrågor och för den som intervjuades att delge ytterligare önskad information. De förutbestämda intervjufrågorna utformades med syfte samt frågeställningar i åtanke och kan ses i bilaga 2 samt 3. Innan de faktiska intervjuerna testades intervjufrågorna på en person med erfarenhet av arbete i DevOps team för att kontrollera hur relevanta svar som kunde förväntas.

Intervjuerna genomfördes digitalt med ljudupptagning för att möjliggöra transkribering. Båda författarna av denna studie deltog vid samtliga intervjuer och vardera tog mellan 10 och 30 minuter. Totalt åtta intervjuer genomfördes varav sju av informanterna var anställda hos fallföretaget och en var från kunden. En av intervjuerna med en anställd på fallföretaget valdes i efterhand bort då svaren ansågs ligga utanför arbetets område. Informanter med olika befattningar valdes för att få bredast möjliga perspektiv på teamets upplevelser. I tabell 1 kan information om informanterna ses. Den informant vars intervju inte användes är struken.

Tabell 1 - Informanternas befattning, arbetsgivare och antal år hos arbetsgivare. Källa: Egen.

Arbetsgivare	Befattning	År hos arbetsgivare
Kund	Product Owner	1 år
Fallföretaget	Vikarierande Product Owner	10 år
Fallföretaget	Konsultchef	16 år
Fallföretaget	Scrum Master	4 år
Fallföretaget	Systemarkitekt	17 år
Fallföretaget	Systemutvecklare	11 år
Fallföretaget	Systemutvecklare	4 år
Fallföretaget	Operativ produktägare	3 år



Figur 2 – Teamstruktur. Källa: Egen.

I figur 2 kan en ungefärlig teamstruktur ses med informanternas befattningar angivna. Av informanterna är det främst Scrum Master, Systemarkitekten och Systemutvecklarna som utgör det faktiska DevOps teamet. Konsultchefen, Product Owner och Vikarierande Product Owner är kopplade till teamet och har varit delaktiga i implementeringsprocessen i någon grad. Vikarierande Product Owner har en stödjande roll för Product Owner.

2.4 Studiens kvalitet

2.4.1 Reliabilitet

Reliabilitet innebär att om andra forskare använder samma metod ska samma resultat kunna uppnås (R. K. Yin, 2018). Det är inte lika enkelt att bedöma reliabiliteten i en kvalitativ studie som i en kvantitativ studie. I en kvalitativ studie används ofta metoder såsom intervjuer som gör att forskaren måste tolka resultaten. Det kan även vara svårt att uppnå samma sociala miljö vilket gör det svårt att upprepa studien och få samma resultat (Patel & Davidson, 2011). Genom att dokumentera och utförligt förklara hur studien har utförts kan reliabiliteten öka (R. K. Yin, 2018).

I denna studie har intervjuerna spelats in för att sedan transkriberats, detta kan anses öka reliabiliteten då intervjusvar har kunnat kontrollerats och risken för bedömningsfel har minskat. Dessutom har båda författarna närvarat och tillsammans sammanställt och tolkat intervjusvaren, också för att minska bedömningsfel. Informanter med olika befattningar har noga valts ut för att få det bredaste möjliga perspektivet på teamets upplevelser. Semistrukturerade intervjuer genomfördes med förutbestämda frågor och en hög grad av standardisering. Eftersom frågorna hade utformats på förhand och ställdes i samma ordning ökar detta reliabiliteten. Att utföra semistrukturerade intervjuer har gjort att en del följdfrågor har ställts under intervjuerna. Eftersom följdfrågorna inte var förutbestämda kan det anses minska möjligheten till att samma resultat kan uppnås.

2.4.2 Validitet

En studies validitet avser hur väl vad man tänkt undersöka överensstämmer med det man faktiskt undersöker (Patel & Davidson, 2011). Om reliabilitet saknas kan detsamma gälla för validitet men hög reliabilitet däremot innebär inte nödvändigtvis hög validitet. Även om upprepning av undersökningens metod skulle ge samma resultat behöver det inte betyda att resultatet mäter det som var tänkt. Att avgöra graden av validitet kan vara komplicerat (Bell & Waters, 2014). Det som undersöks kan ibland vara abstrakt och därav svårt att bedöma, särskilt då det handlar om människors upplevelser eller liknande (Patel & Davidson, 2011). Det kan vara svårt att mäta validitet men man bör ändå försöka överväga det kritiskt (Bell & Waters, 2014).

Validitet berör hela forskningsprocessen i en kvalitativ studie. Det finns inga fasta regler eller procedurer för att bekräfta en kvalitativ studies validitet då varje forskningsprocess är unik. För att styrka validiteten bör dock empiri som insamlas vara tillräcklig för att kunna utgöra en grund för trovärdiga tolkningar som tillför kunskap. Forskningsprocessen och motiverade tolkningar bör också kunna kommuniceras. Under transkription av intervjuer kan en viss påverkan ske då exempelvis kroppsspråk, mimik och betoningar kan förvinna. Det är därför viktigt att reflektera över hur empirin hanteras (Patel & Davidson, 2011).

Då denna studie utförts och uppnått resultat i enlighet med dess syfte kan det anses finnas hög validitet. Detta arbete samlade teoretisk data från en stor mängd källor för att uppnå en mer tillförlitlig kunskapsgrund som stärker validiteten. Insamlad teori utgör också sedan grunden

för tolkningar och slutsatser som diskuterats samt motiverats. Förloppet för forskningsprocessen är också tydligt presenterad i metodavsnittet, vilket ger andra en inblick i arbetet för egen bekräftelse av arbetets validitet. Av de åtta intervjuer som genomfördes användes materialet från sju av dessa i rapporten. Den åttonde informanten var inte kopplad till samma team som övriga sju och valdes därför bort för svaren ansågs ligga utanför arbetets område. Detta ökar arbetets validitet. Dessutom testades intervjufrågorna på förhand för att säkerställa att svaren var relevanta vilket ytterligare ökar arbetets validitet.

Bedömningen av validiteten försvåras eftersom detta är en kvalitativ studie och empirin övervägande är baserade på människors upplevelser. Validiteten kan ha påverkats av att fallföretaget i samband med implementeringen av DevOps även gjort andra förändringar, såsom att börja följa ramverket SAFe. Det innebär att vissa identifierade effekter kan ha andra orsaker än just DevOps. Under intervjuer har frågor och följdfrågor främst specificerats mot DevOps för att insamlad empiri bättre ska följa studiens syfte. Vid hanteringen av empiri från intervjuerna har det också funnits medvetenhet om viss påverkan. Vissa former av uttryck, såsom bland annat kroppsspråk och mimik, har gått förlorade då intervjuerna genomfördes virtuellt med ljudupptagning. Vid transkriberingen kan även betoningar och uttryck missas. Det innebär att arbetets validitet minskar men beaktanden har gjorts för att minska denna påverkan. Då flera intervjuer genomförts där båda författarna deltagit och tillgång till ljudupptagningarna gjort det möjligt att lyssna på intervjusvaren fler gånger. Därför kan ändå en ansevärt rättvisande uppfattning tänkas uppnås. För att stärka validiteten fick även företaget ta del av empirin för att bekräfta att det som sammanstälts var riktigt innan vidare analys och slutsatser gjordes.

2.4.3 Generaliserbarhet

Generaliseringar kommer vanligen från flera undersökningar som replikerats i olika förutsättningar (R. K. Yin, 2018). När alla situationer med alla möjliga förutsättningar inte kan studeras väljs avgränsade fall. Vid bedömning av en undersöknings generaliserbarhet övervägs hur väl resultatet kan stämma även i andra fall. Bedömningen kan inkludera reflektioner kring hur urvalet för situationer eller individer som studeras gjorts. Om urvalet sker slumpmässigt kan generaliseringar göras utifrån det stickprov som utgör en mindre representation av det totala intresseområdet (Patel & Davidson, 2011). Det kan anses omöjligt att uppnå generaliserbarhet i en enskild fallstudie. Fallstudier kan dock förklaras som en empirisk syn på teoretiska koncept och principer. På så sätt kan fallstudier användas för att utöka den befintliga teorin och bidra till dess teoretiska generaliserbarhet (R. K. Yin, 2018)

Detta arbetet har bedrivits som en fallstudie, vilket påverkar möjligheterna till generaliserbarhet. För att få ett bredare perspektiv på fallstudien har urvalet av informanter inkluderat olika befattningar. Att olika perspektiv tagits med kan vara positivt men eftersom urvalet inte var slumpmässigt är det svårare att göra generaliseringar. Eftersom studien dessutom endast berör ett enskilt fall och inte har replikerats i andra förutsättningar kan generaliserbarhet inte förväntas. Arbetet kan dock anses påverka generaliserbarheten hos den teori som använts då den jämförts med empiri från fallstudien.

2.5 Etiska perspektiv

Enligt det humanistisk-samhällsvetenskapliga forskningsrådet finns det fyra huvudkrav att uppfylla för att upprätthålla individskyddkravet under forskning. De fyra huvudkraven innehåller regler för hur de ska uppfyllas och kan konkretiseras som informationskravet, samtyckeskravet, konfidentialitetskravet samt nyttjandekravet (Vetenskapsrådet, 2002).

För att uppfylla informationskravet krävs det att forskare delger uppgiftslämnare och deltagare om vilka villkor som gäller för deras deltagande i projektet. Därutöver måste de även informeras om att deltagandet är frivilligt samt att de innehar rätten att avbryta sitt deltagande när som helst. Deltagare i projektet ska upplysas om alla faktorer som skulle kunna ha en påverkan på deras benägenhet att delta i projektet (Vetenskapsrådet, 2002).

Under samtyckeskravet finns tre regler som skall uppfyllas, dessa förklaras av HSFR och den första regeln innebär att samtycke skall inhämtas från samtliga deltagare. Den andra regeln innebär att medverkande i studien innehar rätten att själva bestämma om de vill delta, på vilka villkor deltagande skall ske samt att medverkande kan avbryta deltagandet utan påföljder. Den tredje regeln under samtyckeskravet innebär att om deltagande väljer att avbryta sin medverkan i studien får denne inte utsättas för påtryckning eller påverkan (Vetenskapsrådet, 2002).

Konfidentialitetskravet innehåller två regler. Den första regeln innebär undertecknandet av tystnadsplikt av all personal om studien behandlar etiskt känsliga uppgifter om identifierbara personer. Den andra regeln under konfidentialitetskravet innebär att enskilda personer som ingår i studien inte ska kunna identifieras av utomstående. Anteckningar, avrapportering och lagring av uppgifter ska ej kunna nås av utomstående (Vetenskapsrådet, 2002).

Kravet om nyttjandet innebär att allt forskningsmaterial som har insamlats under studien inte får användas kommersiellt eller lånas ut i icke-vetenskapliga syften. Därutöver får inte uppgifter insamlade om personer för studier eller forskning användas för åtgärder eller beslut som direkt påverkar den individen (Vetenskapsrådet, 2002).

Författarna av denna studien hade de fyra etiska kraven i åtanke under arbetets gång. För att uppfylla de fyra huvudkraven mottog medverkande personal hos fallföretaget ett missivbrev som innehåller ovanstående information innan deltagande i studien. Missivbrevet kan ses i bilaga 1. Dessutom upprepades denna information innan varje intervju.

2.6 Hållbarhetsperspektiv

Det finns tre grundbegrepp inom hållbar utveckling, vilka är ekonomisk, ekologisk och social hållbarhet. Att uppnå ekonomisk hållbarhet innebär att hushålla med de resurser som finns tillgängliga. Ekologisk hållbarhet syftar till att minska påverkan på naturen och människans hälsa. Social hållbarhet handlar om människors lika värde, att alla människans grundbehov bör uppfyllas sådan att individen känner sig trygg och delaktig (Bergman & Klefsjö, 2012).

Själva arbetet i studien har bedrivits på ett hållbart sätt, där alla tre hållbarhetsperspektiv har beaktats. Genom att använda tillgänglig litteratur eller digital litteratur har den ekonomiska hållbarheten beaktats. Inga utskrifter av litteratur har gjorts vilket också främjar den ekologiska hållbarheten. Ytterligare hänsyn har tagits genom att inga resor har företagits, och arbetet har skett digitalt. Avslutningsvis har det tagits hänsyn till den sociala hållbarheten i arbetet genom att värna om varandras hälsa samt haft en god arbetsrelation med ömsesidig respekt.

I denna studie har alla tre grundbegrepp inom hållbar utveckling beaktats. Studiens slutsats och de rekommendationer som har tagits fram kan anses vara utan negativ påverkan på den ekologiska hållbarheten. Däremot kan de ha en påverkan på den ekonomiska hållbarheten i början av implementeringen genom tillsättning av extra resurser. Långsiktigt kan dock rekommendationerna tänkas främja ekonomisk hållbarhet. Slutligen kan rekommendationerna anses ha en positiv effekt på den sociala hållbarheten genom att minska den negativa påverkan implementeringen kan ha på ett team.

3. Teori

I följande avsnitt presenteras teori som behandlar bakgrunden till arbetets forskningsfrågor. Även grunden för implementering och påverkan på team samt viktiga implementeringsfaktorer som påverkar båda dessa områden tas upp. Slutligen presenteras en sammanfattning av teoriavsnittet.

3.1 Teoretisk bakgrund

3.1.1 Projekt

Ordet projekt kan ha olika betydelser och används såväl i människors privatliv som i organisationer. En passande beskrivning enligt Hallin och Karrbom Gustavsson (2019) kan vara att projekt är en tillfällig aktivitet som genomförs med hjälp av resurser för att uppnå ett mål. Projekt kan vara målstyrande och handla om specifika förutbestämda mål eller målsökande där målen utvecklas under projektets gång. Projektledning handlar om hur projekt bör ledas och utformas. Någon form av projektarbete fanns redan långt tillbaka i tiden men teorin kring projektledning började ta form under 30-talet. Då utvecklades de första modellerna inom försvarsindustrin. De första modellerna var så kallade stage-gate modeller. De traditionella stage-gate-modellerna, eller vattenfallsmodeller, delar in ett projekts syfte i delmål och innefattar tydlig planering över vad som ska göras samt med vilka resurser (Hallin & Karrbom Gustavsson, 2019). Inom projektledning finns numera olika tillvägagångssätt med både mer traditionella metoder och agila metoder (Gemino et al., 2021).

