



LUNDS UNIVERSITET

Ekonomihögskolan

Institutionen för informatik

Transformellt ledarskap i DevOps-team

Ett kollektivt ansvar mot en förbättrad slutprodukt

Kandidatuppsats 15 hp, kurs SYSK16 i Informatik

Författare: Eric Magnusson
Julia Trulsson

Handledare: Magnus Wärja

Rättande lärare: Bo Andersson
Ahmad Ghazawneh

Transformellt ledarskap i DevOps-team: Ett kollektivt ansvar mot en förbättrad slutprodukt

ENGELSK TITEL: Transformational leadership in DevOps teams: A collective responsibility towards an improved end product

FÖRFATTARE: Eric Magnusson och Julia Trulsson

UTGIVARE: Institutionen för informatik, Ekonomihögskolan, Lunds universitet

EXAMINATOR: Odd Steen, Docent, Fil Dr

FRAMLAGD: maj, 2019

DOKUMENTTYP: Kandidatuppsats

ANTAL SIDOR: 77

NYCKELORD: DevOps, Transformellt ledarskap, DevOps-team.

SAMMANFATTNING (MAX. 200 ORD):

Slutanvändare förväntar sig idag snabba och kontinuerliga förändringar. De traditionella avdelningarna, utveckling och drift, har länge varit uppdelade vilket har bidragit till att flaskhalsar har uppstått vid utrullning av nya funktioner. DevOps myntades och kan ses som en förlängning av agila utvecklingsmetoder för att främja samarbetet mellan dessa två och därmed optimera hela mjukvarulivscykeln. Litteratur har uppmärksammat transformellt ledarskap som en viktig del för ett lyckat DevOps-arbete, men fokuserar inte vidare hur detta ledarskap bedrivs i kontext för DevOps-team, vilket denna studie ämnar åt att undersöka. En kvalitativ studie av fem intervjupersoner utfördes med syfte att samla in empiriskt material utifrån en framtagen undersökningsmodell. Denna modell utgjordes av fyra huvudområden: Vision, Kommunikation, Innovativt och lösningsorienterat arbete, samt Individuell omtanke, med respektive ingående aspekter. Studien påvisade att det transformella ledarskapet bedrivs som en kollektiv ansats i DevOps-team där ansvaret är utspritt på flera olika nivåer.

Innehåll

1	Inledning	1
1.1	Bakgrund	1
1.2	Problemområde	1
1.3	Forskningsfråga	2
1.4	Syfte	2
1.5	Avgränsningar	2
2	Litteratur	3
2.1	DevOps	3
2.1.1	Bakgrund	3
2.1.2	Olika perspektiv om DevOps	4
2.1.3	Summering av DevOps	5
2.2	Transformellt ledarskap	5
2.2.1	Dimensioner av transformellt ledarskap	6
2.2.2	Summering av transformellt ledarskap	7
2.3	Transformellt ledarskap i en DevOps-kontext	8
2.3.1	Vision	8
2.3.2	Kommunikation	9
2.3.3	Innovativt och lösningsorienterat arbete	9
2.3.4	Individuell omtanke	10
2.3.5	Undersökningsmodell	11
3	Metod	12
3.1	Metodval	12
3.2	Urval	12
3.3	Intervjuer	13
3.3.1	Intervjuguide	13
3.3.2	Bearbetning av data	14
3.4	Validitet, reliabilitet och etik	15
3.4.1	Validitet	15
3.4.2	Reliabilitet	16
3.4.3	Etik	16
4	Resultat	17
4.1	Vision	17

4.1.1	Mål	17
4.1.2	Mätning	18
4.2	Kommunikation.....	19
4.2.1	Gränsöverskridande kommunikation	19
4.2.2	Kontinuerlig återkoppling	20
4.2.3	Delning av kunskap och verktyg	21
4.3	Innovativt och lösningsorienterat arbete	22
4.3.1	Innovativt och kreativt arbetssätt	22
4.3.2	Lösningsorienterat arbetssätt.....	23
4.4	Individuell omtanke.....	24
4.4.1	Förtroendekultur	24
4.4.2	Agera coach.....	25
4.4.3	Minska beroenden	26
5	Diskussion.....	27
5.1	Vision	27
5.1.1	Mål	27
5.1.2	Mätning	27
5.2	Kommunikation.....	28
5.2.1	Gränsöverskridande kommunikation	28
5.2.2	Kontinuerlig återkoppling	29
5.2.3	Delning av kunskap och verktyg	29
5.3	Innovativt och lösningsorienterat arbete	30
5.3.1	Innovativt och kreativt arbetssätt	30
5.3.2	Lösningsorienterat arbetssätt.....	31
5.4	Individuell omtanke.....	31
5.4.1	Förtroendekultur	31
5.4.2	Agera coach.....	32
5.4.3	Minska beroenden	33
6	Slutsats	34
6.1	Vidare forskning.....	35
7	Bilagor.....	36
B1	– Svensk intervjuguide.....	36
B2	– Engelsk intervjuguide	38
B3	– Transkriberingsprotokoll P1	40
B4	– Transkriberingsprotokoll P2	47
B5	– Transkriberingsprotokoll P3	54
B6	– Transkriberingsprotokoll P4	59

B7 – Transkriberingsprotokoll P5	66
Referenser	76

Tabeller

Tabell 2.1: Undersökningsmodell	11
Tabell 3.1: Översikt över intervjuobjekt samt relaterad information	13
Tabell 3.2: Översikt för färgkodning och ID-taggar	15
Tabell 3.3: Exempel på användning av färgkodning och ID-taggar	15
Tabell 4.1: Översikt av hämtade data för Vision	17
Tabell 4.2: Översikt av hämtade data för Kommunikation	19
Tabell 4.3: Översikt av hämtade data för Innovativt och lösningsorienterat arbete	22
Tabell 4.4: Översikt av hämtade data för Individuell omtanke	24

1 Inledning

1.1 Bakgrund

Organisationer som är verksamma inom IT står inför en gemensam utmaning då marknaden idag förväntar sig snabba förändringar och nya funktioner (Humble & Molesky, 2011; Dyck et al., 2015). Projekt har på ett traditionellt sätt levererats men samtidigt avslutats då systemet överlämnats till driftpersonalen. Genom detta har utvecklare aldrig behövt köra sina system och sålunda förstår inte kompromisserna i skapandet av system som är pålitliga, skalbara och av hög kvalitet (Humble & Molesky, 2011).

Aktiviteter för utveckling och drift har länge skett via separata avdelningar, där var avdelning har enskilt uppsatta mål och riktlinjer. Utvecklare vill ständigt förbättra och skapa nya funktioner, medan driftpersonal vill bibehålla en stabil och pålitlig infrastruktur. Detta har resulterat konflikter mellan avdelningarna, där var part skyller på varandra när problem uppstår (Hüttermann, 2012). Detta har följaktligen skapat en märkbar flaskhals då det har funnits mindre tid att driftsätta fler ändringar, vilket har gjort det agila arbetssättet mindre effektivt för utvecklarna (de França et al., 2016; Hüttermann, 2012).

DevOps, en sammansättning av Development och Operations, myntades under DevOpsDays 2009 i Ghent, Belgien (Kim et al., 2016). DevOps kan ses som en förlängning av den agila mjukvaruutvecklingen, där DevOps istället ser till hela mjukvarulivscykeln med syfte att skapa ett närmare samarbete mellan utvecklarna och driftpersonalen (Armenise, 2015). Kim et al. (2016) menar att ett förbättrat samarbete mellan avdelningarna kan resultera i ett snabbare flöde, och därmed minska ledtid för användare. Därtill, ökar kvalitetsarbetet och infrastrukturen blir mer stabil och pålitlig. Syftet är att säkerställa att förändringarna är redo för driftsättning och kan addera värde för användaren direkt (Kim, 2013).

1.2 Problemområde

Arbetet har traditionellt bedrivits för separata avdelningar där enskilda mål och visioner varit uppsatta (Hüttermann, 2012). För att utvecklare och driftpersonal skall kunna arbeta gränsöverskridande krävs det att de delar vision samt arbetar mot gemensamma mål vilket kan åstadkommas om driftpersonalen är delaktiga i utvecklingen och vice versa med hjälp av kontinuerlig kommunikation (Humble & Molesky, 2011).

För att kunna arbeta gränsöverskridande krävs ett ledarskap vilket, enligt Forsgren et al. (2018), är ett av de mest förbisedda ämnena inom just DevOps. En ledare har auktoriteten att göra de stora förändringar som krävs. Det är även ledarens uppgift att ändra incitament inom rollerna, oavsett om det är inom utveckling, kvalitetssäkring, drift eller säkerhet för att eftersträva en gemensam kultur och vision. De påpekar även att ett transformellt ledarskap behövs och att det finns tydliga samband mellan kännetecknen för transformellt ledarskap och prestanda för att leverera mjukvara. (Forsgren et al., 2018) Vidare menar van Kelle et al.

(2015) att transformellt ledarskap är en viktig faktor för att arbeta utifrån agila arbetsmetoder, samt att det är bäst lämpat då fokus ligger på människor och interaktioner.

Transformellt ledarskap handlar om att inspirera och motivera medarbetare, vilket kan leda till positiva effekter för såväl resultat som individuella aspekter (Bass & Riggio, 2006). Samma ledare kan påverka förmågan att leverera bra kod, hålla en bra systemarkitektur och tillämpa principer om hur de arbetar, utvecklar samt driftsätter produkten (Forsgren et al., 2018). Likt Forsgren et al. (2018), har vi insett att ledarskap i team som arbetar med DevOps är ett förbisett ämne, trots att litteratur understryker vikten av det.

Tidigare litteratur har uppmärksammat transformellt ledarskap inom agil mjukvaruutveckling, men inte valt att fokusera på hur det bedrivs i en DevOps-kontext. Detta kunskapsgap har mynnat ut till vår forskningsfråga för studien.

1.3 Forskningsfråga

Hur bedrivs transformellt ledarskap i team som arbetar med DevOps?

1.4 Syfte

Problemområdet påvisar en ovisshet om hur det transformella ledarskapet bedrivs i DevOps-team, trots att det har uppmärksammat som en viktig faktor i tidigare litteratur. Studien ämnar således åt att undersöka hur transformellt ledarskap bedrivs i team som tillämpar DevOps.

1.5 Avgränsningar

Vi avgränsar oss i denna studie från att ta upp för- och nackdelar med att tillämpa DevOps, samt vilka risker DevOps kan medföra. Vidare tar vi inte i beaktning huruvida storleken på team har någon påverkan för resultatet.

2 Litteratur

I detta kapitel kommer relevant teori behandlas för att genomföra vår studie. Kapitlet är uppdelat i följande delkapitel; DevOps, Transformellt ledarskap, samt Transformellt ledarskap i en DevOps-kontext. De två första avser till att skapa förståelse för vad DevOps och Transformellt ledarskap är. Det sista delkapitlet kommer att ligga till grund för vår undersökningsmodell.

2.1 DevOps

2.1.1 Bakgrund

Under en tid har det varit en markant separation mellan utveckling och driftsättning, där var avdelning varit specialister på sina egna verktyg och ansvarat för en egen fas i livscykeln. Genom detta har det skapats flaskhalsar då de olika faserna krävt manuella steg och godkännande innan den har skickats vidare till nästa (Armenise, 2015). För att skapa en mer effektiv mjukvarulivscykel blev under 90-talet kontinuerlig integration ett populärt tillvägagångssätt mot detta (Ståhl & Bosch, 2013). Detta var ett första steg mot att arbeta mer effektivt med hjälp av delade plattformar som samlade alla kodändringar till en samlad kod (Armenise, 2015). Dock menar Humble & Farley (2010) att kontinuerlig integration enbart fokuserar på utvecklingen vilket skapar flaskhalsar då ändringar som utvecklare gör aldrig testas i en produktionsmiljö vilket förlänger tiden för återkoppling från driftpersonal.