Iterativ och inkrementell mjukvaruutveckling är en grund för de agila metoderna. Framväxten av iterativa och inkrementella modeller har pågått under lång tid och mycket skedde redan på 70- samt 80-talet (Larman & Basili, 2003). De traditionella modellerna innebar att IT-projekt hanterades med förbestämda processer på samma sätt som inom exempelvis biltillverkningen. Inom mjukvaruutveckling fanns dock behov av metoder för mer komplexa system, vilket ledde till utvecklingen av iterativa och inkrementella modeller samt med tiden även agila metoder (Tonnquist, 2018). Under 90-talet kom nya metoder såsom Scrum, Extreme Programming (XP) och Dynamic Systems Development Method (DSDM). De iterativa och agila metoderna utvecklades som ett alternativ till den sekventiella planeringen som upplevdes överdrivet dokumentationskrävande och komplicerad (Hallin & Karrbom Gustavsson, 2019).

3.1.2 Agila manifestet

Bakom det agila arbetssättet finns tolv principer att följa. Dessa tolv principer presenterades först i ett manifest som formulerades av amerikanska mjukvaruutvecklare år 2001 (Tonnquist, 2018). Med de tolv principer som grund handlar agila metoder övergripande om kultur och människor i fokus samt att leverera bra produkter till kunder. Dokumentation och planering används inom agilt arbete men i måttlig utsträckning samt med den turbulenta omgivningen i åtanke (Highsmith, 2001, History: The Agile Manifesto). Det agila manifestets tolv principer kan ses som riktlinjer med utrymme för tolkning och anpassning utifrån organisationens egna förutsättningar samt mål (Tonnquist, 2018). Det agila manifestets tolv principer kan ses i listan nedan (Beck et al., 2001):

1. Högsta prioritet är att, genom tidig och kontinuerlig leverans av värdefull mjukvara, uppnå kundnöjdhet.
2. Förändrade krav, även sent i utvecklingen, välkomnas. Förändring används för att uppnå konkurrensfördelar i agila processer.
3. Fungerande mjukvara levereras ofta.
4. Dagligt samarbete mellan *operations* och utveckling är ett måste.
5. Inblandade individer bör vara motiverade, få det stöd som behövs och ges tillit till att få arbetet gjort.
6. Att kommunicera ansikte mot ansikte innebär effektivare utbyte av information.
7. Det främsta måttet på framgång är fungerande mjukvara.
8. Hållbar utveckling främjas.
9. Mer agilitet kan uppnås genom kontinuerlig uppmärksamhet på bra design och teknisk excellens.
10. Enkelhet är väsentligt, överflödigt arbete bör elimineras.
11. Självstyrda team kan uppnå de bästa arkitekturerna, kraven och designerna.
12. Teamet ska regelbundet reflektera över hur effektivitet kan öka och utifrån det anpassa beteenden.

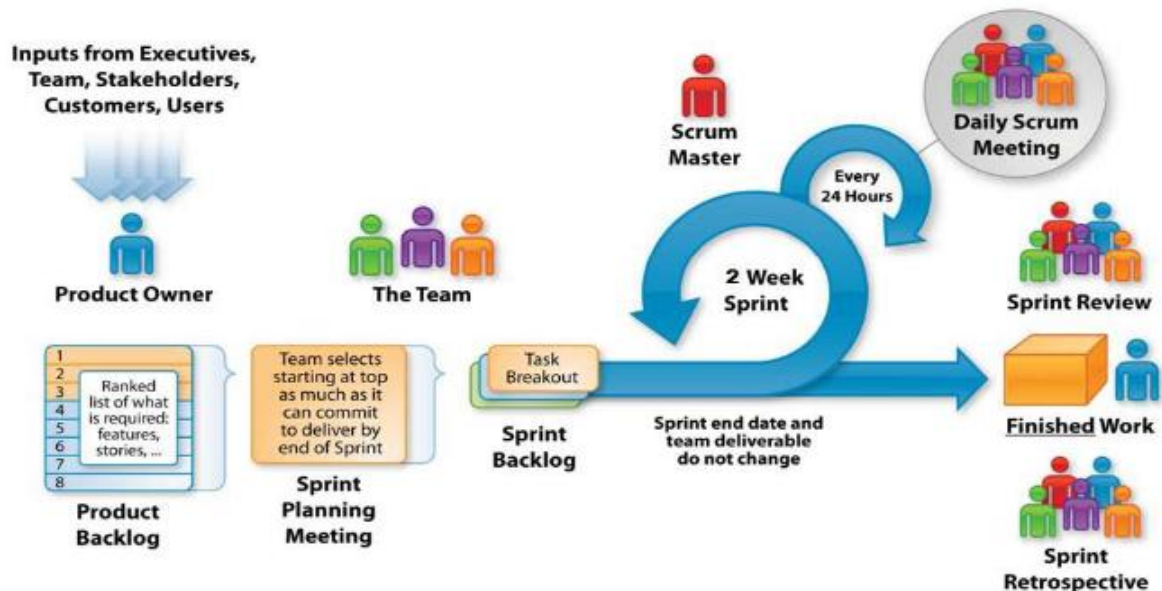
Övergripande områden inom agilt arbete inkluderar visualisering, självstyrande team, dagliga möten och cykliska processer. Visualisering genom exempelvis Kanban-tavlor kan förbättra arbetares fokus då färdiga uppgifters slut markeras och det som ska göras blir tydligare. Självstyrande team kan uppnå större effektivitet genom det ökade ansvar decentraliseringen medför. Det krävs dock relevant kompetens inom teamet och gemensamma mål. Med korta, dagliga möten kan hela teamet få en uppfattning av vad vardera medlem har gjort, ska göra och eventuellt behöver hjälp med. En cyklisk process möjliggör arbete i korta etapper med kontinuerlig feedback och frekventa leveranser. Det innebär att behövda ändringar kan identifieras under arbetets gång och förändringar kan snabbt bemötas (Tonnquist, 2018).

3.1.3 Agila metoder

Agila metoder innebär en större flexibilitet. Innehållet i ett agilt projekt utformas under tiden projektet fortgår, vilket skiljer sig från icke-agila modeller där faserna sedan innan är bestämda (Hallin & Karrbom Gustavsson, 2019). Agila metoder kan förbättra kommunikationen med intressenter och inom arbetslag (Gemino et al., 2021). Det bidrar även till snabbare feedback och snabbare programvaruutveckling (Galup et al., 2020). Med flexibilitet och lyhördhet kan agila metoder bidra till en förbättrad förmågan att snabbt möta efterfrågan från kunder (Christopher, 2011).

Det finns flera olika agila metoder som exempelvis Scrum, XP och DSDM (Hallin & Karrbom Gustavsson, 2019). Scrum är idag en av de vanligaste agila metoderna och handlar om att uppgifter inom projektet delas in i sprinter. Varje sprint är av en bestämd längd och bör skapa värde (Tonnquist, 2018). Scrum metoden utgår från att teamet utformar en *product backlog* där det som ska göras står listat. Arbetet i sprinterna följer sedan *sprint logs* med prioriterade uppgifter från *product backlog*. Sprinter inleds med planeringsmöten och dagliga möten som

kallas *daily scrums* utförs genomgående. Avslutningsvis utförs *sprint reviews* där det som gjort granskas och presenteras för projektets samtliga intressenter (Hallin & Karrbom Gustavsson, 2019). I Scrum team finns utvecklare som skapar, Product Owner som maximerar värdet av det utvecklarna skapat och ansvarar för *product backlog* samt Scrum Master som ansvarar för teamets effektivitet genom efterlevande av Scrum metoden (Schwaber & Sutherland, 2020).

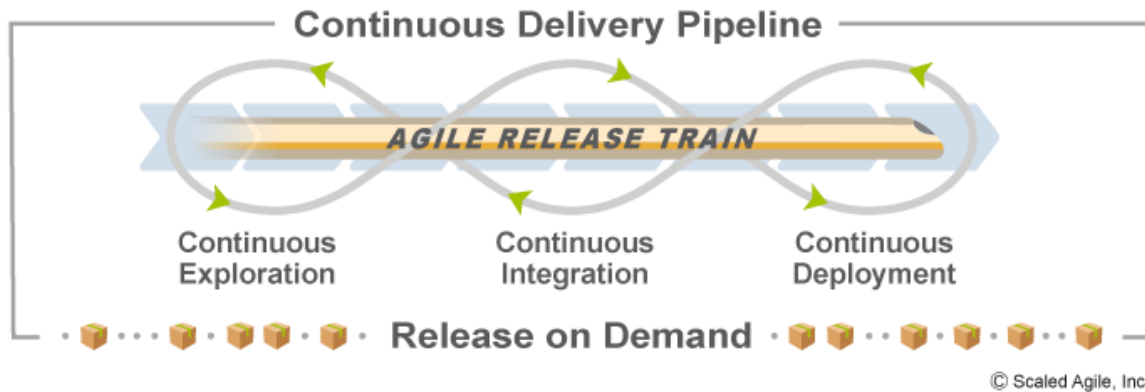


Figur 3 - Scrum baserad agil standardmodell. Källa: (Neve et al., 2017)

Agila modeller kan se olika ut i olika organisationer eller team men en Scrum baserad standardmodell kan ses i figur 3. Ofta är det kunden som utgör Product Ownern och kraven benämns som *Product Backlogs*. *Sprint* är cykeln för en produktleverans och uppgifter som finns kvar från tidigare *Sprint* kallas *Sprint Backlogs*. På *Daily Scrum Meetings* diskuteras daglig status och strategi och *Retrospectives* används för att kontinuerligt se tillbaka samt upptäcka förbättringsmöjligheter i processen (Neve et al., 2017).

3.1.4 Scaled Agile Framework

Det krävs en del överväganden då agila metoder ska tillämpas till många team som utgör stora projekt. För att bibehålla Lean funktioner, samarbeten, kommunikation och självstyrande i teamen har en del olika ramverk utvecklats för användningen av agila metoder på en större skala. Ett av dessa ramverk är SAFe, vilket står för *Scaled Agile Framework* (Faizi et al., 2019). Att följa SAFe kan bland annat leda till ökad produktivitet, bättre kvalitet, mer motiverad och engagerad personal samt att produkter och tjänster snabbare når marknaden (Neve et al., 2017). Roller från Scrum metoden, såsom Product Owner och Scrum Master, kan återfinnas i SAFe men även andra roller definieras, däribland rollen som Systemarkitekt (Scaled Agile Inc, n.d.). Systemarkitekten ska säkerställa att det som utvecklas stämmer överens med det avsedda syftet, vilket kan göras med hjälp av ett *Agile Release Train (ART)* (Scaled Agile Inc, 2021c).



Figur 4 - SAFe Continuous Delivery Pipeline. Källa: (Scaled Agile Inc, 2021b)

Agile Release Trains är en sammankoppling av agila team som inkrementellt kan utveckla, leverera och driva lösningar i en värdeström (Scaled Agile Inc, 2021a). Figur 4 illustrerar att varje ART ingår i en *Continuous Delivery Pipeline* där *Continuous Exploration*, *Continuous Integration* och *Continuous Deployment* tillsammans stödjer leveransen av nya funktioner efter efterfrågan från marknaden. *Continuous Exploration* handlar om att förstå marknadens behov och vad som krävs för att möta dessa. Med *Continuous Integration* kan det som ska göras från Product Backlog kontinuerligt implementeras. *Continuous Deployment* distribuerar kontinuerligt förändringarna som utvecklas (Scaled Agile Inc, 2021b). En del som kan identifieras inom SAFe ramverket är DevOps (Scaled Agile Inc, n.d.).

3.2 DevOps

3.2.1 DevOps framväxt

Programmering och mjukvaruutveckling började växa fram under 50- och 60-talet. Med tiden blev systemen alltmer komplexa och ett behov av specialiserade kunskaper skapades. Fler tydliga roller upprättades, som exempelvis systemadministratörer samt mjukvaruutvecklare, och silostrukturer med specialiserade grupper tog form. Dessa silostrukturer och rollfördelningar skapade en del problem. Synpunkter från kunder som använde programvaran fick svårare att nå de utvecklare som utformade den. När webbapplikationer blev lättare att skapa under 90-talet ökade dessutom kraven på utvecklare att vara snabba och flexibla för att kunna förbli konkurrenskraftiga (Davis & Daniels, 2016).

Över tid har flera olika metoder tillämpats för att förenkla mjukvaruutveckling och *operations*. Fokus i många av dessa har dock separerat utvecklingsprocesser från *operations* funktioner, vilket gör att utvecklings- och *operations*personal inte delar samma mål (Davis & Daniels, 2016). Det är vanligt att olika funktioner, såsom utveckling och *operations*, separeras i tron om att ökad specialisering och standardisering ska innebära mer kostnadseffektiva, säkra samt lätthanterliga processer. Dock kan det istället leda till problem som exempelvis konkurrerande mål, konflikter och otydlig ansvarsfördelning mellan funktioner som hellre skyller problem på varandra än samarbetar för att lösa dem (Comstedt, 2019). DevOps uppkom hos personer som kände frustration över separerande silostrukturer. Istället för silostrukturer och konkurrens

mellan olika funktioner, förespråkar DevOps samverkan och samarbete mellan olika roller (Davis & Daniels, 2016).

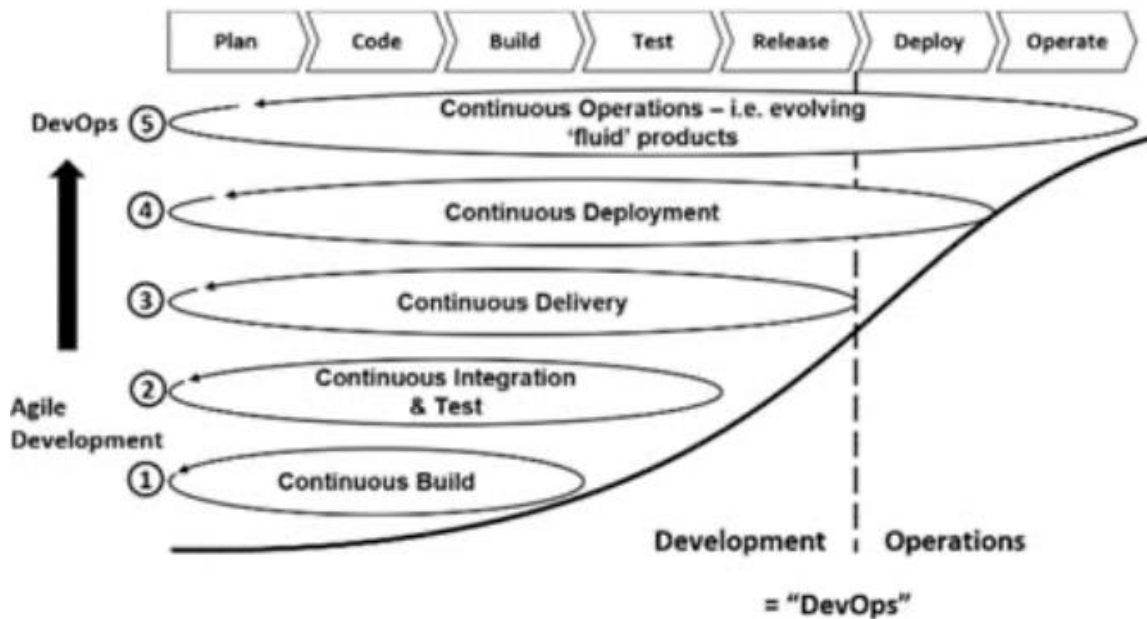
DevOps har utvecklats som en respons till den alltmer uppmärksammade distansen mellan IT-utvecklare och IT-operatörer (Diel et al., 2016) och det gap som ofta finns gällande kommunikation, samverkan, kultur samt samarbete (Diel et al., 2016). Ursprunget för DevOps har också framkommit ur behovet att snabbt nå ut till marknaden med innovativa produkter och funktioner, vilket särskilt gäller inom mjukvaru- och IT-industrin (Ebert et al., 2016).

3.2.2 DevOps innehåll

DevOps cirkulerar kring kultur och teknologi (Hosono & Shimomura, 2012) och är en kombination av agila principer samt Lean (Maroukian & Gulliver, 2020). Det har länge funnits en länk mellan agila arbetssätt och Lean inom IT fältet (Galup et al., 2020) och DevOps har utvecklat detta för att uppnå moderna metoder med det bästa ur båda (Comstedt, 2019). DevOps handlar om att använda feedback och analyser för att nå kvalitet. Två framstående delar inom DevOps modellen är en *Continuous Delivery Pipeline* och *Continuous Improvement* (Qumer Gill et al., 2018). Tre principer som ligger i grunden för DevOps är flöde, feedback och kultur. Den första handlar om att en organisations flöde ska ses på i helhet från början till slut och inte separerat silo för silo. Alla som arbetar i flödet ska även enligt DevOps vara delaktiga förbi något enstaka delmål ända till slutmålet. Den andra principen framhäver vikten av feedback som fångar de problem som kan finnas i flödet för att dessa ska kunna bemötas snabbt. Kultur, den tredje principen, ska i DevOps inbegripa experimentering för förbättringar, problemlösning och lärande. Med kontinuerlig övning, träning och repetition kan arbetet optimeras (Comstedt, 2019).

3.2.3 Hur DevOps utvidgar agil arbetsmetodik

Trots att agila metoder uppmuntrar samarbete mellan alla intressenter, är det vanligt inom IT industrin att exempelvis *operations* inte inkluderas i agila projekt. DevOps har utvecklats som en respons för att motverka detta uteslutande och möjliggöra agila team utan detta kommunikationsgap (Diel et al., 2016). Inom agilt arbete kan det ofta på längre sikt finnas en viss problematik kring den totala leveransen. Även om agilt arbete kan bidra till snabbare leverans och högre kvalitet, är avsikten med DevOps att säkerställa den totala leveransen högsta möjliga kvalitet och snabbaste möjliga genomförande hela vägen från början till slut (Comstedt, 2019). DevOps vidgar det agila arbetssättet samt skapar en bro mellan *operations* och utveckling på ett företag (Wiedemann, 2018).



Figur 5 - Agile to DevOps transformation. Källa: (Gupta et al., 2019)

Utvecklingsarbete som följer agila metoder fokuserar ofta på *Continuous Integration* och test, vilket kan ses i figur 5. Dock bedrivs andra aktiviteter som exempelvis produktlansering, service och *operations* ofta av separata team, vilket i figur 5 står som *Release*, *Deploy* och *Operate*. DevOps kan därför vara ett sätt att utöka det agila arbetet, förbättra feedback och sammankoppla olika funktioner. I DevOps team samarbetar utvecklare och operatörer från utvecklingsstadiet, genom test och konstruktion, till lansering. Som kan ses i figur 5 vidgar DevOps det agila arbetssättet och ger ett mer sammankopplat arbete mellan fler funktioner inom både utveckling och *operations*. Utvecklare och operatörer sammankopplas genom att utveckling, leverans och *operations* integreras (Gupta et al., 2019). På så sätt möjliggör DevOps snabb och flexibel utveckling samt tillämpning av affärsprocesser (Ebert et al., 2016), nya lösningar och metoder (Sill, 2014).

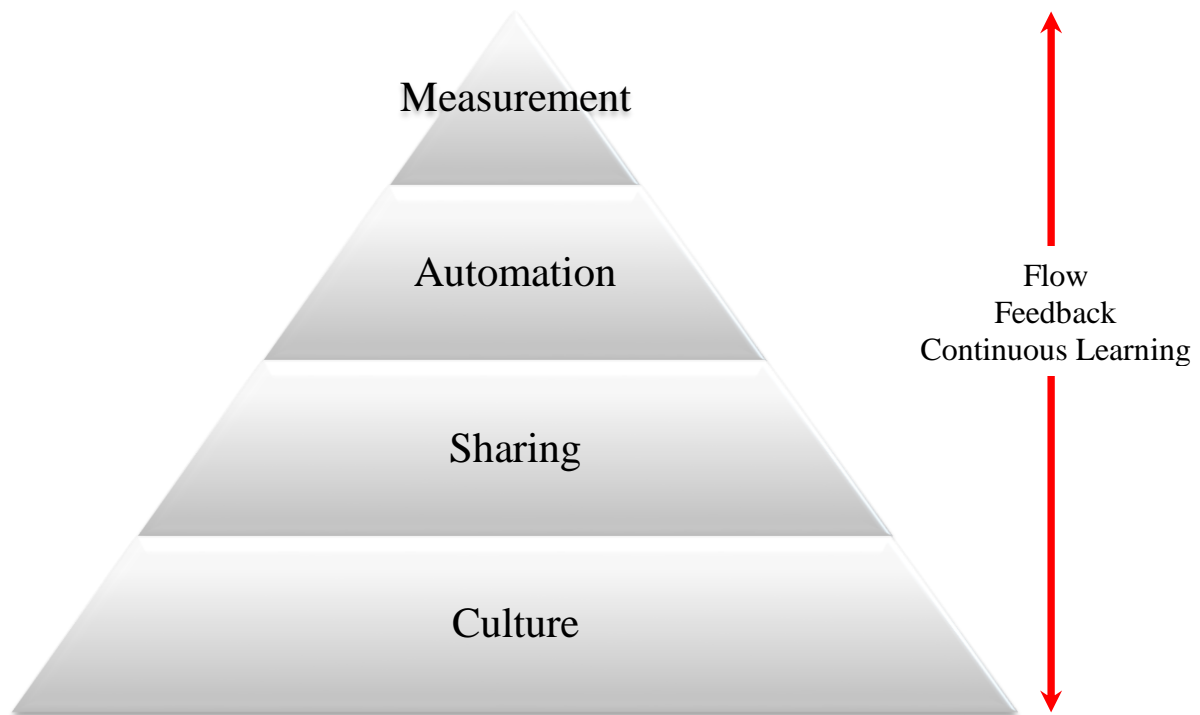
3.3 Implementering

Det är ofta möjligt att implementera DevOps men hur detta genomförs bör anpassas efter förutsättningarna som råder. Arkitekturen, verktygen och kulturen måste matcha företagets egna tillvägagångssätt (Ebert et al., 2016). Eftersom det inte finns någon tydlig allmän definition av DevOps, kan dess implementering genomföras på olika sätt. Företag får därav avgöra hur implementeringen ska gå till och det är viktigt att företagen själva tolkar en egen definition av DevOps på förhand. För att anställda ska förstå varför DevOps införs, vilka fördelar det medför och hur det ska ske, måste ledningen kommunicera företagets uppfattning av vad DevOps innebär (Nybom et al., 2016).

3.3.1 Tillvägagångssätt

Implementeringen av DevOps kan utformas på olika sätt och några exempel kan vara att tillämpa tillfälliga lösningar eller att långsiktigt övergå till tvärfunktionella team. Även

hybridlösningar är möjliga, där tjänster utförs med hjälp av både traditionellt organiserade metoder och DevOps teams (Wiedemann, 2018). Tre möjliga exempel på tillvägagångssätt för att sammankoppla utveckling och *operations* är att blanda ansvar, blanda personal eller bilda överbyggande team. Att blanda ansvar innebär att samtliga ingenjörer tilldelas ansvar för både utveckling och *operations*. Om personal istället blandas förblir rollerna för utveckling och *operations* uppdelade men kommunikationen samt samarbetet förbättras. Ett separat överbyggande DevOps team däremot bildar en förbindelse mellan utveckling och *operations* (Nybom et al., 2016).



Figur 6 - DevOps Pyramid. Källa: (Muñoz & Rodríguez, 2021)

Implementeringen av DevOps kan börja med att den önskade samarbetskulturen och kraven för att nå dit fastställs. På så sätt kan en plan utformas för en gradvis förändring som smidigt integrerar utvecklare och operatörer (Qumer Gill et al., 2018). Ett sätt att betrakta DevOps implementering är att dela upp den i fyra nivåer bestående av kultur, delning, automation och mätning enligt figur 6. För att lyckas behöver dessa fyra behov uppfyllas med stöd av *Flow*, feedback och *Continuous Learning*. Kultur, som utgör *Culture* i botten av pyramiden i figur 6, är en grundläggande del i DevOps som bland annat kan leda till bättre informationsdelning, delat ansvar och mer idéspridning. *Sharing*, som kan översättas till delning, kan sedan utvecklas för bättre förståelse av arbetet, mål och prioriteringar. Med *automation* kan sedan teknisk kunskap delas, test- samt installationstider minskas och data övervakas. *Measurement*, som kan översättas till mätning, behövs slutligen också för att kvaliteten ska öka och omarbetning minska. Triangeln i figur 6 ger också en viss uppfattning om de fyra behovens omfattning under implementeringen (Muñoz & Rodríguez, 2021).

3.4 Implementeringsfaktorer

Kommunikation, automation, ansvar, förändringsvillighet och samarbete är faktorer som behöver beaktas. Dessa faktorer påverkar både implementeringsprocessen och teamets samhörighet samt arbetssätt.

3.4.1 Kommunikationsfaktorer

Om implementeringen ska kunna genomföras säkert och effektivt behöver det finnas en tydlig förståelse för vad DevOps innebär (Qumer Gill et al., 2018). Vad konceptet betyder för organisationen bör därför tydligt kommuniceras för att ge stöd åt implementeringen (Nybom et al., 2016). Implementeringen av DevOps handlar mycket om den inblandade personalen och bör därför centreras kring dessa snarare än att bara ses som införandet av nya tekniker. Implementeringen medför kulturförändringar och därför är det viktigt att teammedlemmar informeras om hur dessa kommer påverka det dagliga arbetet (Indriasari et al., 2020). Även fortsatt träning och coachning kan bidra till förståelse, hantering av problem och en förenklad implementeringsprocess (Qumer Gill et al., 2018).

3.4.2 Automationsfaktorer

För att uppnå snabba processer och samtidigt leverera hög kvalitet krävs mycket automatisering (Ebert et al., 2016). Därav kan viss automation i processer vara nödvändigt vid införande av DevOps (Nybom et al., 2016). Om en del arbete kan automatiseras istället för att allt utförs manuellt kan arbetet förenklas (Nybom et al., 2016) och ledtider kan minskas (Ebert et al., 2016). Risken för misstag som orsakas av exempelvis den mänskliga faktorn minskar även med hjälp av automation (Qumer Gill et al., 2018) och informationsdelning kan förbättras (Nybom et al., 2016). Om till exempel delar av testning, spridning och drift automatiseras kan bättre repeterbarhet, förutsägbarhet och snabbhet uppnås (Gupta et al., 2019). Det finns olika verktyg för automatisering inom DevOps som till exempel innefattar konstruktion, *Continuous Integration*, konfigurationshantering, loggning och övervakning. Vilka verktyg som tillämpas bör dock stämma överens med företagets förutsättningar (Ebert et al., 2016).

3.4.3 Ansvarsfaktorer

DevOps innebär behov av decentralisering, kommunikation och kunskapsdelning. Ledningspositioner kan reduceras för att nå den platta hierarki som behövs för att bli mer agil (Wiedemann, 2018). Decentraliserade IT funktioner möjliggör bra självbestämmande (Indriasari et al., 2020) och inom DevOps team är mycket arbete självständigt med egna beslut och ansvarstagande (Wiedemann, 2018). Starkt stöd från ledningen kan dock fortfarande behövas för att bibehålla viss stabilitet i arbetet (Nybom et al., 2016). Ansvarstagande som delas av alla och tydligt integrerade roller bör förespråkas av ledningen (Qumer Gill et al., 2018).

3.4.4 Förändringsvillighet

Anställdas attityd och inställning är avgörande när team ska övergå till DevOps (Wiedemann, 2018). Det kan vara väldigt svårt att forma en mentalitet som är öppen för förändring (Christopher, 2011) och förändring bemöts lätt med motvillighet (Nybom et al., 2016). Genom

att låta övergången till DevOps ske inkrementellt, lite i taget, kan förändringen gå enklare. När några förändringar ger märkbara förbättringar kan teamet motiveras och införandet av ytterligare steg mot automation och DevOps förenklas. Genom att fokusera på de mest behövda delarna först och sedan inkrementellt förbättra ytterligare kan teamet fokusera på affärsvärde och korta frigöringscykler (Gupta et al., 2019)

3.4.5 Samarbetsfaktorer

En åtgärd för att förbättra samarbetet i teamet och mellan utvecklare och operatörer kan vara att arrangera så alla sitter nära varandra. Att sitta tillsammans, då möjlighet finns, kan öka samverkan, kunskap, stöd och motivation inom teamet (Gupta et al., 2019). Om teammedlemmarna exempelvis är utspridda globalt kan kvartalsmöten för fysiska träffar vara ett komplement till den dagliga kontakt som måste ske med fysisk distans. Kompletterande fysiska kvartalsmöten kan vara ett sätt att minska problem som distans kan medföra. Vid dessa tillfällen kan feedback delges och medlemmarna får en möjlighet att lära sig om kulturen på andra platser (Diel et al., 2016). Oavsett fysisk närhet eller distans mellan teammedlemmar måste förutsättningar ges för kommunikation samt kunskapsdelning (Wiedemann, 2018). Insatser för att förbättra relationerna mellan teammedlemmarna kan också bidra till bättre kommunikation (Nybom et al., 2016).