Organisationer började med ett tvärfunktionellt samarbete mellan utveckling och drift, med andra ord *DevOps*, för att försöka eliminera denna flaskhals (Armenise, 2015). Samarbete har utvecklat kontinuerlig integration till kontinuerlig leverans vilket reducerar tiden för var iteration. Detta medför snabbare återkoppling från användare, reducerade risker och kostnader för driftsättning, en bättre inblick i leveransprocessen, samt effektiv hantering av risker (Humble & Molesky, 2011). Kontinuerlig integration och leverans har bidragit till att kontinuerlig driftsättning har börjat användas med syfte att sätta nya ändringar i produktion så snabbt som möjligt för att generera värde åt användaren på ett pålitligt sätt (Rodríguez et al., 2017). Ändringarna skall genomgå en process av flera enhetstester för att säkerställa lyckad driftsättning (Gmeiner et al., 2015; Humble & Farley, 2010). Leverans och driftsättning går hand i hand då kontinuerlig driftsättning inte är möjlig utan kontinuerliga leveranser. För att säkerställa att ändringar upprätthåller den kvalitet som krävs introducerade Humble & Farley (2010) en leveranspipeline där var ändring genomgår en process av enhetstester som utförs av diverse verktyg innan den sätts i produktionsmiljön. En sådan pipeline eliminerar opassande ändringar i ett tidigt skede, samt genererar återkoppling till ansvariga.

2.1.2 Olika perspektiv om DevOps

Begreppet DevOps förekommer ofta, dock med olika definitioner och perspektiv (Hüttermann, 2012; Roche, 2013). Hüttermann (2012) beskriver att syftet med DevOps är att förbättra samarbetet mellan utvecklings- samt driftpersonal med hjälp av delade mål, processer och verktyg för att organisationer skall producera högkvalitativ mjukvara. För att uppnå detta krävs en bra kultur, automation, mätning och delning. Bang et al. (2013) samt Erich et al. (2014) nämner även dessa fyra grundpelare för ett förbättrat samarbete mellan avdelningarna, oavsett roll. Dessa grundpelare har bildat akronymet CAMS (Culture, Automation, Measurement, Sharing) som först upprättades av Damon Edwards och John Willis. CAMS har kommit att bli en väletablerad modell för de grundläggande principerna inom just DevOps (Wickett, 2016). CAMS vidareutvecklades senare av pionjären Jez Humble som tillade 'Lean' och expanderade därmed akronymet till CALMS (Beyer et al., 2018). Den sistnämnda versionen, CALMS, beskrivs mer ingående nedan.

Kultur – För att utvecklare och driftpersonal skall kunna arbeta gränsöverskridande krävs det ett tidigt samarbete mellan avdelningarna, med regelbunden kommunikation och uppföljningsmöten (Humble & Molesky, 2011). Ravichandran et al. (2016) menar att en fungerande DevOps-kultur skall handla om att skapa delade värderingar och beteenden som genomsyrar hela organisationen. Detta kräver öppen kommunikation, kontinuerlig återkoppling, och gemensamma måttal. Med innovation som ett centralt begrepp för DevOps, förespråkar Kim et al. (2016) att uppmuntra risktagande för att medarbetare skall våga uppmärksamma problem istället för att undanhålla dem. För att kunna arbeta gränsöverskridande och främja ett innovativt arbetssätt kräver DevOps en stark förtroendekultur (Walls, 2013; Davis & Daniels, 2016).

Automation – Att gå från manuella kontroller till automation för att uppnå högre effektivitet i processen genom kontinuerlig integration, leverans samt driftsättning (Farenhorst et al., 2018). Genom automation kan även återkoppling genereras snabbt vilket Hüttermann (2012) anser vara en de viktigaste faktorerna.

Lean – Fundamentala Lean-principer ligger till grund för DevOps, med primärt fokus att skapa värde åt kunden. Genom att eliminera waste från värdekedjan kan förmågan att generera värde ständigt förbättras (Farenhorst et al., 2018). Vidare tillägger Kim et al. (2016) att flödet bör optimeras genom hela värdekedjan för att kunna leverera värde till kunden så snabbt och effektivt som möjligt. Utöver eliminering av waste, optimeras flödet genom att synliggöra arbete, minska storleken på iterationer, samt implementera kvalitetssäkring i det dagliga arbetet (Kim et al., 2016).

Mätning – Mätning måste vara öppen, transparent samt lättåtkomlig för visualisering inom alla beståndsdelar för DevOps (Perera et al., 2017). Mätning kan ske på olika nivåer beroende på om det är ett större alternativt mindre mål (Humble & Molesky, 2011) och det är enligt Hüttermann (2012) mätningen som visar nuvarande status och om det är något som behöver åtgärdas.

Delning – Hüttermann (2012) pekar på delning som ett centralt begrepp för DevOps där idéer, mål, problem, processer, och verktyg delas genom hela organisationen, vilket Perera et al. (2017) instämmer och hävdar är grundläggande för ett lyckat samarbete. Kim et al. (2016) poängterar även att ett snabbt och kontinuerligt flöde av återkoppling över hela värdekedjan gör det möjligt att skapa ett arbetssystem av hög säkerhet och resiliens. Kontinuerlig

återkoppling gör det möjligt att identifiera och åtgärda problem innan de blir allt för komplexa och kostsamma, vilket vidare bidrar till förbättrad slutprodukt.

Även Lwakatare et al. (2015) beskriver DevOps inom fyra element som skall representera dess grundpelare. Olikt från Hüttermanns (2012) samt Bang et al. (2013) menar Lwakatare et al. (2015) att samarbete och övervakning, istället för kultur och delning, tillsammans med automation och mätning är elementen som adresserar problemen som DevOps huvudsakligen skall lösa.

Dyck et al. (2015) är dock inte eniga med Hüttermanns definition då den saknar incitament till varför detta samarbete skall förbättras. De själva menar att DevOps är ett organisatoriskt antågande för att uppnå strategiska mål med hjälp av tvärfunktionellt samarbete både inom, men också mellan team, speciellt inom utveckling och drift. Hüttermann (2012) menar vidare på att delade mål och incitament kan inom olika avdelningar vara en naturlig konflikt och vara svåra att uppnå och skall därför helst ligga i linje med varandra. Han nämner också att DevOps inte kan ses som en egen avdelning eller en arbetsroll utan snarare ett sätt att hjälpa organisationer.

2.1.3 Summering av DevOps

Sammanfattningsvis ser vi att granskad litteratur över DevOps bildar en diffus definition då författare ser det från olika perspektiv. Dock finns tydliga gemensamma grundläggande principer som förekommer från flera författare. Akronymet CALMS, som nämnt i föregående sektion, är ett väletablerat och repetitivt förekommande perspektiv på grundelement för DevOps (Hüttermann 2012; Erich et al., 2014; Bang et al., 2013). DevOps präglas av delade värderingar som genomsyrar hela organisationen för att skapa en god kultur (Ravichandran et al. 2016). Detta för att avdelningarna skall kunna bedriva ett gränsöverskridande samarbete där allt delas med varandra (Humble & Molesky, 2011; Hüttermann, 2012). Hüttermann (2012) och Perera et al. (2017) menar därtill att mätning skall bedrivas öppet och övervakas för att ge indikation på nuvarande situation. Automation genererar mervärde och bidrar till en mer effektiv organisation genom att eliminera onödiga manuella kontroller så att hela värdekedjan optimeras (Farenhorst et al., 2018; Kim et al., 2016).

Vidare hävdar Hüttermann (2012) att DevOps skall ses som ett organisatoriskt antågande med syfte att förlänga den traditionella agila arbetsmetodiken för att se över hela mjukvarulivscykeln som helhet istället för olika faser som Armenise (2015) beskriver. Med utgångspunkt i den teori som behandlats för DevOps olika perspektiv har vi utarbetat följande definition som kommer ligga till grund för denna studie:

DevOps är ett organisatoriskt antågande med syfte att skapa ett nära samarbete mellan utvecklings- och driftpersonal för att uppnå strategiska mål och reducera ledtider genom kontinuerlig förändring, detta med hjälp av en samarbetsvillig kultur, automation, mätning, övervakning och delning.

2.2 Transformellt ledarskap

Forsgren et al. (2018) hävdar att ledarskap har en stor inverkan på resultatet. Ledarskap skall handla om att inspirera och motivera medarbetare, där en god ledare kan påverka teamets

förmåga att leverera kod, utforma system, samt applicera Lean-principer på ledning- och utvecklingsarbetet. Enligt författarna har samtliga faktorer en betydelsefull inverkan på organisationens lönsamhet, produktivitet, och marknadsandel. Faktorerna har därtill även inflytande över kundnöjdhet, effektivitet, samt förmågan att uppnå verksamhetsmål.

I en studie utförd av van Kelle et al. (2015), konkretiserades kritiska framgångsfaktorer för agila projekt, där resultatet påvisade det transformella ledarskapet som en av de viktigaste faktorerna. Studien visade på att primärt fokus skall ligga på delade värderingar gällande agila projekt, vilket syftar till mål, vision och prioriteringar. Att upprätthålla en transformell ledarskapsstil, i kombination med kontinuerlig kommunikation, kan bidra till att skapa delade värderingar genom hela organisationen (van Kelle et al., 2015). Yukl & Kaulio (2011) beskriver att transformella ledare vill skapa en ökad medvetenhet bland medarbetarna om arbetets värde och betydelse. En sådan medvetenhet kan få medarbetarna att prioritera organisationens intresse framför sitt eget. Även Yukl & Kaulio (2011) fann i en studie att det transformella ledarskapet är väsentligt för att leda team, då ledarskapet innefattar beteenden som fokuserar på såväl relationer som förändring.

En transformell ledare skall, enligt Bass & Riggio (2006), motivera och inspirera sina medarbetare. Till följd kan det påverka dem att uppnå bättre resultat, och därtill även utveckla individuella ledarskapsförmågor. Vidare kan det möjliggöra ett mer kreativt tankesätt, samt förmågan att arbeta mer innovativt med lösningar vid problem som uppstår. Walumbwa et al. (2004) instämmer och menar att ett mer djupgående tänkande kan understödja medarbetare att skapa förståelse för vad som krävs för att lyckas. Ett sådant resonemang kan leda till ökat förtroende och därmed bidra till ett större engagemang i organisationen. Fortsättningsvis betonar Yukl & Kaulio (2011) vikten av kompetensutveckling och ökat självförtroende bland medarbetarna, då de följaktligen kan våga ta större ansvar i organisationen. Således blir den transformella ledarens roll att agera coach genom att erbjuda medarbetarna stöd och uppmuntran vid motgångar. En sådan roll skapar respekt och tillit gentemot ledaren, och kan fördelaktigt resultera i att medarbetare motiverar sig att prestera över förväntan. Bass (1985) förstärker detta och menar att transformella ledare kan motivera medarbetare att öka prestationsförmågan genom att förändra deras attityder, uppfattningar, och värderingar. Starka transformella ledare skapar och stödjer hälsosamma team och organisationskultur (Rafferty & Griffin, 2004).

2.2.1 Dimensioner av transformellt ledarskap

Som ett resultat av brist på empiriskt stöd, utförde Rafferty & Griffin (2004) en studie där de identifierade fem dimensioner av transformellt ledarskap. Dimensionerna stöds av andra författare inom området, och beskrivs mer ingående nedan.

Visionärt ledarskap - Transformella ledare har en tydlig bild över vart organisationen är på väg och var den kommer befinna sig om ett par år. Vision är en idealiserad bild av framtiden som bygger på organisatoriska värden. Transformella ledare skall agera som förebilder och har ett beteende som överensstämmer med visionen (Rafferty & Griffin, 2004). Van Kelle et al. (2015) instämmer i sin studie, och menar att en ledare som upprättar en gemensam vision, målbild och prioriteringar är grundläggande för framgång.

Inspirerande kommunikation - Transformella ledare kommunicerar på ett sätt som inspirerar och motiverar medarbetarna, även i en osäker eller föränderlig miljö. Kommunikationen skall vara positiv och uppmuntrande, vilket bygger motivation och självförtroende bland

medarbetarna (Rafferty & Griffin, 2004). Enligt Bass & Riggio (2006) kan motivation och inspiration även generera bättre arbetsprestationer, och medarbetare tenderar att utveckla individuella ledarskapsförmågor.

Intellektuell stimulans - Transformella ledare stimulerar medarbetare att arbeta innovativt och kreativt genom att ifrågasätta antaganden, öka medvetenhet om problem, samt betrakta gamla situationer ur nya perspektiv. Medarbetare uppmanas att arbeta kreativt och lösningsorienterat för att enklare kunna identifiera problem och vidare förbättra arbetet. Förslag på nya idéer och tillvägagångssätt uppmuntras, och misslyckanden skall inte kritiseras (Rafferty & Griffin, 2004). Ett innovativt arbetssätt och ett ökat självförtroende kan skapa större engagemang och ansvarstagande i organisationen (Walumbwa et al., 2004; Yukl & Kaulio, 2011). Forsgren et al. (2018) menar även att transformellt ledarskap möjliggör en kultur där kontinuerlig innovation och lärande är en stor del av allas dagliga arbete.

Stödjande ledarskap - Transformella ledare uttrycker omtanke för medarbetare och tar hänsyn till medarbetares individuella behov. Ledaren är en god lyssnare som understödjer individuella skiljaktigheter genom att agera som stödjande och uppmuntrande coach. En stödjande ledare ger medarbetare utvecklingsmöjligheter genom att förse dem med utmanande arbetsuppgifter (Rafferty & Griffin, 2004). Yukl & Kaulio (2011) hävdar att agera coach hjälper ledaren att vinna tillit och respekt, vilket har påvisats frambringa ökad motivation hos medarbetarna som därtill presterar över förväntan.

Personlig bekräftelse - Transformella ledare uppmärksammar och värderar var enskild medarbetares prestation, samt berömmar och belönar uppnående av resultat som är förenliga med visionen (Rafferty & Griffin, 2004). Avolio & Bass (1995) syftar även på att sådan bekräftelse kan ske i form av återkoppling; att individuellt rikta sig till den person som förväntats fullfölja en uppgift. Var enskild medarbetare skall ha möjlighet att lära sig av framgång så väl som misstag.

2.2.2 Summering av transformellt ledarskap

Bass & Riggio (2006) beskriver att en transformell ledare skall motivera och inspirera sina medarbetare för att skapa delade värderingar inom team och organisation. Detta görs genom fem dimensioner: visionärt ledarskap, en inspirerande kommunikation, intellektuell stimulans, stödjande ledarskap och personlig bekräftelse (Rafferty & Griffin, 2004). Visionärt ledarskap syftar till att skapa en gemensam målbild som delas över hela organisationen.

Kommunikation skall ske på ett inspirerande sätt som motiverar medarbetarna att förbättra prestationsförmågan (Rafferty & Griffin, 2004; Bass & Riggio, 2006). Intellektuell stimulans sker genom att förespråka ett innovativt och lösningsorienterat arbetssätt, vilket skapar ett större engagemang inom organisationen (Walumbwa et al., 2004; Yukl & Kaulio, 2011). Ett stödjande ledarskap bedrivs genom att vara en god lyssnare, samt genom att uppmuntra och vägleda medarbetare (Rafferty & Griffin, 2004; Yukl & Kaulio 2011). Personlig bekräftelse sker genom att uppmärksamma och bekräfta arbetsprestationer för var enskild individ (Rafferty & Griffin, 2004).

Aspekterna kan vidare bidra till positiva effekter som att uppnå bättre resultat, individuella ledarskapsförmågor, kreativitet inom problemlösning och ökat självförtroende (Bass & Riggio, 2006). Transformellt ledarskap är något som skall bedrivas i organisationer som tillämpar agila utvecklingsmetoder som DevOps enligt Forsgren et al. (2018) och van Kelle et

al. (2015). Yukl & Kaulio (2011) belyser även vikten av transformellt ledarskap för team som genomgår förändring.

2.3 Transformellt ledarskap i en DevOps-kontext

Forsgren et al. (2018) menar att ett engagerat ledarskap är grundläggande för en lyckad DevOps transformation. Vidare lyfter författarna att transformellt ledarskap är nödvändigt för att kunna (1) etablera och understödja generativa och högt tillförlitliga kulturella normer, (2) skapa teknologier som möjliggör utvecklarens produktivitet, minimera ledtider för driftsättning av kod, samt stödja mer pålitliga infrastrukturer, (3) förespråka innovation och experimenterande för att skapa och implementera bättre produkter snabbare, samt (4) arbeta gränsöverskridande över organisatoriska silos för att arbeta i linje med strategin. Samtliga faktorer är ett resultat av vad transformellt ledarskap kan medföra, vilket uttryckligen ligger i linje med DevOps vision.

Van Kelle et al. (2015) påpekar att det transformella ledarskapet speglar en adaptiv ledarskapsstil då mycket kretsar kring att motivera, inspirera, uttrycka visioner, samt engagemang. De menar också på att denna ledarskapsstil är mest optimal, då den fokuserar mycket på mänskliga faktorer och interaktioner. Att vara adaptiv är idag en nödvändighet då marknaden ständigt förändras och kräver att organisationer anpassar sig snabbt och ofta (Humble & Molesky, 2011; Dyck et al., 2015).

Nedan presenteras framtagna huvudområden som är påvisande för både transformellt ledarskap och DevOps. Huvudområdena kommer även att ligga till grund för studiens fortfarande.

2.3.1 Vision

Ravichandran et al. (2016) konstaterar att ett fungerande arbetssätt med DevOps kräver delade värderingar och beteenden som genomsyrar hela organisationen. Detta kräver öppen kommunikation, kontinuerlig återkoppling och gemensamma måttal. Van Kelle et al. (2015) instämmer och hävdar att delade värderingar blir enklare att uppnå med stöd av transformellt ledarskap och kommunikation. Utvecklare och driftpersonal tenderar att förbättra samarbetsförmågan och enklare arbeta mot gemensamma mål vid eliminering av flaskhalsar som minskar effektiviteten och produktiviteten i organisationen (Sharma, 2014). Vidare beskriver Sharma (2014) att det kan uppstå konflikter vid uppsättning av gemensamma mål, då var avdelning kan ha särskilda mål. Utvecklarna vill ständigt förändra medan driftpersonalen vill ha ett stabilt system. Genom att arbeta med en delad målbild skall DevOps, enligt Hüttermann (2012), fylla sitt huvudsakliga syfte: ett stärkt samarbete mellan utveckling och drift. Även van Kelle et al. (2015) påvisar i sin studie att en ledare som upprättar en gemensam vision och målbild är essentiellt för att nå framgång.

Mätning bör bedrivas gemensamt och delas över hela organisationen, istället för inom separata avdelningar (Farroha & Farroha, 2014). Humble & Molesky (2011) beskriver emellertid att mätning kan ske på olika nivåer i organisationen, där KPI:er (Key Performance Indicator) omsorgsfullt bör sättas utifrån teamets egen förmåga. Enligt Forsgren et al. (2018) är det felaktigt att mäta produktiviteten i antalet rader kod alternativt funktionaliteter. Det skall hellre ske i hur välmående systemet är utifrån antalet ändringar (Humble & Molesky,

2011). Genom att mäta och visualisera arbetet kan transformella ledare skapa en ökad medvetenhet och förståelse hos medarbetarna (Yukl & Kaulio, 2011). Även Humble & Molesky (2011) poängterar vikten av att visualisera arbetet, vilket gör att alla blir mer involverade genom hela arbetsprocessen.

2.3.2 Kommunikation

För att kunna arbeta gränsöverskridande behöver driftpersonalen vara involverade i utvecklingen samt att utvecklarna är delaktiga i driftsättningen med en regelbunden kommunikation och uppföljningsmöten. I en DevOps-kontext sker denna kommunikation bland annat i form av att meddela relevant driftpersonal att en ny funktionalitet är på ingång och att detta inte sker samma dag som release (Humble & Molesky, 2011). Walls (2013) belyser vikten av att följa och diskutera produkten gränsöverskridande genom hela livscykeln och att ett kollektivt fokus skall ligga på produkten som helhet, inte på specifika funktioner eller processer. Ravichandran et al. (2016) förstärker detta och menar vidare att öppen kommunikation bidrar till att enklare skapa en fungerande DevOps-kultur. Genom att kommunicera på ett inspirerande och motiverande sätt, menar Bass & Riggio (2006) att medarbetare ökar prestationsförmågan och har förmågan att utveckla individuella ledarskapsförmågor.

För att skapa ett snabbt kontinuerligt flöde för återkoppling krävs det att personal svärmar runt problem som uppstår. Personal samlas för att lösa problem omedelbart innan nytt arbete kan påbörjas. Återkoppling genereras snabbt för alla i hela värdekedjan, specifikt för personen som orsakade felet, genom att isolera problemet och motverka faktorer som dölja det och dess effekt. Om problem inte adresseras direkt, ökar risken att samma problem uppstår redan i nästa driftsättning (Kim et al., 2016). Vidare möjliggörs kontinuerlig återkoppling utifrån kontinuerliga leveranser vilket hjälper teamen att erhålla detta både internt och externt (Sharma, 2014). Kontinuerlig återkoppling gör det möjligt att identifiera och åtgärda problem innan de blir allt för komplexa och kostsamma, vilket bidrar till en förbättrad slutprodukt (Kim et al., 2016). Vidare menar Davis & Daniels (2016) att informationsdelning eliminerar flaskhalsar och minskar beroenden av såväl team som enskilda personer.

Att dela med sig av kunskap och verktyg är viktigt när man tillämpar DevOps. För ett lyckat samarbete krävs det att man delar med sig av idéer och problem (Perera et al., 2017). Vidare påpekar Rakowski (2015) vikten av att överföra kunskap som den viktigaste kärnan i DevOps. Att inte göra detta brister hela idén om att skapa rätt kultur, skapa acceptans inom automation, samt att ha mätinstrument som fungerar. Hüttermann (2012) instämmer att delning är fundamentalt för DevOps, där kunskap och verktyg delas över hela organisationen.

2.3.3 Innovativt och lösningsorienterat arbete

Genom motivation och inspiration kan transformella ledare stimulera sina medarbetare att arbeta mer innovativt och kreativt med problem som uppstår (Bass & Riggio, 2006). Forsgren et al. (2018) hävdar att transformella ledare möjliggör en kultur som innefattar innovation och lärande i det dagliga arbetet. Innovation är ett centralt begrepp för DevOps, men om organisationen inte accepterar misslyckanden riskerar medarbetare att bli begränsade i deras arbete till att hitta nya lösningar och effektivisera pågående arbetssätt. Misslyckanden kan vara värdefulla så länge organisationen lär sig av misstagen och förbättrar arbetet (Rakowski, 2015). Ravichandran et al. (2016) menar även att enskilda prestationer inte skall bära skuld

för uppkomna problem. Istället för att kritisera misslyckanden uppmanar transformella ledare att tänka kreativt och lösningsorienterat, vilket bidrar till att mer effektivt kunna identifiera och åtgärda problem som uppstår (Rafferty & Griffin, 2004).