3.5 Påverkan på team

Att övergå från silostrukturer som separerar *operations*- och utvecklingsfunktioner kan inbegripa förbättringar av arbetsprocesser, kultur och teknologi (Nybom et al., 2016). När DevOps implementeras förändras ofta företags interna IT funktioner med nya strukturer och processer för att passa det mer agila arbetssättet (Wiedemann, 2018). Ett kulturskifte sker också vid införandet av DevOps. Kunskaperna hos utvecklarna behöver vidgas utöver kodning och kompletteras med administration samt testning. Det är också väldigt viktigt med kommunikation, samarbete (Ebert et al., 2016) och gemensamma mål (Comstedt, 2019). Om implementering av DevOps ska vara framgångsrik måste ledningen stödja utformningen av de gemensamma mål samt tekniker som *operations*- och utvecklingsfunktionerna behöver ha (Nybom et al., 2016).

3.5.1 Fördelar

DevOps kan leda till mer agilt arbete samt förbättrad innovation, lyhördhet, kvalitet och flexibilitet (Wiedemann, 2018). Några fler fördelar är färre kravförändringar, kvalitetssäkring och snabba utgivningscykler (Ebert et al., 2016). Genom att utveckla samarbete och automation kan implementeringen av DevOps stärka förmågan att introducera innovationer (Indriasari et al., 2020) och förbättra organisationens effektivitet (Nybom et al., 2016). Modellen länkar utveckling samt *operations* (Gupta et al., 2019) och möjliggör snabbare övergång från formella idéer till praktisk implementering (Sill, 2014).

Genom att utveckling och *operations* sammankopplas kan företag oftare erbjuda bättre service (Nybom et al., 2016). Den ökade samverkan kan således bättre betjäna kunder (Diel et al., 2016)

och högt kundvärde kan skapas. Automationen ger också mer affärsvärde då ökad förutsägbarhet och pålitlighet tillåter teamet att arbeta snabbare (Gupta et al., 2019). Det ökade samarbetet kan också bidra till mer förståelse mellan funktioner. En utvecklare som exempelvis förstår *operations* behov bättre kan också utveckla bättre lösningar till dessa (Nybom et al., 2016).

3.5.2 Nackdelar

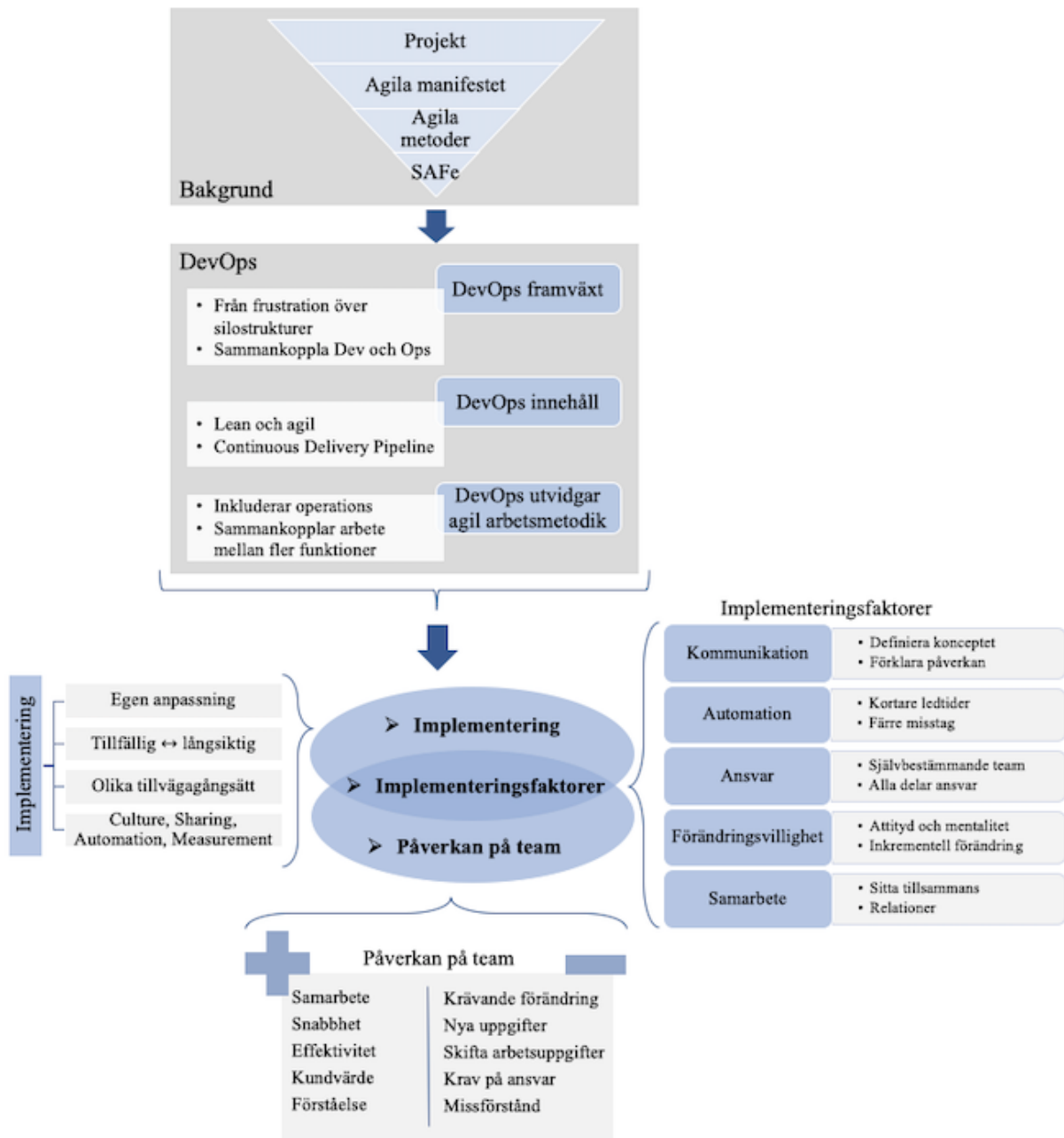
För att uppnå fördelarna med DevOps måste en del gamla arbetsätt, tankemönster och vanor brytas. Det kan finnas en viss motvillighet mot att gå ifrån traditionella silostrukturer (Qumer Gill et al., 2018). Implementationen medför således en signifikant kulturförändring, vilket är krävande att genomgå (Bierwolf et al., 2017). Att nå acceptans för den samarbetskultur som är framstående inom DevOps är viktigt men inte alltid enkelt (Wiedemann, 2018). Det kan vara svårt att motivera inblandad personal att förbättra samarbetet (Gupta et al., 2019). Förändring sker långsamt hos människor och det kan orsaka instabilitet, osäkerhet och motvillighet under övergången (Nybom et al., 2016). Om motvillighet hindrar interaktioner kvarstår en uppdelning som försämrar samarbete och samverkan (Qumer Gill et al., 2018)

När uppgifter från båda funktionerna, utveckling och *operations*, förväntas genomföras av alla, som i tillvägagångssättet med att blanda ansvar, medföljer särskild problematik. Det kan vara krävande att lära sig nya arbetsuppgifter. Om det till exempel är svårt för utvecklare att lära sig uppgifter från *operations* kan det medföra motvillighet. Att sedan skifta arbetsuppgifter mellan de två funktionerna är svårare än att endast behöva skifta uppgifter inom en funktion, vilket kan göra arbetet mer komplicerat och orsaka frustration (Nybom et al., 2016). Problematik kan även uppstå om det är svårt att uppnå ansvarstagande då utvecklare inte alltid känner ansvar för produkter bortom själva utvecklingsfasen (Qumer Gill et al., 2018)

Den tvetydighet som finns kring DevOps på grund av att konceptet inte har en standarddefinition eller fastställd implementeringsprocess kan också utgöra en nackdel (Nybom et al., 2016). Missförstånd av konceptets funktionalitet och syfte leder till problematik. Det finns en felaktig tanke om att snabbare processer alltid medför minskad kvalitet, vilket kan försämra förståelsen av DevOps. Ibland angrips det som en normativ metod eller en stor allt-i-ett-teknik, vilket också är en feluppfattning av konceptet (Qumer Gill et al., 2018).

3.6 Sammanfattning teori

Nedan följer en sammanfattande bild över denna rapports teori i figur 7 samt en förklarande text av teorins uppbyggnad. I ovalerna finns de tre rubrikerna implementering, implementeringsfaktorer och påverkan på team. Dessa rubriker utgör en grund för strukturen i empirin och diskussionen.



Figur 7 - Struktur över teoriavsnitt. Källa: Egen

För att förstå innebörden av de tre rubrikerna implementering, implementeringsfaktorer och påverkan på team behövs en del bakgrund och teorier om DevOps. Bakgrunden inleds med projektledningens teorier som växte fram redan under 30-talet. Sedan dess har stage-gate, vattenfall och agila modeller utvecklats. Inom agilt arbete finns ett agilt manifest som ligger till grund för olika ramverk och projektmetoder såsom scrum, SAFe och slutligen DevOps. DevOps utvidgar det agila arbetssättet och handlar om att övergå från silostrukturer till mer sammankoppling av funktioner. Implementeringen av DevOps bör anpassas efter företaget och det finns olika möjliga tillvägagångssätt. Fyra behov som kan vara bra att beakta är *Culture*, *Sharing*, *Automation* och *Measurement*. Det kan identifieras faktorer inom kommunikation, automation, ansvar, förändringsvillighet och samarbete som påverkar implementeringen samt

teamet. Till följd av implementeringen kan fördelar som bättre samarbete, snabbhet, effektivitet, kundvärde och förståelse uppnås. Det kan dock också medföra krävande förändring, nya och skiftande arbetsuppgifter, ökat krav på ansvar samt missförstånd.

4. Empiri

I detta avsnitt presenteras insamlat material som inleds med en nulägesbeskrivning. Därefter följer rubrikerna implementering, implementeringsfaktorer och påverkan på team.

4.1 Nulägesbeskrivning

Tidigare jobbade det studerade teamet mer enligt vattenfall men har successivt rört sig mer mot agila arbetssätt, enligt företaget. På senare år har delar av teamet arbetat mer agilt och därmed fått erfarenhet av att arbeta enligt bland annat scrum metoden. Under år 2019 började kunden uttala ett intresse för att arbeta med DevOps och mot hösten år 2020 började teamet implementationen. Förr brukade teamets resurser hyras ut leverans för leverans men numera är resurserna uthyrda på heltid. Enligt företaget har Covid-19 pandemin påverkat teamets arbete och övergången till DevOps, bland annat genom försenad implementering och ökat behov av distansarbete. Teamet, som innehåller personal från både fallföretaget och kunden, beskrevs av fallföretaget ha som syfte att leverera det värde som efterfrågas. Det inkluderar utveckling, design, förvaltning, drift och support inom system som används hos kunden.

4.2 Implementering

Implementeringen av DevOps förklarades syfta till att bryta barriären mellan *development* och *operations* genom att sammanfoga båda funktionerna i teamet. En Systemutvecklare uppgav att *development* och *operations* även innan detta har arbetat tillsammans men då var fortfarande ansvaret uppdelat. Att erfarenhet av denna kontakt fanns i teamet upplevde Scrum Mastern gav mer förståelse som kunde underlätta implementeringen. Vid övergången till DevOps skulle arbetet bedrivas mer som ett autonomt team där alla har ansvar för allt, enligt Systemutvecklaren. Scrum Mastern påpekade att implementationen är väldigt viktig för ett fortsatt fungerande arbete. Product Ownern menade att en god grund innebär att man istället för omvälvande förändringar kan justera under tiden.

4.2.1 Information innan implementering

Alla informanter förutom Product Ownern uppgav att de fått kunskap om DevOps från en intern SAFe utbildning med inriktning mot DevOps som fallföretaget tillhandahöll. Vikarierande Product Ownern och Scrum Mastern beskrev den utbildningen som deras första kontakt med konceptet medan resterande fem informanter sedan innan hört talas om det. En av Systemutvecklarna förklarade att man började höra mer om DevOps runt år 2016 och i samband med det började fallföretaget visa intresse för konceptet. Systemutvecklaren tillade också att det sedan var SAFe utbildningen med inriktning mot DevOps som gav mer praktisk förståelse.

Scrum Mastern berättade att den nämnda SAFe utbildningen gav teoretisk information. Utöver det menades det inte vara helt tydligt hur själva DevOps implementeringen skulle gå till och vilka skillnader det skulle leda till. Även en Systemutvecklare och Systemarkitekten menade att det inte gavs mycket information om hur implementeringen skulle gå till. En av Systemutvecklarna förklarade att informationen kom sent och Product Ownern menade att

tydlig information fanns i det SAFe ramverket som skulle följas. Konsultchefen berättade att information om kundens krav och syn på DevOps kommunicerats, men framförde även att kunden inte hade mycket praktisk erfarenhet. Systemarkitekten menade att kunden inte upplevdes veta hur implementeringen skulle genomföras. Istället förklarade Systemarkitekten att tillvägagångssättet kontinuerligt har fått arbetats fram tillsammans med kunden. En av Systemutvecklarna uppgav också att det skett metodiskt lite i taget och att teamet haft påverkan på vad som ska tas an. Systemutvecklaren framförde att Product Ownern som kom från kunden hade en viktig del i hur implementeringen skulle genomföras. Det förklarades vara Product Ownern som kommunicerat hur implementeringen praktiskt skulle ske enligt SAFe och DevOps.