Att uppmärksamma risktagande personal skall förespråkas enligt Kim et al. (2016), vilket resulterar i att de vågar lyfta och diskutera problem istället för att dölja dem. Organisationer måste se problemen för att kunna lösa dem. Genom att aktivt arbeta med problem som uppstår genereras lärdomar inom organisationen, en sorts lärandekultur. Detta för att göra det möjligt för organisationen att förhindra att problemet återkommer, samt att tidigare upptäcka och hantera liknande problem i framtiden (Kim et al., 2016; Farroha & Farroha, 2014).

Transformella ledare anser att medarbetare skall ägna sig åt ett mer djupgående tankesätt i det dagliga arbetet, vilket Walumbwa et al. (2004) och Yukl & Kaulio (2011) påstår skapar ökat förtroende, engagemang, och ansvarstagande hos medarbetarna.

2.3.4 Individuell omtanke

För att kunna arbeta gränsöverskridande och främja ett innovativt arbetssätt kräver DevOps en stark förtroendekultur (Walls, 2013; Davis & Daniels, 2016). Vid brist på tillit riskerar medarbetare att agera mer vaksamt över sina projekt och ansvarsområden, vilket begränsar möjligheten att dela information och kan därmed få en negativ inverkan på produktiviteten (Davis & Daniels, 2016). Förtroende i en DevOps-kultur syftar delvis till att medarbetare från driftavdelningen kan tillföra väsentliga reflektioner vid utveckling av programvara (Ravichandran et al., 2016). Därtill, bör driftarbetare förlita sig på att utvecklarna arbetar utifrån ett professionellt förhållningssätt och ser till vad som är mest fördelaktigt för produkten i fråga (Walls, 2013). Ravichandran et al. (2016) menar att en förtroendekultur således kräver kollektiv respekt, där allas arbete värdesätts och ingen känner rädsla för att göra sin röst hörd. Ett starkt transformellt ledarskap kan skapa denna typ av organisationskultur för att driva och stötta team (Rafferty & Griffin, 2004). DevOps-team avstår från att peka finger och skylla ifrån sig, utan förespråkar istället en kultur där hela teamet bär ansvar för såväl framgång som misslyckande (Farroha & Farroha, 2014).

Hussain et al. (2017) påvisade i sin studie att ledarskapsattribut aktivt söks för DevOps-roller. Baserat på insamlade data hamnade stort fokus på transformellt ledarskap och roller som mentor, medlare, facilitator och innovatör. För en sådan roll var arbetsgivarnas förväntningar att ta ett övergripande ansvar för teamet genom utbildning och mentorskap. Yukl & Kaulio (2011) hävdar att en transformell ledare kan öka tillit genom att betraktas som en coach i form av ett stödjande och uppmuntrande arbetssätt. En sådan ledare kan säkerställa att teamet följer planerat arbete, agera mentor samt förespråka vikten av ett samarbetsvilligt team (Wiedemann, 2018).

Davis & Daniels (2016) menar att en god ledare är någon som kan frambringa det bästa från sina medarbetare, istället för att förlita sig på ett fåtal specialister i teamet. Detta kan vara direkt avgörande för att bedriva en effektiv organisation. Farroha & Farroha (2014) instämmer att man skall undvika en beroende-mentalitet och istället uppmärksamma teamet som helhet.

2.3.5 Undersökningsmodell

Följande undersökningsmodell kommer ligga till grund för insamling av empiriska data och baseras på redovisade huvudområden för hur transformellt ledarskap bedrivs i team som tillämpar DevOps. Vi har identifierat flera ingående aspekter för var huvudområde som är tydligt presenterade i undersökningsmodellen (Se Tabell 2.1). Syftet med aspekterna är att vidare kunna urskilja, sortera, och analysera insamlade data.

Tabell 2.1: Undersökningsmodell

Huvudområde	Aspekter	Litteratur
Vision	Gemensamma mål	Ravichandran et al. (2016); van Kelle et al. (2015); Sharma (2014); Hüttermann (2012); Farroha & Farroha (2014); Humble & Molesky (2011); Forsgren et al. (2018); Yukl & Kaulio (2011)
	Mätning	
Kommunikation	Gränsöverskridande kommunikation	Humble & Molesky (2011); Walls (2013); Ravichandran et al. (2016); Bass & Riggio (2006); Kim et al. (2016); Sharma (2014); Davis & Daniels (2016); Perera et al. (2017); Rakowski (2015); Hüttermann (2012)
	Kontinuerlig återkoppling	
	Delning av kunskap och verktyg	
Innovativt & lösningsorienterat arbete	Innovativt och kreativt arbetssätt	Bass & Riggio (2006); Forsgren et al. (2018); Rakowski (2015); Ravichandran et al. (2016); Rafferty & Griffin (2004); Kim et al. (2016); Farroha & Farroha (2014); Walumbwa et al. (2004); Yukl & Kaulio (2011)
	Lösningorienterat arbetssätt	
Individuell omtanke	Förtroendekultur	Walls (2013); Davis & Daniels (2016); Ravichandran et al. (2016); Rafferty & Griffin (2004); Farroha & Farroha (2014); Hussain et al. (2017); Yukl & Kaulio (2011); Wiedemann (2018)
	Agera coach	
	Minimera beroenden	

3 Metod

Under detta kapitel presenteras vald metod för att kunna utföra studien hur det transformella ledarskapet bedrivs i team som tillämpar DevOps. Därefter kommer tillvägagångssättet presenteras mer ingående, med beskrivning för arbetet innan, under, och efter samtliga intervjuer.

3.1 Metodval

Vi har i denna studie valt att använda oss av den kvalitativa ansatsen då vår frågeställning sker genom en explorativ ansats där vi önskar utforska verkligheten genom en djupare analys och då vi anser att det råder en lucka i tidigare forskning, vilket innebar att förkunskapen inte var så stor. Vi värdesatte att den kvalitativa metoden erbjöd större flexibilitet än den kvantitativa, då vi under studiens gång har kunnat anpassa metoden för datainsamling allt eftersom studien fortlöpte då insikter i intervjuer gav möjlighet för vidare analys (Jacobsen, 2002).

3.2 Urval

Ett urval styrs utifrån vilken slags information forskaren vill åt och tas fram utifrån syftet med forskningsfrågorna, vilka bör peka mot en speciell typ av urvalsgrupp (Jacobsen, 2002). Det har ställts krav på urvalet i studien då det ämnar till att besvara hur det transformella ledarskapet bedrivs i team som arbetar med DevOps. På grund av både tidsbegränsning och en finansiell aspekt valde vi informanter från organisationer som var tillgängliga för oss (Se Tabell 3.1), det vill säga ett så kallat bekvämlighetsurval (Jacobsen, 2002). Vidare beskriver Jacobsen (2002) att under kvalitativa metoder är det bättre att intervjua färre personer då dessa datainsamlingsmetoder kräver betydligt mer tid än kvantitativa metoder. Kvalitativa data är rika på detaljer och upplysningar, vilket bidrar till stora mängder data att analysera. Att hinna med att analysera all dessa data under en viss tidsram är orimligt och bidrar till att materialet inte är lika bearbetat. Vi har därför valt att intervjua fem personer inom radien för vårt kontaktnät som sades tillämpa DevOps i sin organisation där den kvalitativa metoden appliceras. Samtliga företag ansågs befinna sig inom ramen för uppsatsens syfte då de besitter erfarenhet kring studiens forskningsfråga.

Tabell 3.1: Översikt över intervjuobjekt samt relaterad information

Företag	Informant	Roll och relevant erfarenhet	Datum och längd	Metod
F1	P1	<i>DevOps Specialist</i> – Bakgrund mestadels inom operations, IT-chef och driftansvarig för IT	17-04-2019 – 1:29:45	Personligt möte
F1	P2	<i>DevOps</i> – Bakgrund som systemspecialist inom Linux och drift inom IT	17-04-2019 – 46:30	Personligt möte
F2	P3	<i>Ansvarig för Digital Transformation</i> – Bakgrund inom systemutveckling och projektledning	24-04-2019 – 23:50	Digital telefoni
F3	P4	<i>IT-konsult</i> – Bakgrund som utvecklare och har certifieringar inom DevOps	25-05-2019 – 51:42	Personligt möte
F4	P5	<i>Chapter Lead, DevOps Engineer</i> – Bakgrund inom både operations och utveckling	07-05-2019 – 53:55	Personligt möte

3.3 Intervjuer

För att få studiens intervjupersoner att känna sig avslappnade och bekväma under intervjutillfället önskade vi hålla majoriteten av intervjuerna på deras respektive arbetsplatser och därmed i intervjupersonernas naturliga miljö (Jacobsen, 2002). Intervjuerna skedde främst under personliga möten, men med det företag som låg utanför geografisk närhet valde vi att sköta per telefon då längre resor skulle vara både tids- och kostnadskrävande. Vi valde att låta intervjuobjekten bestämma var de personliga mötena skulle äga rum. Att hålla intervjun i deras naturliga miljö kan bidra till bättre svar till skillnad om den skulle vara obekant. Dock finns en risk med att utföra intervjuer i intervjupersonernas naturliga kontext, vilket är enligt Jacobsen (2002) att det kan förekomma företagsrelaterade störningsmoment, såsom att något avbryter konversationen.

Genom att använda sig av telefonintervjuer och öppna frågor kan det vara svårare att upprätta en personlig relation och det kan i sin tur öka risken att informanterna ger felaktig information (Jacobsen, 2002). Via telefonintervju blir det också svårt att observera i hur informanten beter sig, till skillnad från personliga möten, ansikte mot ansikte, där man enklare kan se kroppsmönster och om informanten känner sig besvärad under specifika frågor (Jacobsen, 2002). Innan varje intervju har informanterna blivit meddelade om vad intervjun har för syfte samt att de har rätten att bli anonyma om så varit fallet. Vidare användes två telefoner som inspelningsverktyg för att minimera risken att inspelat material går förlorat. Vi förde även anteckningar under intervjuerna med syfte att kunna ställa relevanta följdfrågor utan att avbryta intervjuobjektet.

3.3.1 Intervjuguide

Vi valde att använda oss av semi-strukturerade intervjuer, vilket enligt Jacobsen (2002) innebär att författarna upprättar en guide med relevanta områden som intervjun önskar belysa och därefter kan intervjuobjekten fritt tala om sina erfarenheter. Guiden i sig var utformad

efter relevant teori, transformellt ledarskap i kontext med DevOps. Vi värdesatte flexibla intervjuer då dessa kunde generera ny information kring transformellt ledarskap inom DevOps-team. Vår intervjuguide på svenska samt engelska, som finns under bilaga B1 och B2, är uppdelad i fem kategorier utefter undersökningsmodellens huvudområden (Se Tabell 2.1):

Bakgrund - Inledande frågor främst för att skapa en relation med intervjuobjektet.

Vision - Hur arbetet med gemensamma mål och hur mätning bedrivs.

Kommunikation - Hur kommunikationen fungerar gränsöverskridande, i form av återkoppling och delning av kunskap och verktyg.

Innovativt och lösningsorienterat arbetssätt - Hur innovativt och lösningsorienterat arbete, samt om risktagande uppmuntras.