4.2.2 Tillvägagångssätt

Product Ownern kommer från kunden och uppgavs ha arbetat agilt samt känt till DevOps konceptet under många år. Denna informant förklarades vara ansvarig för förberedelserna innan implementeringen. Det uppgavs att förberedelserna inkluderade att sätta upp verktygen, strukturerna och ramarna för arbetssättet. Det nämndes även att fallföretagets konsulter inte var tillgängliga att delta i förberedelserna.

Från fallföretagets perspektiv presenterade Konsultchefen i sin intervju att SAFe utbildningen med inriktning mot DevOps ingick i implementeringen. Vidare menades information om kundens arbetssätt och avtal ha delgetts till fallföretaget. Det förklarades att Konsultchefen tagit del i att sätta upp kriterierna utifrån beställarens krav, avtalsskrivandet samt utformningen av teamet. Systemarkitekten förklarade att teamet utgjordes av anställda från fallföretaget samt från kunden som bidrog med kompletterande kompetens inom främst *operations*. Systemarkitekten beskrev implementeringen som att teamet började arbeta efter DevOps principer.

Scrum Mastern berättade att dennes roll i implementeringen inkluderade att säkerställa den tvärfunktionella sammansättningen i teamet. Vidare angav Scrum Mastern att implementeringen började med SAFe utbildningen med inriktning mot DevOps. En Systemutvecklare framförde att efter den utbildningen började försök till praktisk applicering av teorin. Informanten med befattningen Vikarierande Product Owner uppgav att implementeringen innefattade anpassning av verktyg. Denne informant var delaktig genom utformningen av en *Delivery Pipeline* i syfte att uppnå *Continuous Integration* och *Continuous Deployment*. Det menades bidra till arbetet mot automatisering av systemet. Både Scrum Mastern och Systemutvecklaren menade i intervjuerna att Product Ownern var ledande i implementeringen av DevOps. Product Ownern menades ha skapat ett nytt projekt i verktyget Jira och definierat hur arbetet skulle bedrivas. Inom det förklarades det ingå tankar om att vara *T-shaped*, kompetensspridning och ständiga förbättringar. Systemutvecklaren framhäver även i intervjun att en agil coach i ett senare skede kopplades in för ytterligare stöd och säkerställande för att inte falla tillbaka i gamla arbetssätt.



Figur 8 - Systemutvecklarens uppfattning av implementeringen. Källa: Egen

Figur 8 illustrerar en av Systemutvecklarnas intervjusvar gällande implementeringen i teamet. Systemutvecklaren uppgav sig varit delaktig i implementeringen som en av medlemmarna i teamet. Det förklarades innebära att anamma DevOps principerna och bidra till spridningen av kunskap, främst gällande *operations* i detta fall. Systemutvecklarens uppfattning av implementeringen beskrevs som utformning av teamet, planering, ekonomiska frågor, kartläggning och automation. Utformningen av teamet förklarades innebära beaktande av vilka kunskaper personalen besatt i dåläget samt vad de var villiga att lära sig. Från Systemutvecklarens perspektiv uppgavs DevOps ställa höga krav på ansvarstagande hos enskilda personer. Därefter uppgavs att en intern planering enligt *Agile Release Trains* och *SAFe* bedrevs inom teamet som övning av konceptet. Sedan behövde ekonomiska frågor tänkas igenom, enligt Systemutvecklaren. Kartläggningen som därefter förklarades ha gjorts berörde vilken automation som fanns, vilka verktyg som kunde användas, vart förbättringar kunde göras och vilka förbättringar som kunde ge snabbt värde för minst resurser. Slutligen nämner Systemutvecklaren automation som en del av implementeringen.

Implementeringen upplevdes ske successivt enligt tre informanter. Vikarierande Product Ownern, Konsultchefen och Product Ownern använde förklarande ord som ”gradvis”, ”stegvis” respektive ”efterhand”. Scrum Mastern förklarade det däremot som en förändring ”från en dag till en annan” och ”som en 180° vändning” jämfört med tidigare arbetssätt.

4.2.3 Förbättring av implementeringsprocessen

Den agila coachen, som var en nyligen tillförd hjälp, hade med fördel kunnat introducerats tidigare i implementeringsprocessen, enligt en Systemutvecklare. Även Vikarierande Product Ownern och Product Ownern angav att det hade kunnat förenkla implementeringen. Systemutvecklaren förklarade att den agila coachens roll enbart fokuserat på teamets utveckling inom DevOps och menade att det kan vara bra att ha en dedikerad person som tar hand om övergången.

Ytterligare medvetenhet och beredskap för de förändringar som implementeringen skulle medföra hade kunnat förenkla processen, enligt en Systemutvecklare. Ett önskemål framfördes även av Systemarkitekten om en tydligare kommunicerad målbild från ledningen. Scrum Mastern menade att implementeringen började i fel ände och att den istället skulle börjat överst i kundens organisation. Systemarkitekten uppgav också att istället för att ett existerande team kompletterades borde kanske teamet stöpts om så att behövda resurser funnits från början. Product Ownern nämnde att det möjligen fanns ett större behov av utbildning inom ramverket inför implementeringen och poängterade att det är väldigt viktigt för att syftet med förändringar ska förstås.

4.3 Implementeringsfaktorer

En viktig faktor som Product Ownern och Vikarierande Product Ownern i intervjuerna menade har påverkan på implementeringen av DevOps är att ramverket följs. Ett stort fokus förklarades av Product Ownern läggas på att försöka “släppa gamla arbetsmetodiker” och arbeta enligt det nya arbetssättet till fullo. En Systemutvecklare menade dock på att det tog lång tid att anpassa sig till det nya arbetssättet. Istället för att arbete bedrevs helt från början till slut, förklarades en del uppgifter fortfarande hamna hos den med mest kompetens inom det området.

4.3.1 Kommunikation

Under intervjun med en av Systemutvecklarna framkom att “förut så kändes det att det var en liten barriär man skulle ta sig igenom för att kunna prata med någon hos kunden”. Systemutvecklaren menade att detta blivit bättre då kommunikationen nu är snabbare med kortare ledtider. Vikarierande Product Ownern bedömde kommunikationen som bra såväl inom teamet som med ledningen. Kommunikation är ett ständigt förbättringsarbete och en kontinuerlig dialog, enligt Product Ownern.

Med facit i hand så kunde man kanske jobbat ännu mer aktivt med det kan jag tänka. Det är liten kulturskillnad, det är en sak att starta upp ett helt nytt team från scratch där alla går in med samma blanka papper och ska lära sig ett arbetssätt här var det ju en existerande grupp som jobbade på ett sätt redan och skulle konvertera sitt arbetssätt. Och det ställer ganska mycket krav på att vara tydlig med kommunikation och vad förändringen innebär. (Konsultchef)

Systemarkitekten uppgav att kommunikationen inom teamet har fungerat bra men både Systemarkitekten samt en av utvecklarna påpekade att kommunikationen från ledningen kunde varit bättre. Systemarkitekten berättade att det upplevdes som att ledningen inte vetat säkert vad de vill. Systemutvecklaren påpekade ett stort behov av övning vid övergången till det nya arbetssättet, vilket teamet inte varit riktigt beredd på. Även hur det praktiskt skulle genomföras och den fysiska placeringen beskrevs som otydligt. Det kan ha orsakat en defensiv inställning hos teamet men kommunikation uppifrån var annars bra, enligt Systemutvecklaren.

4.3.2 Ansvar

Ledningens intresse för implementeringen framhövdes av Systemarkitekten som menade att stöd och resurser har givits för genomförandet. Riktlinjerna för hur det ska göras beskrevs dock som något vaga. Vikarierande Product Ownern sa i intervjun att “vi trodde nog att vi skulle bli lite bättre vägledda”. Scrum Mastern berättade i intervjun att det har märkts att implementation av DevOps hos kunden kan ha börjat i fel ände då det förklaras att förändringen behöver komma uppifrån. Implementeringen menades istället ha börjat nere hos teamet, vilket lett till brist på förankring och kunskap om SAFe och DevOps längre upp i kedjan. Även Vikarierande Product Ownern beskrev att en del system hos kunden följer agilt arbete medan andra fortfarande följer vattenfallsmetoder. Enligt Vikarierande Product Ownern leder skillnader i mognadsgrad till en del besvär och kommunikationssvårigheter då de som inte utvecklats lika mycket har svårt att förstå teamets arbetssätt.

4.3.3 Förändringsvillighet

Sex av sju informanter nämnde förändringsvillighet som en faktor som påverkat implementeringen av DevOps. En Systemutvecklare förklarade att det är viktigt att vara försiktig med förändring och menade att om för mycket görs på en gång kan det skapa motvillighet. Under intervjun förklarade Scrum Mastern att mindset och förändringskultur är påverkande faktorer. Det framfördes att en positiv inställning till förändringen varit avgörande för implementeringens framgång i teamet. Det är inte möjligt utan tilltro till förändringen, enligt Scrum Mastern. Vikarierande Product Ownern förklarade det som att “det är ju teamets vilja att bli bättre, det är en otroligt stark drivkraft”. Systemarkitekten påpekade också betydelsen av “att teamet har varit villiga att tagit sig an det här och är sugna på arbetsuppgifterna”. En av Systemutvecklarna framförde det som att “vi i vårt team brukar vara ganska beredd på att prova nya saker och det är jätteviktigt”. Att ha mod att prova det nya arbetssättet är viktigt, enligt Product Ownern.

4.3.4 Samarbete

Scrum Mastern menade att implementeringen av DevOps har gjort *operations* till en närmare del av teamet, vilket förklarades ha lett till ökad effektivitet och förenklad kommunikation. Konsultchefen framförde i intervjun att Covid-19 pandemin påverkat implementeringen. Det förklarades att “hela DevOps bygger på att fördelen är att man sitter tillsammans och pratar över bordet och det har dom ju faktiskt inte gjort förutom i början”. För att underlätta under de rådande omständigheterna förklarade en Systemutvecklare att tekniska lösningar har använts för att teammedlemmar ska kunna sitta tillsammans åtminstone virtuellt. Systemutvecklaren menade att det “minskar det där jobbiga att behöva ringa upp nån”.

4.4 Påverkan på team

Systemarkitekten berättade att arbetsuppgifterna har utvidgats sedan implementeringen av DevOps och nu även inkluderar uppgifter som tidigare bara tillhört funktionen *operations*. Det förklarades att alla nu ska arbeta med allt, vilket kan innebära att vissa måste arbeta med sådant där intresse saknas. Även om arbetsuppgifter inte nödvändigtvis ligger inom en anställds intresseområde förklarades det bidra till ökad kompetens, enligt Systemarkitekten. En av Systemutvecklarna framhävde i intervjun att det blivit betydligt fler möten sedan DevOps implementerats. Det förklarades handla om mycket planering och utvärderingar, vilket har inneburit “mindre egen tid på kammaren där man sitter fokuserat och kanske utvecklar någonting”. Product Ownern förklarade att allas deltagande på mötena är viktigt och att de har en tydlig agenda som syftar till att leverera ut något.

4.4.1 Fördelar

De fördelar från DevOps implementeringen som nämndes under intervjuerna visas i tabell 2. Antal repetitioner står för hur många av informanterna som nämnde vardera fördel och även utvalda citat från intervjuerna kan avläsas i tabellen.

Tabell 2 - Fördelar från DevOps implementeringen som nämndes under intervjuerna. Källa: Egen

Fördel	Antal repetitioner	Exempel citat
Konkurrenskraftighet	3	“Vi får ett bättre rykte av att vara ett effektivare team”. (Scrum Master)
Ökad kompetens	3	“Vi som jobbar i de här teamet åt kunden har fått en bredare kunskap”. (Systemarkitekt)
Bättre kompetensspridning	2	“Kompetensspridning sker per automatik mer eller mindre”. (Systemarkitekt)
Arbetet går snabbare	2	“Vi som team levererar mer kvalitet tror jag och att vi som team kan leverera snabbare”. (Scrum Master)
Ökat samarbete	2	“De samarbetar tätare, ännu tätare än tidigare”. (Konsultchef)
Bättre kommunikation	2	“Kommunikationen blir enklare”. (Scrum Master)
Tydlig rollfördelning	2	“Nu blev det ju tydligare rollfördelning dels”. (Scrum Master)
Närhet till kund	2	“Vi har kommit närmare kunden”. (Konsultchef)
Roligt arbetssätt	2	“Det är riktigt kul, jag tror de flesta tycker det är riktigt kul”. (Vikarierande Product Owner)
Enklare processer	1	“Alltså jag slipper hålla på med alla dom gamla processerna”. (Product Owner)

Vikten av att arbeta med de mjuka bitarna framhövdes i intervjun med Vikarierande Product Ownern som menade att stämningen har blivit bättre och att det nya arbetssättet upplevs roligt. Även Systemarkitekten ansåg att det var roligt och uttryckte glädje över att fått tagit del av det. Det förklarades att ett större engagemang kommit från den delaktighet som sträcker sig från början till slut. Systemarkitekten uppgav att “när man ska vara med i själva *operations* av det man bygger, gör man det inte ordentligt så biter det en i svansen sen att man får massa support och fel”.

En av Systemutvecklarna menade att efter implementeringen av DevOps går arbetet snabbare då teamet är mer sammankopplat och det är “fler ögon på saker när man jobbar”. Under intervjun förklarade Scrum Mastern att problem blir lösta snabbare, vilket minskar ledtider. En av Systemutvecklarna förklarade att automation och kontinuerliga förbättringar nu prioriteras, vilket bidrar till att förhindra problem. Scrum Mastern menade att ett större fokus på mätningar

gör det enklare att upptäcka och åtgärda problem. Om något går dåligt eller inte fungerar förklarades det märkas direkt och därför kan ett proaktivt arbete bedrivas så att kunden inte behöver drabbas. Systemutvecklaren påpekade också att mätvärden kan användas för att konkretisera kompetens uppåt och gentemot beställare. På så sätt förklarades fallföretaget kunna visa att de kan erbjuda välfungerande DevOps team. Vidare menade Scrum Mastern att genom ett närmare samarbete med *operations* och snabbare leveranser kan teamet uppnå ett rykte om snabbhet, effektivitet och kvalitet.