Individuell omtanke - Hur förtroendekulturen ser ut för att minska beroenden av specifika individer eller team, samt hur arbetet stöds i form av coachning

Övrigt - Avslutande frågor för att få ett professionellt avslut på intervjun men även ge intervjuobjektet rum för att ta upp information som kan vara intressant för studien.

3.3.2 Bearbetning av data

Insamlade data bör vara väl överskådlig och lättförståelig, och vi valde således att strukturera och hantera intervjuresultaten utefter Jacobsens (2002) väletablerade tre steg: beskrivning, systematisering samt kategorisering och kombination.

Vi började med att eftersträva en så grundlig och detaljerad beskrivning av data som möjligt. Samtliga ljudinspelningar transkriberades och har redigerats samt översatts från talspråk till skriftspråk så detaljrikt vi ansåg som nödvändigt för att få en mer läsbar text som underlättade för oss författare att gå tillbaka till. Samtliga transkriberingar återfinns i kronologisk ordning utifrån urvalet (Se Tabell 3.1) under bilaga B3-B7.

Därefter följde vi nästa steg i Jacobsens tre steg - systematisering och kategorisering. Genom att överskåda den insamlade informationen, dela upp data i olika grupper för att belysa vilka ämnen de innehöll samt och reducera icke relevant data kunde vi skapa en överblick och genom det förmedla vad vi funnit (Jacobsen, 2002). Under systematiseringen färgmarkerades meningar utifrån respektive huvudområde. Vidare delades färgade meningar upp med hjälp av en eller flera ID-taggar utifrån huvudområdets ingående aspekter. Detta är för att skapa en bild av intervjuobjektens berättelser. Beskrivning av färgmarkering och ID-taggar finns i vår kodtabell (Se Tabell 3.2). Denna kodning är baserad utifrån vår undersökningsmodell (Se Tabell 2.1) där huvudområden och aspekter presenteras.

Tabell 3.2: Översikt för färgkodning och ID-taggar

Huvudområde	Färg	ID	Aspekter
Vision	Röd	V1	Gemensamma mål
		V2	Mätning
Kommunikation	Grön	K1	Gränsöverskridande kommunikation
		K2	Kontinuerlig återkoppling
		K3	Delning av kunskap och verktyg
Innovativt och lösningsorienterat arbete	Blå	A1	Innovativt och kreativt arbetssätt
		A2	Lösningsorienterat arbetssätt
Individuell omtanke	Gul	O1	Förtroendekultur
		O2	Agera coach
		O3	Minska beroenden

Nedan (Se Tabell 3.3) är ett exempel utifrån transkriberingsprotokollen där ett stycke är uppdelat i två huvudområden; Rött för vision samt blått för innovativt och lösningsorienterat arbete. Meningarna har vidare fått en identifikation i form av en ID-tagga beroende på vilken aspekt det berör i respektive huvudområde.

Tabell 3.3: Exempel på användning av färgkodning och ID-taggar

Rad:	Information:	Aspekt:
71	Arbetar ni aktivt för att upptäcka fel?	
72	Ja, absolut. Just när man gör manuella jobb att man ser någonting som inte stämmer. Eller att utvecklare hör av sig att "nja, det gick inte att deploya en container" "klustret skalar inte upp". Ja men då är det en config-fil någonstans.	A2, K1

Slutligen avslutade vi analysarbetet med att leta efter specifika meningar och orsaker samt försökte generalisera eller bringa viss ordning i data (Jacobsen, 2002). I denna avslutande fas jämförde vi insamlad data mot varandra samt mot olika teoretiska påståenden. Analysarbetet sammanställdes och överfördes därefter till diskussionen.

3.4 Validitet, reliabilitet och etik

3.4.1 Validitet

Validitet uppnås då flera människor har samma uppfattning om ett begrepp, så kallad intersubjektivitet (Jacobsen, 2002). Validitet kan delas upp i två delar: intern och extern validitet.

Intern validitet betyder resultatens giltighet och hur säker forskaren kan vara på att de slutsatser som dras kring ett testresultat verkligen stämmer (Jacobsen, 2002). Vi anser att studien har en hög intern validitet då samtliga intervjuobjekt har varit relevanta för studiens syfte. För att inte ge studien ett alltför smalt fokus inom tillämpningsområdet har vi intervjuat personer med varierande roller. Gemensamt för samtliga intervjuobjekten är att de haft djup insyn och erfarenhet av ledarskapet i sina respektive DevOps-team. Vi har dessutom valt mogna aktörer på marknaden vilket ökar studiens interna validitet.

Extern validitet rör överförbarhet och behandlar huruvida resultatet är generaliserbart och om det kan appliceras i andra situationer (Jacobsen, 2002). Det är inte avgörande för en kvalitativ studie som denna att resultatet är externt generaliserbart till större populationer, men för att uppnå hög extern validitet har vi under urvalsprocessen valt aktörer inom olika branscher vilket vi anser vara tillräckligt för att uppnå ett representativt urval för populationen, och därmed ge oss ett generaliserbart resultat.

3.4.2 *Reliabilitet*

Reliabilitet innebär enligt Jacobsen (2002) att studiens resultat skall vara pålitligt och lätt att följa. Reliabiliteten för en undersökning har säkerställts när flera studier oberoende av forskare kommer fram till samma resultat (Jacobsen, 2002). Huruvida denna studie anses vara reliabel är såklart svårt att bevisa då studien endast genomförts av två författare under en kortare period om knappt tio veckor. Inför samtliga intervjuer sammanställdes dock tydliga intervjuguider för att få relevanta och genomtänkta svar för att på så sätt öka reliabiliteten. Genom att spela in och transkribera intervjuerna reducerades även risken för snedvridningar i intervjumaterialet (Jacobsen, 2002).

3.4.3 *Etik*

Vid en undersökning finns det etiska principer att ha i åtanke. Jacobsen (2002) belyser tre viktiga grundkrav; krav på privatliv, att bli korrekt återgiven och informerat samtycke. Samtliga intervjuobjekt har frivilligt deltagit i undersökningen. Förfrågningar har skickats ut via mail och de har med andra ord kunnat tacka ja eller avböja sin medverkan. De har i förfrågan informerat om studiens fullständiga syfte, vilket givit dem rätt till informerat samtycke redan från start (Jacobsen, 2002).

Genom hela studien har vi ställt krav på privatliv gällande information som kan vara känslig, identifierande eller privat. För att undvika att ärliga svar inskränker på privatlivet har vi valt att erbjuda samtliga företag och intervjupersoner att vara anonyma. Detta har också intervjuobjekten informerats om redan vid start.

Slutligen, har vi ställt krav på att korrekt återge intervjuobjektens svar för att generera en så rättvis tolkning av det empiriskt insamlade materialet som möjligt. Citat bör exempelvis inte tas ur ett sammanhang, då det i sådana fall kan förvränga den ursprungliga meningen i uttalandet (Jacobsen, 2002). Alla intervjuobjekt har fått tillgång till den slutgiltiga presentationen av transkriberade data för att säkerställa att våra tolkningar var korrekt återgivna, och således upprätthålla en kvalitetsvalidering för insamlade data.

4 Resultat

Under detta kapitel kommer det empiriska resultatet presenteras utifrån den data som samlades in under våra fem intervjuer. Resultatet kommer presenteras i samma ordning som huvudområdena utifrån vår undersökningsmodell (Se Tabell 2.1).

4.1 Vision

Resultatet kopplat till huvudområdet vision uppdelat i de två ingående aspekterna: mål och mätning. Tabell 4.1 presenterar utifrån transkriberingsprotokollen intervjuobjekten samt vilken rad data är hämtad ifrån.

Tabell 4.1: Översikt av hämtade data för Vision

Vision	P1 (F1)	P2 (F1)	P3 (F2)	P4 (F3)	P5 (F4)
Mål	16, 20, 30, 32, 58, 60	16, 18, 38, 56, 74	12, 14, 16, 22	11, 13, 17, 27, 47, 55, 57, 59, 61, 63	12, 14, 22, 38, 42, 44, 48, 52, 70, 72
Mätning	46, 48, 50, 52	86	16, 18, 20	19, 21	6, 8, 10, 22, 42, 46

4.1.1 Mål

Det är tydligt att de har en uttalad strategi på F1, med övergripande mål om att göra det enkelt för alla och använda samma tekniska plattformar (P1:16, P2:16). Det är mer gemensamma mål för hela företaget, men att det kan förekomma att avdelningar har egna som eftersträvar att vara sammanhängande med företagets och dessa mål styrs utifrån vad intressenterna prioriterar (P2:18, P1:20). Vidare sätts det även upp mål för vad som skall uppfyllas inom vissa perioder och dessa kan förändras löpande med arbetet (P1:30, 32). De bedriver veckomöten för hela teamet där arbetsfördelning och prioriteringar behandlas, och för att alla skall få en känsla om vad som behövs göras (P1:58; P2:56). Om arbetet är för stort så blir det svårt att arbeta efter de gemensamma målen (P1:60). Det är därför viktigt att måla upp en helhetsbild och ta hänsyn till varandra och inte bara kasta över kod till driftsättningen (P2:74). Det också ett gemensamt tankesätt att lära av varandra (P2:38).

På F2 är det av stor vikt att hela teamet arbetar med en gemensam målbild (P3:12, 14). Likt F1, arbetar F2 med både övergripande mål mot en slutprodukt, samt delmål som är specifika för respektive team. Exempelvis har DevOps-teamet särskilda delmål då de använder sig av egna mätetal jämfört med andra team (P3:16). Trots dessa individuella delmål är det essentiellt att hela teamet delar samma vision och värderingar för att kunna uppnå organisatoriska mål (P3:22).

P4 understryker att det är viktigt på F3 att ha en gemensam bild för vart man vill samt att detta även stöds ovanifrån då ett av företagets värdegrund är 'tillsammans' (P4:59, 57, 55). Det är förstått att det måste finnas engagemang nerifrån och upp där kulturen är viktig, och utan kulturen kan inte DevOps bedrivas (P4:61, 63). Det är även viktigt att förstå hela poängen med arbetssättet och att arbeta utifrån att leverera värde snabbt (P4:47, 63). F3 arbetar utifrån arbetsmetodiken Scrum där en Product Owner (PO) sätter omfång samt hanterar och prioriterar en backlog (P4:11). De arbetar utifrån block över kalenderåret där större mål finns med, exempelvis om de skall rulla ut en ny funktion (P4:13). Det är sedan teamet själva som skall se till att dessa målen följs upp (P4:17). Allt detta koordineras på olika nivåer då de har olika rollout-planer för respektive marknad (P4:27).

Även P5 understryker kulturen som en viktig faktor för att bedriva DevOps, samt att alla delar samma värderingar och vision (P5:22). Vidare skall man arbeta mer kundcentrerat, i kombination med att ha klara mål, ansvarstagande och uppföljning för att lyckas (P5:42, 48). De vill inte att drift skall ses som en separat avdelning (P5:44), utan utvecklare måste ha större förståelse för hur driftsättningen fungerar (P5:38). Respektive PO fastställer, tillsammans med teamet, egna KPI:er de skall arbeta för att uppfylla (P5:12). De individuella målen kan se olika ut men det är viktigt att de delar samma helhetsbild (P5:14). Man vill bygga upp en medvetenhet på alla nivåer kring vad det huvudsakliga målet med DevOps är. Utan det målet understryker P5 att man aldrig kommer utvecklas (P5:48, 52, 70), då man lätt bara fokuserar på features och således missa de icke-funktionella kraven som skalbarhet och resiliens (P5:44). Vidare poängterar P5 att en sådan medvetenhet minskar risken för motstånd hos medarbetarna, vilket även är starkt kopplat till mognadsgraden (P5:48). P5 beskriver att vision, starkt ledarskap, och tekniska förutsättningar fundamentalt för att bedriva en DevOps-organisation (P5:72).