Vi är inte riktigt där, men snart skulle jag säga ... så kan du ställa in det här teamet i princip hos vilken kund som helst, sen tar det ett tag att lära sig verksamheten, men som DevOps team funkar dom riktigt, riktigt bra. (Vikarierande Product Owner)

Scrum Mastern och Vikarierande Product Ownern berättade att teamet blivit mer *T-shaped*. Teammedlemmarna har, enligt Vikarierande Product Ownern, fått lära sig mycket sedan implementeringen av DevOps och Systemarkitekten uppgav att kunskapsbasen har breddats. Systemarkitekten framförde även att kompetensspridningen förbättrats. En av Systemutvecklarna förklarade att dedikerade möten förut behövde hållas då något nytt skulle läras men att det numera kan ske mer automatiskt i exempelvis ett ärende som ändå ska utvecklas.

Övergången till DevOps har enligt Konsultchefen bidragit till mer närhet och transparens gentemot kunden. Det förklarades att teamet numera jobbar i kundens verksamhet istället för att vara outsourcat, vilket gör det mer uppenbart för kunden vad teamet gör. Product Ownern som kommer från kunden menade att när mer nu sköts direkt hos kunden har processer, som till exempel fakturering, förenklats. En av Systemutvecklarna berättade att mer samhörighet till kunden upplevs då anställda från fallföretaget och kunden nu är förenade i ett team. Systemutvecklaren menade också att det bidragit till enklare kommunikation, vilket även Scrum Mastern nämnde.

4.4.2 Nackdelar

Vid en fråga om vilka nackdelar DevOps medfört för fallföretaget kommer en av Systemutvecklarna, Vikarierande Product Ownern, Systemarkitekten och Scrum Mastern inledningsvis inte på några. Konsultchefen, Scrum Mastern och Systemutvecklaren förklarade att de nackdelar som slutligen nämnts verkar vara övergående och kan förbättras med tiden. Tabell 3 visar de nämnda nackdelarna som uppkom under implementeringen och antal repetitioner anger hur många informanter som nämnde vardera. Därutöver har några citat från intervjuerna valts ut.

Tabell 3 - Nackdelar från DevOps implementeringen som nämndes under intervjuerna. Källa: Egen.

Nackdel	Antal repetitioner	Exempel citat
Sämre leveransförmåga	5	“Det har tagit tid och energi att komma in i det här arbetssättet, som har tagit bort tid från kanske att producera”. (Konsultchef)
Högre arbetsbelastning	3	“Den första tiden var ju otroligt stressande”. (Systemutvecklare)
Större behov av resurser	2	“Det har också medfört att vi behöver fler resurser än tidigare och så skulle det ju inte riktigt vara”. (Konsultchef)

Vikarierande Product Ownern menade att det i början tog längre tid att leverera men framhävde att det berott på bristande kunskap om arbetssättet. En initial skepticism förklarades nu ha övergått i tilltro till arbetssättet. Scrum Mastern påpekade under intervjun att det behövs tålamod eftersom det under uppstarten kan gå lite sämre. På grund av det menade konsultchefen att det har behövts fler resurser men framförde också en föreställning om att det är övergående. Systemarkitekten förklarade att den ökade kompetensspridningen som implementeringen har medfört har krävt tid och resurser. Det förklarades ha bidragit till att teamet inte hunnit leverera ut i samma utsträckning.

En av Systemutvecklarna uppgav även att den första tiden var påfrestande för individerna i teamet. En viss beredskap för de inlärningskrav implementeringen skulle medföra förklarades ha funnits men då det missbedömdes ökade ändå belastningen. Vikarierande Product Ownern framhävde att det periodvis blir jobbigt på grund av den höga arbetsbelastningen. Systemutvecklaren menade att arbetsbelastningen och leveransförmågan påverkats eftersom det saknas vana av arbetssättet och mycket detaljer gällande tillvägagångssättet har behövts diskuteras fram. Systemutvecklaren menade att detta kommer gå lättare då teamet fått mer erfarenhet.

4.4.3 Teamets förslag på förbättringar

Ett behov av fortsatt förbättring av kompetensspridning framfördes under intervjun med Systemarkitekten. Det föreslogs att det skulle kunna gå snabbare om någon ny medlem med behövd kompetens togs in i teamet. Scrum Mastern förklarade att det i nuläget är främst en person som har uppgifter inom *operations* i teamet. Det förklarades innebära ett visst personberoende och belyste ett behov av att teamet blir mer *T-shaped*. Konsultchefen nämnde att mer arbete med kulturen och mätningar av leveranser kan leda till förbättringar. Arbetet med förbättringar är kontinuerligt enligt Vikarierande Product Ownern, en Systemutvecklare och Product Ownern. Systemutvecklaren uppgav också att behovet för förändringar har minskat i samband med att arbetet fungerar allt bättre och att förbättringsarbetet hittills har finslipat arbetssättet.

5. Diskussion och analys

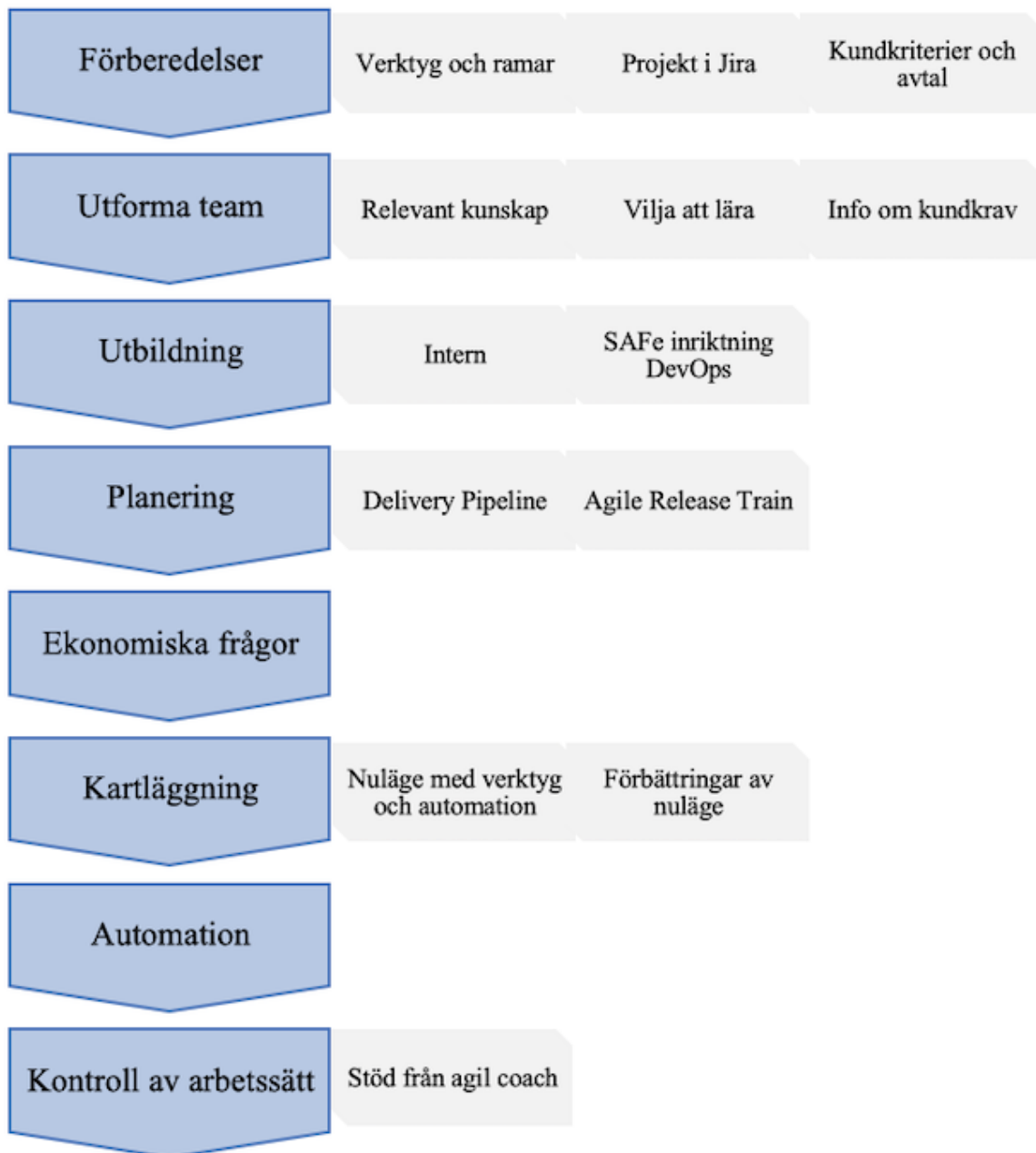
I följande text diskuteras och analyseras den teoretiska referensramen och det empiriska materialet. Strukturen följer rubrikerna implementering, implementeringsfaktorer och påverkan på team.

5.1 Implementering

DevOps är ett sätt att sammankoppla funktionerna *development* och *operations* (Diel et al., 2016). Fallföretagets syn på konceptet stämmer överens med teorin då teamets användning av DevOps har som avsikt att bryta barriären mellan de två funktionerna. Företag måste själva utforma implementeringen av DevOps eftersom det saknas ett fastställt definierat tillvägagångssätt enligt Nybom et al. (2016). Detta kunde ses hos fallföretaget som följde ett ramverk för införandet men ändå själva uppgavs ha definierat hur det skulle gå till. I intervjuerna framkom dock att genomförandet av implementeringen inte var tydligt för alla informanterna. Det uppgavs även att kunden inte verkade veta hur det skulle gå till. Otydligheten kring implementeringen kan därav berott på att kunden inte hade mycket praktisk erfarenhet av DevOps. Det framgår i teorin att det är viktigt att på förhand kunna definiera och kommunicera konceptets innebörd, syftet med implementeringen samt dess inverkan enligt Nybom et al. (2016). Det är svårt när erfarenhet saknas men ett större fokus på detta från teamets ledning hade kunnat förenkla implementeringen.

5.1.1 Tillvägagångssätt

DevOps kan användas som en tillfällig lösning eller en långsiktig förändring till tvärfunktionella team. Däremellan finns dessutom hybridlösningar (Wiedemann, 2018). I intervjuerna nämndes inget om att DevOps används för endast en begränsad tid och det talades om att implementeringen hoppades ge en bra grund för fortsatt arbete. Det tyder på att teamet i det här fallet ska förbli tvärfunktionellt på lång sikt. Nybom et al. (2016) presenterar de tre tillvägagångssätten blanda ansvar, blanda personal och överbryggande team. Av dessa kan fallföretaget tänkas tillämpat en kombination av blanda ansvar och blanda personal. I enlighet med tillvägagångssättet att blanda ansvar, uppgavs att alla teammedlemmar har ansvar för allt. Dock angavs fortfarande en person bära det främsta ansvaret för *operations*. Det tyder på fortsatt uppdelade roller som stämmer mer med tillvägagångssättet att blanda personal.



Figur 9 - Tolkning av fallföretagets implementeringsprocess. Källa: Egen

I Figur 9 har Systemutvecklarens perspektiv från figur 8 kompletterats med övriga informanternas svar, vilket gett denna utökade tolkning av implementeringsprocessen på fallföretaget. Vissa kopplingar kan dras till teorins fyra nivåer av DevOps implementering *Culture*, *Sharing*, *automation* och *Measurement* som kan ses i figur 6 (Muñoz & Rodríguez, 2021). I förberedelsefasen fastställdes ramar som kan tänkas främja den önskade kulturen. *Culture* kan även ses i utformningen av teamet där viljan att lära betonades och i utbildningsfasen där lärande om DevOps kultur kan tänkas ha ingått. Även *Sharing* kan ses i förberedelsefasen, utformningen av teamet och utbildningsfasen eftersom information kring kundkriterier, kundkrav respektive arbetssättet sprids. Automation finns tydligt med som en egen fas och men *Measurement* togs inte upp i intervjuerna vid frågor om implementeringens genomförande. Dock nämndes det vid andra delar av intervjuerna att fler mätningar görs sedan införandet av

DevOps, vilket tyder på att *Measurement* trots allt haft en del i implementeringen. De fyra nivåerna från figur 6 i teorin stöds av *Flow*, feedback och *Continuous Learning* (Muñoz & Rodríguez, 2021). Alla tre kan ha haft inflytande men endast feedback och *Continuous Learning* kan tydligt urskiljas ur intervjuvaren. I kartläggningen identifieras till exempel förbättringsmöjligheter och vid kontroll av arbetssättet används feedback från den agila coachen för en fortsatt kontroll samt förbättring av arbetet.

5.1.2 Förbättring av implementeringsprocessen

Från intervjuerna framkom några förbättringsförslag på fallföretagets utförande av implementeringen. Den agila coachen upplevdes som en bra tillförd hjälp men ett förslag var att ha en dedikerad person som kan fokusera på implementeringen redan från början. Ett förslag kan således vara att någon vägleder hela implementeringsprocessen. Det kan då gå enklare att genomföra och bemöta eventuella hinder. Det framfördes även en önskan om tydligare kommunikation från ledningen. Eftersom det även nämndes att ledningen saknade erfarenhet och att implementeringen börjat i fel ände, bör implementeringen istället ha börjat hos ledningen. Även här hade en dedikerad erfaren person som den agila coachen möjligen kunnat förutse detta behov och motverkat otydlighet från början. Ytterligare ett förslag var att DevOps teamet från början skulle innehålla de kompetenser som behövdes istället för att kompletteras med nödvändiga resurser efterhand. Betydelsen av att alla utbildades för att förstå förändringarna poängterades också. Det kan anses viktigt att utbildning och behövda resurser finns i teamet för att undvika missförstånd och skapa förståelse för syftet med DevOps.

5.2 Implementeringsfaktorer

Samtliga av de fem implementeringsfaktorerna som identifierats i teorin kan i någon form återfinnas i empirin. Fallföretagets tillämpning av automation beskrevs dock inte särskilt detaljerat. Det nämndes i intervjuerna att implementeringen har inneburit att automation numera prioriteras mer. Detta stärks av Nybom et al. (2016) som framhäver automationens väsentlighet inom DevOps. De övriga implementeringsfaktorerna inom kommunikation, ansvar, förändringsvillighet och samarbete analyseras och diskuteras mer nedan.