4.1.2 Mätning

För att hålla koll på statistisk och se till att problem upptäcks i tid, använder sig F1 av diverse mätverktyg (P1:46). Dessa används främst i övervakningssyfte där organisationen kan ta del av information om hur applikationer mår och hur hårt de arbetar (P2:86). F1 använder sig av dashboards för att visualisera sidvisningar och status för aktuella driftsättningar (P2:86, P1:48) som hela organisationen har möjlighet att ta del av och använda (P1:50, 52; P2:86). Utöver de mätpunkter som visas på stora skärmar, har vem som helst möjlighet att skapa egna grafer och följa applikationer av intresse (P1:52). I de flesta fall vet den enskilde utvecklaren bäst själv vilka mätpunkter som är mest intressanta att lyfta fram, vilket gör att utomstående kan ha svårt att avgöra om vissa värden är för höga eller låga (P1:52).

Att olika team har olika delmål, grundar sig i att F2 använder sig av olika mätetal (P3:16). På F2 bedrivs mätning för DevOps-team främst i form av SLA (Service Level Agreement) och antal fel (P3:18). P3 vill att allt skall vara mätbart, då det blir möjligt att följa upp allt som sker i teamen (P3:20).

För F3 bedrivs mätning i form av story-points, där estimering följs upp efter sprintens längd kontra kapacitet. Efter driftsättning, grundar sig mätning emellertid på KPI:er som conversion rates (P4:19). P4 beskriver vidare att olika radiatorer visar olika KPI:er, där conversion rates behandlas av en PO som har andra intressenter för hemsidan (P4:21).

På F4 är mätning grundläggande för att kunna vara en transparent organisation och generera snabb återkoppling (P5:22). P5 beskriver att en stor del av arbetet för att ta fram mätetal,

handlar om att mäta mognadsnivåer för olika team. Syftet är att avgöra huruvida ett team är grönt, orange, eller rött. En PO kan sätta KPI:er för exempelvis antal gröna team, men i slutändan är det upp till teamen (P5:6). Baserat på mätetalen, kan man tydligare analysera effektiviteten och kvaliteten för arbetet som utförs (P5:6). Var team fastställer egna KPI:er de skall uppfylla, utifrån teamets mognadsgrad. P5 understryker att det är ett föråldrat arbetssätt att utomstående personer skall sätta KPI:er inom ett team (P5:8). F4 har emellertid börjat röra sig mer mot OKR:er (Objective and Key Result). Det handlar om att man skall diskutera hur man skall mäta och kunna uppfylla ett uppsatt mål, vilket P5 menar är ett mycket bättre tillvägagångssätt än rena KPI:er. Målet är att arbeta med OKR:er på alla olika nivåer, där alla mål är sammankopplade till huvudmålen som finns på högsta nivån (P5:10). En stor del av DevOps-arbetet handlar om att bli mer kundfokuserade, således anser P5 att F4 behöver arbeta med att ta fram fler mätpunkter på hemsidan för att kunna få ytterligare återkoppling från kunder (P5:42). F4 använder sig delvis av automatiserade mätningssystem som gör det möjligt att effektivt kunna följa upp, spåra och mäta anrop eller vad som tar lång tid (P5:46).

4.2 Kommunikation

Resultatet kopplat till huvudområdet kommunikation uppdelat i de tre ingående aspekterna; gränsöverskridande kommunikation, kontinuerlig återkoppling samt delning av kunskap och verktyg. Tabell 4.2 presenterar utifrån transkriberingsprotokollen intervjuobjekten samt vilken rad data är hämtad ifrån.

Tabell 4.2: Översikt av hämtade data för Kommunikation

Kommunikation	P1 (F1)	P2 (F1)	P3 (F2)	P4 (F3)	P5 (F4)
Gränsöverskridande kommunikation	14, 18, 22, 36, 38, 58	4, 22, 32, 50, 54, 56, 60, 66, 80	8, 10, 24, 26	13, 23, 25	18, 20, 22, 24
Kontinuerlig återkoppling	20, 34, 44, 46, 50	32, 62, 66, 78, 90, 92	34, 36, 40, 42, 44, 46, 64, 66	21, 31, 45	22, 42, 46, 48
Delning av kunskap och verktyg	6, 8, 12, 14, 16, 42, 50	12, 30, 32, 34, 36, 38, 42, 50, 62, 68, 88, 98	28, 30, 54		20, 22, 24, 32, 50, 66, 68, 70

4.2.1 Gränsöverskridande kommunikation

P1 anser att han har god insyn i utvecklarnas arbete då han regelbundet närvarar på deras möten (P1:14), vilket innefattar vad utvecklarna diskuterar och har för problem (P1:18). Även samarbetet mellan redaktionen och utvecklarna är tätt, där redaktionen välkomnar utvecklare att delta på deras möten. Detta är något som P1 understryker inte hade fungerat att göra på distans (P1:22). Utveckling och drift arbetar som ett team, där man vill ha en känsla för vad utvecklarna gör i syfte av att vara medveten om pågående projekt samt planerade förändringar (P1:36, 38). Vidare bedrivs det veckomöten där arbetet planeras mer ingående (P1:58).

F1 använder morgonmöten för hela teamet för att ta upp saker från diverse parter om vad som behöver göras under dagen (P2:22, 56, 60). Med sin nya roll i DevOps-teamet, upplever P2 ett närmre samarbete med utvecklingsteamet (P2:4). Det finns en gemensam

kommunikationskanal där det nämns om något nytt skall driftsättas samt om man identifierat eller löst problem (P2:32, 50). Även utvecklare kan använda sig av kanalen om det råder problem i någon av miljöerna (P2: 54, 80). Driftpersonalen har kontakt med utvecklarna, som i sin tur står för kundkontakten (P2:66, 80).

På F2 innefattar teamen flera olika roller (P3:8), som i sin tur är uppdelade i olika strukturer med egna ledare (P3:10). Då de arbetar efter Scrum bedrivs dagliga möten där de involverar alla team leaders. Det finns även en dedikerad plats för någon som inte är team leader att medverka (P3:24, 26).

P4 beskriver att han samverkar med olika POs, som i sin tur samverkar med andra POs (P4:13). De arrangerar större möten där flera team medverkar, med syfte att presentera nya features och PI-plans (Project Incremental Plans) (P4:23). Kommunikationen i F3 sker inte bara gränsöverskridande mellan team utan även externt med kunder i andra länder för att få bekräftelse huruvida de exempelvis är redo för driftsättning eller vad som krävs för att gå ut med en specifik feature (P4:25).

På F4 har teamen fria tyglar och får sköta kommunikationen som de själva behagar, så länge det är sker effektivt (P5:18). Det är viktigt att det finns transparens i arbetet och att de arbetar tillsammans i produktteamen med en öppen kommunikation (P5:20, 22, 24).

4.2.2 Kontinuerlig återkoppling

På F1 vill man gärna att utvecklingsteamerna skall arbeta väldigt nära redaktionerna. Enligt P1, är syftet att utvecklarna skall arbeta i linje med redaktionens vision och kunna få kontinuerlig återkoppling på utfört arbete (P1:20). Om problem uppstår, är det på allas initiativ att både upptäcka och informera om problemet (P1:46). Återkoppling kan komma från alla nivåer i organisationen, från såväl honom själv som ledningen eller utvecklarna (P1:34, 44). F1 använder sig av dashboards som finns tillgängliga för alla medarbetare, vilket resulterar i att alla kan ta del av feedback som delas på skärmarna (P1:50). Utöver dashboards använder sig F1 av en gemensam kanal där samtliga medarbetare delar med sig av aktiviteter under arbetets gång (P2:32). Genom att dela med sig av arbetet kan övriga delar av organisationen ge respons på exempelvis problem som lyfts (P1:50). P2 beskriver att problem även kan lyftas under morgonmöten, där resterande deltagare hjälper till att diskutera fram en lösning (P2:62).

Trots att team ledaren har ett övergripande ansvar, så förekommer det att denne även rådfrågar medarbetare på lägre nivåer för att kontrollera utfört arbete, om inga fel har upptäckts på egen hand (P2:78). F1 ser gärna att utvecklarna arbetar nära kunden för att kunna ta del av problem och förslag direkt från slutanvändaren (P2:66). Slutligen beskriver P2 att samtliga i teamet får återkoppling i form av larm från applikationer och containrar. Sådana larm utlöses när problem uppstår, och då undersöker teamet bakomliggande orsaker (P2:92, 90).

P3 berättar att F2 arbetar med en leveranspipeline, vilket innebär att all form av återkoppling går genom en kundservice-portal för teamet (P3:34). På F2 har produktteamet direktkontakt med kunden med syfte att få återkoppling direkt från slutanvändaren (P3:40). Utvecklarna har inte någon direktkontakt med kund utan produktionsteamet delar istället med sig av informationen till utvecklarna om vad som behövs göras (P3:42). Trots att driften inte har någon direkt talan i produktutvecklingen, så kan de komma med givande inputs som kan bidra till produktens backlog. Det kan innefatta inputs som är relaterade till prestandan för

produkten, eftersom man inte ser slutresultatet förrän produkten är driftsatt (P3:44, 46). Innan hade man en väldigt rigid organisationsstruktur innan den nuvarande rotationsmodellen infördes (P3:66). Modellen innebär att DevOps-teamet utgörs delvis av permanenta medarbetare, men även tillfälliga medarbetare som vanligtvis tillhör utvecklingsteamet eller andra avdelningar i organisationen. De sistnämnda medarbetarna roterar med intervaller om två veckor (P3:36), med syfte att bland annat skapa en kontinuerlig feedbackcykel (P3:64). Innan rotationsmodellen implementerades, saknades återkoppling mellan avdelningarna helt. Bristen på återkoppling resulterade i att det inte fanns något utrymme för förbättring mellan avdelningarna (P3:66).

På F3 äger varje marknad sin egen webbplats, där de kontinuerligt följer upp lösningar och återkopplar globalt för incidenter som uppstår (P4:21). Vid problem, får den berörda marknaden en incident, och om det sker globalt genereras incidenter från samtliga marknader (P4:45). Återkoppling kommer direkt till en kommunikationskanal om att problem har uppstått, vilket sedan teamet gemensamt åtgärdar (P4:31).

P5 hävdar att ett Lean-orienterat arbete genererar flödesoptimering, snabb återkoppling, och reducerar waste som effektiviserar arbetet (P5:22). Vidare beskriver P5 att det finns agila myter som menar att sådant arbete inte kräver någon direkt planering, utan att allt arbete sker ad hoc. Sanningen är dock att agila arbetssätt är baserade på en väldigt tydlig sprintplanering och retrospektiv kopplat till det. Resultatet blir att man kan ta del av återkoppling, och därmed analysera vad som fungerade bra och vad som behöver åtgärdas (P5:48). På F4 har utvecklarna ingen personlig kundkontakt, istället sker återkoppling mer mekaniskt och automatiskt. Sådan återkoppling genereras från touchpoints där kundens beteende granskas och mäts, därtill bedrivs A/B-testning och conversion rates analyseras (P5:42). Larm och buggar är exempel på återkoppling som är automatiserat i de nya teamen. Man vill implementera larm på allt, så att de kan få snabb återkoppling om fel uppstår i infrastrukturen eller liknande (P5: 46).