5.2.1 Kommunikation

I både intervjuvaren och Indriasari et al. (2020) framgår det att övergången till DevOps orsakar kulturförändringar. Vikten av kommunikation framförs av Indriasari et al., (2020), Nybom et al. (2016) och Qumer Gill et al. (2018). Även informanterna nämnde krav på tydlig kommunikation men hade olika uppfattningar av hur det arbetet såg ut. Enighet kunde ses kring att det fungerat bra inom teamet och det angavs att kommunikationen förbättrats i samband med implementeringen. Det framkom dock att kommunikationen med ledningen upplevdes lite olika. Kanske kan denna skillnad bero på att informanternas olika roller ger olika perspektiv. Vikarierande Product Ownern uppgav att kommunikationen med ledningen fungerat bra men denne informant står också lite mer utanför teamet i en mer stödjande roll. Systemarkitekten och en Systemutvecklare, som är direkt verksamma inom teamet, menade att det hade kunnat fungerat bättre. Det upplevdes saknas tydlighet kring vart ledningen ville och hur det skulle

uppnås. Som Nybom et al. (2016) och Qumer Gill et al. (2018) förklarar är det viktigt att kommunicera just vad DevOps ska innebära för en organisation och, enligt Indriasari et al. (2020), hur det kommer påverka arbetet. Då detta inte gjordes med tydlighet är det förståeligt att det i intervjuerna framfördes ett behov av mer fokus på kommunikation. Det uppgavs också att otydlighet kan ha orsakat en något defensiv inställning. Det stärker påståendet från teorin om att kommunikation har stor betydelse vid implementering av DevOps. Qumer Gill et al. (2018) påpekar att det behövs fortsatt träning samt coaching för att nå förståelse. Det verkar fallföretaget ha beaktat eftersom kommunikation beskrivs som ett ständigt förbättringsarbete och en kontinuerlig dialog.

5.2.2 Ansvar

Att DevOps innebär mindre bestämmande från ledning, decentralisering och eget beslutfattande framgår i både Wiedemann (2018) samt Indriasari et al. (2020). Detta innebär också mer ansvarstagande (Wiedemann, 2018) som ska delas av alla (Qumer Gill et al., 2018). Detta stämmer överens med intervjusvaren kring implementeringen av DevOps i teamet som beskrevs syfta till att uppnå självbestämmande och delat ansvar. Trots att decentralisering och självbestämmande framhävs kan ledningen ändå anses ha en väldigt viktig roll. Nybom et al. (2016) förklarar att ledningens stöd behövs och Qumer Gill et al. (2018) menar att uppmaning för ansvarstagande bör komma från ledningen. I intervjun framgår att ledningen visat intresse och stöd för implementeringen men att vägledningen varit lite svag. Det kan anses att ledningen erbjudit stöd men att mer kunnat behövts. Detta kan möjligen bero på bristande kunskap hos ledningen. I intervjuerna framförs att implementering borde ha börjat hos ledningen. Det framgår att skillnaderna i mognadsgrad inom DevOps försvårat bland annat kommunikationen. Om teamet kommit längre och lärt sig mer är det förståeligt att resten av organisationen har svårt att förstå och ge det stöd som förespråkas i teorin. Trots att det inom DevOps team är mycket självbestämmande och ansvar som teorin framhäver, är det tydligt från intervjuerna att ledningens stöd också är viktigt. På grund av detta kan det vara viktigt att ledningen först har kunskap kring DevOps för att sedan kunna stödja teamet i implementering.

5.2.3 Förändringsvillighet

Under intervjuerna framgick det tydligt att informanterna ansåg att förändringsvillighet var en viktig faktor under implementeringen. Ord som mindset, förändringskultur och positiv inställning användes för att beskriva förändringsvilligheten. Wiedemann (2018) påpekar att attityd och inställning är avgörande för implementeringen, vilket teamet verkar medvetna om. Därutöver nämner Nybom et al. (2016) att förändringar lätt bemöts med motvillighet vilket också nämndes under intervjuerna. Genom att införa DevOps inkrementellt kan övergången underlättas (Gupta et al., 2019). I intervjuerna framkom att implementeringen skett gradvis, vilket kan vara en anledning till att motvillighet inte har upplevts hos teammedlemmarna. Gupta et al. (2019) beskriver att teamet genom inriktning på affärsvärde och korta frigöringscykler kan fokusera på de mest behövda delarna först och sedan inkrementellt förbättra ytterligare. Dock fanns motstridande svar gällande implementeringsprocessen i empirin som både beskrevs som successiv men även plötslig. Detta kan tänkas bero på hur frågan uppfattades av informanterna. Givetvis är det möjligt att informanterna helt enkelt upplevde

implementeringsprocessen olika. Dock skulle också “180° vändningen” kunna hänvisa till att det var en stor förändring som inleddes, vilket inte utesluter att implementeringen fortsättningsvis skett successivt.

5.2.4 Samarbete

Implementeringen av DevOps på fallföretaget förklarades ha lett till ökad effektivitet och förenklad kommunikation genom att *operations* har blivit en närmare del av teamet. En samarbetsfaktor som kan ge möjlighet till ökad samverkan, kunskap, stöd och motivation inom teamet är enligt Gupta et al. (2019) att sitta tillsammans. Det beskrevs dock under intervjuerna att Covid-19 pandemin har försvårat implementeringen och samarbetet då teamet inte har haft samma möjlighet att sitta tillsammans fysiskt. Wiedemann (2018) menar att oavsett om teamet sitter nära eller inte måste det finnas förutsättningar för kommunikation och kunskapsdelning. Sådana förutsättningar har försökt uppnås på fallföretaget genom att teamet har använt virtuella tekniska lösningar för att underlätta distansarbetet. Detta kan anses vara en utmärkt tillfällig lösning tills möjligheten att återigen sitta tillsammans finns.

5.3 Påverkan på team

5.3.1 Fördelar

En av de fördelar som flest informanter nämnde under intervjuerna var den konkurrenskraft som DevOps medför. I samband med att utveckling och *operations* sammankopplas menar Nybom et al. (2016) att bättre service kan erbjudas och enligt Gupta et al. (2019) kan kundvärdet öka. Detta stämmer in på fallföretaget som enligt intervjuerna nu utvecklar ett ännu bättre rykte. Det större fokuset på mätningar som DevOps medfört gör det även lättare för fallföretaget att visa teamets kompetens för beställare. I intervjuerna framkom även att de ökade mätningarna och automationen gjort det enklare att upptäcka samt förhindra misstag. Detta är även något som Qumer Gill et al. (2018) tar upp genom att beskriva hur automation kan minska risken för misstag orsakade av exempelvis mänskliga faktorn. Färre misstag kan förmodas främja bättre kvalitet och öka konkurrenskraften.

Automation kan enligt Gupta et al. (2019) ge mer snabbhet och enligt Ebert et al. (2016) bidrar DevOps till snabbare utgivningscykler. Även Sill (2014) framhäver snabbhet från idé till användning som en fördel. Denna ökade snabbhet av leveranser upplevdes på fallföretaget enligt intervjuerna. Även mer effektivitet nämndes både under intervjuerna och av Nybom et al. (2016) i teorin. Kompetensspridningen har också förbättrats enligt intervjuerna och beskrivs nu som mer automatisk. Implementeringen innebär enligt Ebert et al. (2016) att kunskaper hos teammedlemmar måste vidgas, vilket upplevdes på fallföretaget. Det framhövdes i intervjuerna att implementeringen bidragit till nya kunskaper och kompetenser för medlemmarna i teamet. Med tanke på att DevOps team är tvärfunktionella kan detta anses vara en positiv följd av införandet av konceptet.

Fördelar från samarbete mellan utveckling och *operations* poängteras i teorin av Nybom et al. (2016) men även under intervjuerna. Nybom et al. (2016) förklarar att samarbetet leder till en bättre förståelse mellan funktionerna som i sin tur bidrar till bättre lösningar. Detta

exemplifieras tydligt av intervjuerna där det beskrevs att arbetet görs mer ordentligt eftersom det finns mer insyn i eventuella följder eller komplikationer. Om utveckling sker med *operations* i åtanke kan problem tänkas undvikas och bättre kvalitet uppnås. I intervjuerna framkommer att bättre samarbete bidrar till kvaliteten och Ebert et al. (2016) konstaterar att DevOps ger bättre kvalitetssäkring.

Enligt intervjuerna har implementeringen av DevOps gett fallföretaget mer närhet till kunden och lett till mer transparens. En fördel som togs upp i intervjun med kundrepresentanten, som är Product Owner för teamet, var att processer förenklas då de numera görs hos kunden. Huruvida transparens och förenklade processer är kopplade till DevOps är dock inte helt säkert eftersom teamet i samband med implementeringen även övergick från att vara outsourcat till att istället arbeta i kundens verksamhet. Dessa fördelar har inte identifierats i den teorin som använts, vilket också kan indikera att de kanske beror på andra förändringar än just DevOps implementeringen. En annan fördel som inte setts i teorin men som framkom under intervjun är att teamet upplever arbetet med DevOps som roligt.

5.3.2 Nackdelar

Flera av informanterna kom inledningsvis inte på några nackdelar under intervjuerna, vilket skulle kunna tyda på att de som slutligen nämndes inte haft en alltför märkbar inverkan.

Den nackdel som flest informanter ändå nämnde under intervjuerna var sämre leveransförmåga. Det förklarades att avsaknaden av erfarenhet vid uppstarten krävde mycket tid, vilket gjorde det svårare att leverera lika mycket som innan. Nybom et al. (2016) förklarar att krav på inläring medföljer tillvägagångssättet blanda ansvar, vilket fallföretaget verkar ha upplevt. Denna inläring kan enligt Nybom et al. (2016) vara krävande vilket kan förklara varför teamet från fallföretaget inte haft samma leveransförmåga under implementeringen. Det större kravet på lärande under uppstarten har enligt intervjuerna också lett till att fler resurser behövs. I början av implementeringsprocessen, då vana ännu inte finns, verkar således två nackdelar vara sämre leveransförmåga och större behov av resurser.

Den krävande uppstarten, fortsatt lärande och nya ansvarsområden kan även tänkas orsaka ökad belastning på teammedlemmar. Nybom et al. (2016) menar att när tillvägagångssättet blanda ansvar används för DevOps implementeringen behöver personal skifta mellan uppgifter från utveckling och *operations*. Detta försvårar arbetet eftersom det är mer komplicerat än att endast arbeta inom en funktion, enligt Nybom et al. (2016). I intervjuerna framgick att teammedlemmarna från fallföretaget behövt vidga sina arbetsuppgifter efter implementering och att det ibland är hög arbetsbelastning. Det kan anses troligt att den ökade belastningen beror på utökningen av arbetsuppgifterna och den inläring det krävt. I intervjuerna nämndes att de nya arbetsuppgifterna inte föll inom alla teammedlemmars intresseområde. Detta kan vara ett tecken på den motvillighet mot att lämna traditionella silostrukturer som Qumer Gill et al. (2018) presenterar.

Alla de nackdelar som hittills diskuterats kan tänkas vara begränsade till uppstarten. I intervjuerna var det tre informanter som direkt beskrev nackdelarna som övergående. Även i teorin beskrivs nackdelar som hänvisar till den förändring som implementeringen medför. Bierwolf et al. (2017) menar att förändringar måste ske för att fördelarna med DevOps ska uppnås. Flera källor beskriver denna förändring som svår och presenterar nackdelar det kan medföra. Nybom et al. (2016) nämner instabilitet under förändringar vilket skulle kunna kopplas till fallföretagets upplevda påverkan på leveransförmågan. Bierwolf et al. (2017) menar också att övergången är krävande, vilket teamet på fallföretaget upplevt bland annat genom det ökade resursbehovet. Svårigheter att få teammedlemmar att acceptera och motiveras av DevOps samarbetskultur nämns av Gupta et al. (2019) och Wiedemann (2018). Detta kan tänkas upplevas hos de teammedlemmar som i intervjuerna uppgavs ha bristande intresse för de nya arbetsuppgifterna. Eftersom dessa nackdelar i teorin samt i vissa intervjuer anses gälla just under implementationen kan det tänkas att teamet från fallföretaget inte kommer ha dessa problem på längre sikt. Det kan anses oundvikligt att en förändring medför en del svårigheter men dessa kan upphöra då erfarenhet och kunskap ökar.

En nackdel som nämns i teorin av Qumer Gill et al. (2018) men som fallföretaget inte verkar ha upplevt är svårigheten att få utvecklare att känna ansvar bortom utvecklingsfasen. I kontrast uppgavs under intervjuerna att den ökade delaktigheten som DevOps inneburit istället ökat motivationen att utföra ett bra arbete. Det tyder på att det finns ett ansvarstagande i teamet.

En nackdel kan, enligt Nybom et al. (2016), komma från att DevOps innebörd eller implementeringsprocess inte har en tydlig definition. Qumer Gill et al. (2018) menar att missförstånd av konceptet sedan kan leda till problem. Sådana problem eller missförstånd nämndes inte under intervjuerna och har därav kanske inte uppstått på fallföretaget. Dock uppgavs att implementeringen ha försvårats på grund av att kunden och ledningen saknat kunskap och erfarenhet kring DevOps. Det skulle kunna innebära att missförstånd trots allt har skett, även om informanterna inte beskriver det. Från intervjuerna framgår att den bristande erfarenheten har haft följder såsom inledande skepticism, otydlighet kring implementering och missbedömning av krav. Dessa följder kan också tänkas ha bidragit till nackdelarna som upplevts i teamet.

5.3.3 Teamets förslag på förbättringar

I intervjuerna framfördes förslag för hur fallföretaget kan förbättra sitt nuvarande arbete. Ett förslag var att teamet skulle kunna dra fördel från en ny medlem med behövlig kompetens inom främst *operations*. Det skulle kunna minska personberoendet och göra teamet mer T-shaped, vilket det finns behov av enligt intervjuerna. Fortsatta förbättringar av arbetet med kompetensspridning, kulturen och mätningar nämndes också. Eftersom förbättringsarbetet beskrevs som kontinuerligt i intervjuerna är det möjligt att dessa områden redan genomgår förbättringar. Enligt intervjuerna har behovet av förbättringar minskat under implementeringens fortgång. Det kan dock tänkas att det behovet aldrig försvinner helt eftersom förbättringsarbetet är kontinuerligt och det alltid lär finnas möjligheter att utvecklas ytterligare.