4.2.3 Delning av kunskap och verktyg

F1 har börjat gå mot att använda gemensamma plattformar för att gynna kunskapsdelning då det är få som är duktiga både inom utveckling och driftsättning (P1:6, 16). Driften hjälper utvecklarna när de inte har förståelse för alla steg som det innefattar (P1:8, 12). När det väl uppstår problem är P1 positiv till de låga murarna, att man arbetar mer tillsammans (P1:42). Dock anser P1 att ytterligare kunskap kan delas genom medverkande i de olika avdelningarnas möten (P1:14). Systemövervakning sker genom dashboards som visas för alla, vilket ger indikationer om problem som finns eller kan tänkas uppstå (P1:50).

Att vara hjälpsam bedrivs kollektivt i F1 där man även kan rådfråga ledare vid problem (P2:12, 30, 34, 98). Om något i produkterna ändras sker informationsspridning i kanaler så att vem som helst kan följa det och lära sig (P2:32, 36). I kommunikationskanalerna delas det även om man identifierat felaktigheter (P2:50). Dessa felaktigheter kan tas upp vid möten där eventuella lösningar delas vidare till respektive ansvarig (P2:62). Man delar också med sig om kunskap för hur man gör vissa tekniskt operativa uppgifter för att stötta utvecklare om de vill sätta upp de miljöer som behövs för att bli mer självständiga och effektiva (P2:38, 42, 68). Vidare delas verktyg automatiskt när det skapas nya projekt (P2:88).

Alla som är med på veckomötena i F2 delar med sig av informationen och kunskapen med resterande i teamen (P3:28). På mötena finns en dedikerad plats som roteras, för att någon utomstående skall kunna delta och bättre sprida informationen samt kunskapen (P3:30, 54).

På F4 anses delning viktigt för att få återkoppling och arbeta mer transparent, och därmed bibehålla en öppen kultur (P5:20, 22). Vidare förespråkas det att sprida kunskap om saker som intresserar individerna, exempelvis nya programmeringsspråk (P5:24). Man skall arbeta med ett produkttänk och gemensamt tackla problem som uppstår. P5 understryker genom att ta hand om varandras kod kan lärandet ytterligare ökas (P5:32). Inputs kommer ofta från utvecklarna eftersom de är uppdaterade med nya trender och delar gärna med sig av det (P5:50). Vidare beskriver P5 en framtida ledare som mer tekniskt kunnig och utbildad för att kunna ta beslut baserat på fakta (P5:66, 68), vilket baseras på att sådan ledare skall ha full kontroll och kännedom om allt som sker i teamet (P5:70).

4.3 Innovativt och lösningsorienterat arbete

Resultatet kopplat till huvudområdet innovativt och lösningsorienterat arbete uppdelat i de två ingående aspekterna; innovativt och kreativt arbetssätt samt lösningsorienterat arbetssätt. Tabell 4.3 presenterar utifrån transkriberingsprotokollen intervjuobjekten samt vilken rad data är hämtad ifrån.

Tabell 4.3: Översikt av hämtade data för Innovativt och lösningsorienterat arbete

Innovativt och lösningsorienterat arbete	P1 (F1)	P2 (F1)	P3 (F2)	P4 (F3)	P5 (F4)
Innovativt och kreativt arbetssätt	30, 34, 40, 54	20, 48	48, 50, 60	49, 51, 53	18, 22, 24, 26, 28, 62, 64, 66, 72
Lösningsorienterat arbetssätt	28, 34, 56	8, 36, 50, 52, 54, 66, 84	46, 52, 58, 62	23, 33, 41, 47, 55	22, 30, 32, 36, 44

4.3.1 Innovativt och kreativt arbetssätt

Det är fria tyglar på F1 när det kommer till att agera för förbättring, där P1 understryker att man arbetar nära utvecklare för att ha möjlighet att lyfta och förverkliga deras idéer och förslag (P1:40). Generellt arbetar F1 aktivt med ett kreativt tankesätt som genomsyrar hela organisationen, och det uppmuntras att man tänker till och bidrar med sådant som kan gynna slutprodukten (P1:34, 54). Man motsätter sig ett arbetssätt där medarbetare blir tillsagda vad de skall göra och följer instruktioner till punkt och pricka, trots att de är medvetna om att problem kommer att uppstå. Ett sådant arbetssätt förekommer vanligen i externa utvecklingsteam i länder med annan kultur, vilket skiljer sig markant från det arbetssätt som F1 förespråkar (P1:34). Utvecklarna arbetar med nya idéer och förslag när de har tid över eftersom tidsbegränsade projekt prioriteras (P1:30). Likt P1, anser P2 att F1 välkomnar alla medarbetare att uttrycka sina idéer och förslag (P2:20, 48).

Medarbetare på F2 uppmuntras att vara kreativa och aktivt arbeta med problemlösning (P3:48), där alla typer av nya idéer och inputs välkomnas på dagliga möten (P3:50). F2 ger team möjligheten att implementera vad de vill, men skall först godkännas av teamets tekniska

ledning. Detta gör man för att förslag skall nå ett slags konsensus innan det implementeras (P3:60).

På F3 äger var team sin tekniska stack och har tillåtelse att göra ändringar för att generera mervärde (P4:49). P4 hävdar att det inte är fel att tänka nytt och utmanas, så länge man ser till att arkitekturen som helhet fungerar (P4:53). F3 uppmuntrar att implementera nya och moderna lösningar, och P4 anser att det är bra när saker förändras och förbättras med tiden. Om man är flexibel och adaptiv kan man enklare vända med vindar som råder (P4:51).

F4 vill inte samarbeta mer, istället vill de samarbeta mer effektivt. De vill vara snabbfotade och känna sig bemyndigade (P5:18). F4 har en värdering som kallas 'Different with a meaning', där det explorativa tankesättet kommer in i bilden genom att våga testa sig fram för att hitta nya lösningar (P5:22). Till en början saknade F4 struktur för innovation helt, och därefter övergick de till att ha ett enskilt team som arbetade med det - inkubation. Denna struktur resulterade i att innovationen inte fick någon spridning och kunde inte skalas på ett bra sätt. Idag arbetar F4 istället med att bygga in innovation i allt de gör (P5:24). Det nuvarande arbetssättet är mer kopplat till inkrementell innovation, vilket sker inom sitt område i organisationen. Vidare försöker F4 arbeta med Pareto-regeln 80/20, där 20% av det dagliga arbetet skall gå till att utbilda sig själv eller innovationsarbete som är kopplat till sin domän (P5:24). Som en förlängning på detta, har man även arbetat med att implementera Special Interest Groups (SIG) i organisationen. En SIG skall vara kopplat till något område man vill utvecklas inom, oavsett om det handlar om exempelvis ett programmeringsspråk eller arbetsmetod (P5:24). F4 har även en enhet som heter Development and Innovation, vilket är ett forum där medarbetare har möjlighet att pitcha sina idéer och få pengar för sin exploration (P5:26). De vill lyfta samtliga medarbetare och uppmuntra de till att dela med sig av nya idéer (P5:28).

P5 poängterar även det är viktigt att vara mottaglig för förändring (P5:64), där roller och arbetssätt behöver ändras i takt med att teamen växer (P5:62). För att lyckas arbeta i ständig förändring är det viktigt att vara lyhörd och ha förmågan att kunna rätta kursen efter vägen (P5:66). P5 beskriver att det är nödvändigt att F4 har en vision om att kunna transformera sig och bli mer digitala, både ur ett perspektiv som Chapter Lead men även från högsta ledningen (P5:72).

4.3.2 Lösningorienterat arbetssätt

Man beskriver att ett lösningorienterat arbetssätt uppmuntras i F1 (P1:34). Det kan hända att de släpper kod som de vet är inte tillräckligt bra. Anledningen är att det är viktigare att få ut fyra av fem funktioner, istället för inga alls och därmed låta kunden vänta (P1:28, 56). F1 använder sig även av en kommunikationskanal där utvecklare kontinuerligt delar med sig av sina problem (P2:8) som hela teamet arbetar gemensamt för att lösa, vilket skapar lärande inom organisationen (P2:50, 52, 36). Teamen arbetar aktivt för att upptäcka fel, och P2 understryker att ett manuellt arbete gör det enklare att identifiera problem (P2:54). Utvecklare arbetar nära kunden för att kunna åtgärda problem direkt (P2:66), där även driftpersonal kan hjälpa utvecklare att felsöka problem (P2:84).

På F2 får utvecklare inputs från driftpersonalen för att ständigt kunna förbättra slutprodukten (P3:46). DevOps-teamet hanterar kvalitetssäkringen som är en automatiserad process, vilket hjälper dem att identifiera problem i ett tidigt skede (P3:58). Problem som beror på driften är DevOps-teamets ansvar, medan utvecklingsteamet bär övergripande ansvar för fel i koden

(P3:52). Vidare menar man att det är av stor vikt att vara mottaglig för förändring eftersom allt förändras väldigt fort inom denna bransch (P3:62).

F3 arbetar med fokus på arbete utifrån att addera värde; om ingenting fungerar, adderas inget värde. Vid sådana tillfällen krävs det att kunna släppa allt annat och arbeta lösningsorienterat för att se till att organisationen levererar värde till kunden (P4:33, 41). Detta kan resultera i att ett helt team arbetar över helgen för att kunna åtgärda de mest kritiska problemen (P4:41). Vidare beskriver man att om en utvecklare får i uppgift att genomföra en process två gånger, så är det automatiserat den tredje gången, om inte tidigare. Ett sådant arbetssätt gör att teamet kan leverera mer värde då det inte längre kräver administration eller byråkrati (P4:47). Medarbetare brukar svärma runt inkrementella projektplaner för att diskutera om de skall engagera sig till planen eller inte (P4:23). Enligt P4 är det viktigt att vara mottaglig som DevOps-evangelist; det handlar inte bara om en last med autonoma processer, utan det är en stor kulturaspekt som skall vägas in också (P4:55).

Simplicity, ett värdeord för F4, syftar till att arbeta med smarta och enkla lösningar (P5:22). Vidare beskriver P5 att teamen arbetar gemensamt med problemlösning (P5:32), där felaktig kod antingen rullas tillbaka alternativt att de tillsammans arbetar fram en lösning som åtgärdar problemet direkt (P5:30). Det blir allt viktigare att utveckling sker förberedande och på ett smart sätt (P5:36). Det är väsentligt att drift-tänket kommer in tidigare i utvecklingsprocessen, så att den nedåtgående spiralen med fokus på funktioner kan brytas. Med större förståelse för driften, kan feature-jakten bryggas på ett bättre sätt (P5:44).

4.4 Individuell omtanke

Resultatet kopplat till huvudområdet individuell omtanke uppdelat i de tre ingående aspekterna; förtroendekultur, agera coach och minska beroenden. Tabell 4.4 presenterar utifrån transkriberingsprotokollen intervjuobjekten samt vilken rad data är hämtad ifrån.

Tabell 4.4: Översikt av hämtade data för Individuell omtanke

Individuell omtanke	P1 (F1)	P2 (F1)	P3 (F2)	P4 (F3)	P5 (F4)
Förtroendekultur	28, 54, 58, 60	12, 20, 24, 32, 94	26, 38, 48, 50, 60	15, 29, 33, 41, 53	6, 8, 26, 30, 48, 62
Agera coach	54, 58, 60	8, 12, 14, 28, 34, 82		29, 45, 63	6, 12, 16, 24, 26, 56, 64, 66, 68
Minska beroenden	24, 42, 44, 46	22, 34, 94, 96	26, 36, 38, 52	17, 33, 37, 39, 45, 49, 63	6, 16, 18, 30, 32, 36, 54, 56, 58, 60, 62, 70

4.4.1 Förtroendekultur

Idéer som implementeras på F1 kan komma från vem som helst i organisationen (P1:54). De har en kultur där det är odramatiskt att driftsätta ny kod, eftersom man gör det hela tiden (P1:28). På DevOps-avdelningen syns alla, då det inte är meningen att man skall gömma sig (P1:58). P1 beskriver att det alltid är viktigt att ha team där alla syns och hörs, oavsett om man arbetar med DevOps eller inte. I stora organisationer är det större risk att individer

hamnar i skymundan, vilket kan resultera i sämre välmående bland medarbetarna. Man menar att individens välmående ligger till grund för bra arbetsprestationer (P1:60).