6. Slutsats

Följande avsnitt presenterar slutsatser och rekommendationer som uppfyller arbetets syfte och frågeställningar. Därutöver redovisas arbetets begränsningar och framtida forskning föreslås.

6.1 Teoretiskt bidrag

Detta arbete syftade till att granska hur DevOps implementering inom IT-konsultbranschen kan gå till och hur ett team kan påverkas av detta. För att uppnå detta syfte och följaktligen forma ett vetenskapligt bidrag har följande frågor besvarats:

6.1.1 Hur kan implementeringen av DevOps genomföras?

Det finns inget fastställt tillvägagångssätt för hur implementeringen ska gå till men olika möjliga angreppssätt kan användas. Genomförandet av DevOps implementering kan således se olika ut och behöver anpassas till varje enskilt företag. Det framgår i studien att det är viktigt att både definiera och kommunicera vad företaget vill uppnå och hur det ska ske. Beaktande av faktorer som kommunikation, automation, ansvar, förändringsvillighet och samarbete framstår vara betydande för implementeringens framgång. Att ha en dedikerad person, börja hos ledningen samt ha behövda resurser och utbildning i teamet är rekommendationer som vuxit fram under studiens gång. Eftersom många faktorer spelar in och implementeringsprocessen behöver anpassas för varje företag är det, som tidigare konstaterat, svårt att specificera ett bestämt tillvägagångssätt.

6.1.2 Vilken påverkan har DevOps på ett team?

I studien framkommer att DevOps i någon grad påverkar samtliga av faktorerna kommunikation, automation, ansvar, förändringsvillighet och samarbete hos ett team. Det finns både för- och nackdelar med DevOps. Några fördelar som i studien framstod tydligast var ökad kompetens och konkurrenskraft. Den mest framstående nackdelen var försämrad leveransförmåga. Den, och flera andra nackdelar, framstod som övergående men dess påverkan på teamet bör ändå beaktas.

6.2 Praktiskt bidrag

Det praktiska bidraget från detta arbete tillför rekommendationer för företag inom IT-konsultbranschen. Dessa rekommendationer, som listas nedan, är framtagna för att underlätta implementeringen av DevOps och den påverkan på team som kan finnas.

- Kunskap behöver förankras hos ledningen i organisationen innan implementering inleds inom teamet.
- Lämna plats för inläring och uppstartssträcka i början.
- Mer utbildning inom DevOps kan ge ytterligare förståelse för konceptet och dess syfte.
- Se till att teamet innehåller all behövlig kompetens inom såväl utveckling som *operations*.
- Ha en dedikerad person vid implementering av DevOps, exempelvis en agil coach.
- Kontinuerligt förbättringsarbete bör alltid finnas.

6.3 Arbetets begränsningar

Eftersom arbetet endast utförts under en tidsperiod på 10 veckor har fallstudiens omfattning fått begränsas till undersökning av enbart ett team. Covid-19 pandemin har även begränsat möjligheterna för studiens genomförande på sådant sätt att endast virtuella intervjuer varit utförbara.

6.4 Framtida forskning

Baserat på denna studies bidrag och slutsatser har förslag till framtida forskning tagits fram, vilka kan ses nedan.

- För ett bredare perspektiv skulle vidare studier förslagsvis kunna följa ett team under hela implementeringsprocessen istället för att undersökningen görs i efterhand. Alternativt skulle flera team som befinner sig i olika skeden av implementeringen kunna undersökas.
- Mer kunskap om DevOps långsiktiga påverkan skulle kunna uppnås genom studier av team som har arbetat enligt DevOps längre.
- Generaliserbarheten skulle kunna öka genom att undersöka hur rekommendationerna kan tillämpas på liknande företag inom IT-konsultbranschen men även inom andra organisationer eller branscher.
- Avslutningsvis skulle det kunna studeras vilken effekt en erfaren dedikerad person har på implementeringsprocessen.

Referenser

- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., & Jeffries, R. (2001). *Manifesto for agile software development*.
- Bell, J. (1993). *Doing your research project: A guide for first-time researchers in education and social science* (2. ed.). Open University Press.
- Bell, J., & Waters, S. (2014). *Doing Your Research Project: A Guide for First-time Researchers* (6. ed.). McGraw-Hill Education.
- Bergman, B., & Klefsjö, B. (2012). *Kvalitet från behov till användning* (5. ed.). Studentlitteratur AB.
- Bierwolf, R., Frijns, P., & van Kemenade, P. (2017). Project management in a dynamic environment: Balancing stakeholders. *2017 IEEE European Technology and Engineering Management Summit (E-TEMS)*, 1–6. <https://doi.org/10.1109/E-TEMS.2017.8244226>
- Blomkvist, P., & Hallin, A. (2014). *Metod för teknologer: Examensarbete enligt 4-fasmodellen*. Studentlitteratur AB.
- Christopher, M. (2011). *Logistics & Supply Chain Management* (4. ed). Pearson Education Limited.
- Comstedt, M. (2019, March 11). Detta är DevOps, en grundläggande sammanfattning. *Onbird*. <https://onbird.se/devops-en-grundlaggande-sammanfattning/>
- Davis, J., & Daniels, R. (2016). *Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale*. O'Reilly Media, Inc.
- Diel, E., Marczak, S., & Cruzes, D. S. (2016). Communication Challenges and Strategies in Distributed DevOps. *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, 24–28. <https://doi.org/10.1109/ICGSE.2016.28>
- Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *IEEE Software*, 33(3), 94–100. <https://doi.org/10.1109/MS.2016.68>
- Faizi, S. M., Rahman, S., & Hopkins, K. (2019). Implementing Large Enterprise Resource Planning Systems with Agile Methods. *2019 2nd International Conference on Innovation in Engineering and Technology (ICIET)*, 1–6. <https://doi.org/10.1109/ICIET48527.2019.9290587>
- Galup, S., Dattero, R., & Quan, J. (2020). What Do Agile, Lean, and ITIL Mean to DevOps? *Communications of the ACM*, 63(10), 48–53. <https://doi.org/10.1145/3372114>
- Gemino, A., Horner Reich, B., & Serrador, P. M. (2021). Agile, Traditional, and Hybrid Approaches to Project Success: Is Hybrid a Poor Second Choice? *Project Management Journal*, 52(2), 161–175. <https://doi.org/10.1177/8756972820973082>

- Gupta, R. K., Venkatachalapathy, M., & Jeberla, F. K. (2019). Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team: A Case Study of a Healthcare Organization. *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*, 30–34. <https://doi.org/10.1109/ICGSE.2019.00020>
- Hallin, A., & Karrbom Gustavsson, T. (2019). *Projektledning* (3. ed.). Liber.
- Hosono, S., & Shimomura, Y. (2012). Application Lifecycle Kit for Mass Customization on PaaS Platforms. *2012 IEEE Eighth World Congress on Services*, 397–398. <https://doi.org/10.1109/SERVICES.2012.80>
- Indriasari, E., Supangkat, S. H., & Kosala, R. (2020). Digital Transformation: IT Governance In The Agile Environment A Study Case Of Indonesia High Regulated Company. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, 9(4), 1557–1562.
- Larman, C., & Basili, V. R. (2003). Iterative and Incremental Developments: A Brief History. *Computer*, 36(6), 47–56. <https://doi.org/10.1109/MC.2003.1204375>
- Loseke, D. R. (2013). *Methodological Thinking: Basic Principles of Social Research Design*. SAGE Publications.
- Maroukian, K., & Gulliver, S. R. (2020). The link between transformational and servant leadership in DevOps-oriented organizations. *Proceedings of the 2020 European Symposium on Software Engineering*, 21–29. <https://doi.org/10.1145/3393822.3432340>
- Muñoz, M., & Rodríguez, M. N. (2021). A guidance to implement or reinforce a DevOps approach in organizations: A case study. *Journal of Software: Evolution and Process*, 1–21. <https://doi.org/10.1002/smr.2342>
- Neve, J. R., Godbole, K., & Neve, R. (2017). Productivity and process improvement using ‘Scaled Agile’ approaches: An emphasized analysis. *2017 International Conference on Inventive Computing and Informatics (ICICI)*, 793–798. <https://doi.org/10.1109/ICICI.2017.8365245>
- Nybom, K., Smeds, J., & Porres, I. (2016). On the Impact of Mixing Responsibilities Between Devs and Ops. In H. Sharp & T. Hall (Eds.), *Agile Processes, in Software Engineering, and Extreme Programming* 131–143. Springer International Publishing. https://doi.org/10.1007/978-3-319-33515-5_11
- Olsson, H., & Sörensen, S. (2003). *Forskningsprosessen: Kvalitative og kvantitative perspektiver*. Gyldendal akademisk.
- Patel, R., & Davidson, B. (2011). *Forskningsmetodikens grunder: Att planera, genomföra och rapportera en undersökning* (4. ed.). Studentlitteratur AB.
- Qumer Gill, A., Loumish, A., Riyat, I., & Han, S. (2018). DevOps for information management systems. *VINE Journal of Information and Knowledge Management Systems*, 48(1), 122–139. <https://doi.org/10.1108/VJIKMS-02-2017-0007>

Scaled Agile Inc. (n.d.). *SAFe 5.0 Framework*. Scaled Agile Framework. Retrieved April 29, 2021, from <https://www.scaledagileframework.com/>

Scaled Agile Inc. (2021a). *Agile Release Train*. Scaled Agile Framework. <https://www.scaledagileframework.com/agile-release-train/>

Scaled Agile Inc. (2021b). *Continuous Delivery Pipeline*. Scaled Agile Framework. <https://www.scaledagileframework.com/continuous-delivery-pipeline/>

Scaled Agile Inc. (2021c). *System and Solution Architect/Engineering*. Scaled Agile Framework. <https://www.scaledagileframework.com/system-and-solution-architect-engineering/>

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*.

Sill, A. (2014). Cloud Standards and the Spectrum of Development. *IEEE Cloud Computing*, 1(3), 15–19. <https://doi.org/10.1109/MCC.2014.50>

Sohlberg, P., & Sohlberg, B.-M. (2013). *Kunskapens former: Vetenskapsteori och forskningsmetod* (3. ed.). Liber.

Tonnquist, B. (2018). *Project Management* (4. ed.). Sanoma Utbildning.

Vetenskapsrådet. (2002). *Forskningsetiska principer inom humanistisk-samhällsvetenskaplig forskning*. Vetenskapsrådet.

Wiedemann, A. (2018). IT Governance Mechanisms for DevOps Teams—How Incumbent Companies Achieve Competitive Advantages. *Proceedings of the 51st Hawaii International Conference on System Sciences*, 4931–4940. <https://doi.org/10.24251/HICSS.2018.617>

Yin, L., & Filkov, V. (2020). Team discussions and dynamics during DevOps tool adoptions in OSS projects. *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 697–708. <https://doi.org/10.1145/3324884.3416640>

Yin, R. K. (2018). *Case study research and applications: Design and methods* (6. ed.). SAGE Publications, Inc.

Bilagor

Bilaga 1 - Missivbrev

Missivbrev – Intervju



Information om deltagande i intervju

Vi är två studenter som går högskoleingenjörsprogrammet inom industriell ekonomi på högskolan i Gävle. Vi håller just nu på med vårt examensarbete och är glada över att få studera er team. Syftet med vårt arbete är att undersöka implementeringen av DevOps. Vi vill därför genomföra en intervjustudie för att få ökad förståelse kring de erfarenheter ni har haft i arbetet med DevOps.

Intervjuerna genomförs via Microsoft Teams och beräknas ta 30-60 minuter. Det är frivilligt att delta och deltagandet kan närsomhelst avbrytas. Intervjuerna kommer att spelas in för transkribering men det raderas efter arbetet avslutats. Intervjuresultatet behandlas konfidentiellt och inga namn kommer nämnas. Informationen från intervjuerna kommer användas i examensarbetet och sammanställas i vår kandidatuppsats.

Vid frågor kontakta vänligen någon av nedanstående:

Lilly Carlsson
Student
Lilly@deepniner.net

Jeanette Langsager
Student
Jeanettelangsager@hotmail.com

Carina Dahlberg
Handledare
Carina.Dahlberg@hig.se

Bilaga 2 – Intervjufrågor personal på fallföretaget

1. Vad är din befattning?
2. Hur länge har du arbetat på företaget?
3. Hur har du varit delaktig i implementeringen av DevOps?
4. Hur fick du information om DevOps som koncept?
5. Vilken information fick ni om hur DevOps implementeringen skulle gå till?
6. Hur upplever du att kommunikationen har fungerat under implementeringen av DevOps?
7. Hur gick implementeringen av DevOps till?
8. Hur har din arbetsdag förändrats efter implementeringen av DevOps?
9. Vilka faktorer tror du har påverkat implementeringen av DevOps mest?
10. Vilka fördelar anser du att DevOps har medfört för företaget?
11. Vilka nackdelar anser du att DevOps har medfört för företaget?
12. Med det du vet i dag, på vilket sätt hade du förändrat implementeringen av DevOps?
13. Har du några förslag på hur ert nuvarande arbete med DevOps skulle kunna förbättras?
14. Finns det något mer du vill tillägga?
15. Är det okej om vi kontaktar dig om vi behöver ytterligare information?

Bilaga 3 – Intervjufrågor kundrepresentant

1. Vad är din befattning?
2. Hur länge har du arbetat med fallföretaget?
3. Vad levererar fallföretaget till er som kund?
4. Hur ser ditt samarbete med fallföretaget ut?
5. Hur har du varit delaktig i teamets implementering av DevOps?
6. Hur fick du information om DevOps som koncept?
7. Vilken information fick ni om hur DevOps implementeringen skulle gå till?
8. Hur upplever du att kommunikationen har fungerat under implementeringen av DevOps?
9. Hur gick implementeringen av DevOps till?
10. Hur har din arbetsdag förändrats efter implementeringen av DevOps?
11. Vilka faktorer tror du har påverkat implementeringen av DevOps mest?
12. Hur har implementeringen av DevOps påverkat företagets arbete med er som kund?
13. Med det du vet i dag, på vilket sätt hade du förändrat implementeringen av DevOps?
14. Har du några förslag på hur det nuvarande arbetet med DevOps skulle kunna förbättras?
15. Finns det något mer du vill tillägga?
16. Är det okej om vi kontaktar dig om vi behöver ytterligare information?