P2 tillägger att medarbetare alltid har möjlighet att lyfta sina idéer och göra sin röst hörd (P2:20). Uppdrag ansvaras av den som känner sig mest mogen eller lämpad för att ta sig an uppgiften (P2:24, 94). Innan kod driftsätts, brukar aktiviteten lyftas med teamet för att kontrollera att ingen ser någon problematik med den (P2:32). F1 har även en tekniskt stödjande roll som kallas 'Hatten', som roteras veckovis mellan alla teamets medlemmar (P2:12).

För P3 är det viktigt att alla känner sig involverade i arbetet, så att alla har något att säga till om. Alla har möjlighet och tillåtelse att lyfta sina idéer och inputs på de dagliga mötena, vilket sedan diskuteras gemensamt med hela teamet (P3:48, 50). Det är inte alltid samma personer som arbetar i DevOps-teamet (P:38), vilket gör att alla kan känna sig engagerade i teamet oavsett vilken roll de har (P3:26). Man låter teamen vara autonoma och ger dem möjlighet att arbeta utifrån eget bevåg (P3:60).

På F3 strävar man efter att ha en sund kultur där man inte tvingar någon till att göra något de inte vill (P4:53). Alla team har olika mognadsnivåer (P4:29), där vem som helst i ett moget team kan ta sig an problem som uppstår (P4:33). Man vill att teamen själva skall ta ansvar för att välja en passande metod för deras team och arbetssätt, oavsett om det är Scrum, Kanban, eller Lean (P4:15). Således bär hela teamet även ansvar för problem som uppstår (P4:41).

Frihet under ansvar är en viktig faktor för DevOps (P5:48). P5 vill att teamen sätter sina egna KPI:er baserat på deras mognadsgrad (P5:6), vilket är svårt för någon utomstående (P5:8). F4 förespråkar att alla lyfter sina idéer och gör sin röst hörd, och tar gärna input från industrin när det kommer till nya verktyg eller liknande (P5:26). Vidare beskriver P5 att teamet tar eget ansvar för budget och dagligt arbete (P5:62). Man skall heller inte skylla ifrån sig eller peka finger, vilket är något som ligger invävt i kulturen (P5:30).

4.4.2 *Agera coach*

På F1 bedrivs det ett restriktivt ledande för att bibehålla sammanhållningen samt vägleda istället för att lägga sig i arbetet (P1:58, P2:28, 34). Vem som helst kan komma för att ge uppmuntrande i form av väl utfört jobb (P1:54). Vidare är det viktigt att alla syns inom teamen och får sin röst hörd (P1:60). P2 beskriver att teamet har en roterande position vars funktion är att stötta utvecklare som har problem, som en slags support (P2:8, 12, 14, 82).

P4 beskriver att de arbetar med olika mognadsgrader inom teamen som ledaren ser över för att kontrollera om de behöver mer handpåläggning eller kan arbeta fullt autonoma (P4:29). Skulle det ske incidenter så är det teamets och PO ansvar att se till att det finns tillräckligt med resurser (P4:45). Man menar vidare att arbeta med DevOps inte skall vara toppstyrt. Genom att arbeta fram rätt kultur och aktivt stötta denna, kommer teamen sköta sig självständigt (P4:63).

På F4 så sprids idéer nerifrån och upp där ledaren är lyhörd och lyssnar till detta (P5:24, 26, 64, 66, 68). P5 beskriver även att det övergår mer till att team tillsammans är fullt ansvariga för arbetet och att ledare stöttar detta (P5:16, 56). Ledaren arbetar för att tydliggöra arbetet utifrån vad teamen själva vill uppnå där anställda får komma med inputs och beskriva hur de bäst arbetar (P5:6, 12, 16).

4.4.3 *Minska beroenden*

På F1 är man ganska beroende av varandra då många personer är kopplade till produkter samt att det förekommer team som är mer specialiserade (P1:24). Trots att respektive avdelning är ansvarig för deras egna incidenter, är det fortfarande låga barriärer mellan dem (P1:42, 46). Vidare skall utvecklare ha koll så att deras kod verkligen fungerar (P1:44). För att minska beroendet av specifika personer används en roterande position samt ett kollektivt ansvar för att personer som orsakat problem åtgärdar det (P2:22, 34, 96). Skulle det finnas incidenter så kan den som känner sig mest manad att ta sig an problemet ta hand om det (P2:94).

Likt F1, beskriver P3 att F2 även har en roterande position för att minska beroende av specifika individer (P3:26, 36). Detta resulterar i att det inte alltid är samma personer som arbetar i teamet (P3:38). När det väl uppstår problem så kan det variera vem som är ansvarig för att åtgärda dem, men det är oftast de själva som är ansvariga (P3:52).

På F3 har teamen, beroende på mognadsgrad, fullt kollektivt ägandeskap där vem som helst kan ta sig an problem och inputs (P4:17, 33, 37, 39). Mellan team kan det förekomma olika programmeringsspråk vilket de bestämmer själva, så länge det fungerar. De tar fullt ansvar för sitt eget affärsområde och har friheten att organisera detta själva (P4:45, 49, 63)

P5 belyser att team skall innefatta tvärfunktionell kompetens (P5:6). F4 arbetar med att minska beroendet av specifika roller och maktpositioner (P5:36, 54, 56, 70), där teamen själva är ansvariga för vad de gör (P5:16, 60). Det skall finnas kollektivt ledarskap i teamen där ansvaret delegeras och sprids vidare (P5:58, 62, 18). Det skall även bedrivas ett kollektivt ansvar i att hantera problem som uppstår (P5: 30, 32).

5 Diskussion

Följande kapitel redovisar det empiriska resultatet utifrån intervjuerna i relation till teorin för att identifiera likheter och olikheter. Huvudområden, tillsammans med underliggande aspekter, kommer att diskuteras utifrån ordningen i vår framtagna undersökningsmodell (Se Tabell 2.1).

5.1 Vision

5.1.1 Mål

Litteraturen beskriver genomgående vikten av att arbeta med en gemensam målbild och delade värderingar. Dessa skall genomsyra hela organisationen, och inte varieras mellan team eller avdelningar. Empirin lyfter att samtliga företag delar litteraturens uppfattning, där samtliga intervjupersoner belyser vikten av att alla medarbetare arbetar utifrån en gemensam organisatorisk målbild, där alla delar samma värderingar och vision. P5 poängterar även att det är viktigt att öka medvetenheten om de övergripande målen för att kunna utvecklas som en DevOps- organisation. En sådan medvetenhet kan bidra till minskat motstånd och ökad produktivitet, då alla arbetar mer målorienterat med ett större kundfokus. Sharma (2014) menar att eliminering av sådana flaskhalsar förenklar och effektiviserar ett gränsöverskridande samarbete.

På F1, F2 och F4 förekommer det att olika enheter har individuella mål, utöver de gemensamma målen. Dessa särskilda mål kan vara för att det bedrivs olika typer av mätning i respektive enhet alternativt att mål kan förändras löpande under arbetets gång. Att använda sig av individuella mål kan, enligt Sharma (2014), bidra till att det uppstår konflikter då mål kan ligga i motsats till varandra. Trots individuella mål, anser företagen att det är viktigt att de fortfarande ligger i linje med de övergripande målen.

Konflikter som uppstår i samband med skilda mål kan, enligt P2 och P5, motverkas genom att driften involveras tidigt i utvecklingsarbetet, vilket gör att utvecklarna får en bredare förståelse för driftsättningen. Enheterna bör sätta upp en gemensam målbild och ta mer hänsyn till varandra. Detta ligger i linje med Hüttermann (2012) som poängterar att ett sådant arbetssätt stärker samarbetet mellan utvecklarna och driftpersonalen. Empirin visar även att flertalet företag kommunicerar ut vision och aktiviteter på kontinuerliga möten med hela teamet. Van Kelle et al. (2015) hävdar att delade värderingar enklare uppnås med hjälp av regelbunden kommunikation.

5.1.2 Mätning

Ingen av intervjuobjekten bedriver mätning för endast produktivitet. Fokus ligger istället på systemets välmående, vilket överensstämmer med Forsgren et al. (2018) och Humble & Molesky (2011). Mätning för produktivitet kan ändra medarbetares beteende i att exempelvis

driftsätta onödig kod. Utifrån empirin kan mätning syfta till olika aktiviteter sinsemellan företagen. Uppföljning är emellertid en gemensam nämnare för samtliga organisationer. Genom att göra allt mätbart, kan arbete enklare följas upp och förbättras. Därtill, bedriver F4 mätning i form mognadsnivåer för olika team. Mognadsgraden ligger till grund för teamens individuella måttal, vilket P5 understryker kan förenkla förmågan att analysera effekten och kvaliteten för arbetsprestationer. Vidare understryker P5 att KPI:er bör utarbetas inom teamen utifrån egen förmåga, vilket ligger i linje med Humble & Molesky (2011). Författarna hävdar att mätning skall ske omsorgsfullt på olika nivåer. Trots att F4 kombinerar mätning på gemensam och individuell nivå, är det i enlighet med Farroha & Farroha (2014) som poängterar att måttal skall delas över hela organisationen. En transparent organisation kan fördelaktigt generera snabb och kontinuerlig återkoppling.

Yukl & Kaulio (2011) och Humble & Molesky (2011) beskriver att mätning och visualisering av arbete kan bidra till ökad förståelse och medvetenhet bland medarbetarna. Både F1 och F2 bedriver mätning i form av övervakning. Det förstnämnda företaget använder sig av visualisering av arbete som finns tillgänglig för alla i organisationen. Övervakningen syftar till att kontrollera och analysera status för applikationer och system, vilket gör det möjligt att identifiera och åtgärda problem redan i ett tidigt skede.

5.2 Kommunikation

5.2.1 Gränsöverskridande kommunikation

Empirin visar att samtliga företag har en gränsöverskridande kommunikation mellan utvecklarna och driftpersonalen. På flera företag har kommunikationen även fått en större spridning med kontinuerlig kontakt med såväl kunder som ledning. Kommunikationen sker delvis genom kommunikationskanaler där problem diskuteras, men även genom kontinuerliga möten där personal från olika avdelningar medverkar. Syftet är att enheterna skall få en större insyn i varandras arbete, informera om pågående aktiviteter, samt underlätta för framtida planeringsarbete. Humble & Molesky (2011) uttrycker sig liknande och menar att ett gränsöverskridande samarbete möjliggörs genom att involvera utvecklare och driftpersonal i varandras arbete genom uppföljningsmöten och regelbunden kommunikation. Författarna nämner även att driftpersonal skall underrättas när ny funktionalitet är på väg. Detta är något som P2 understryker och beskriver att information om nya funktioner delas till hela teamet med hjälp av kommunikationskanaler.

Ett gränsöverskridande samarbete skapar en ökad medvetenhet för produkten genom hela livscykeln. Samtliga företag skapar en sådan medvetenhet hos medarbetarna genom att bedriva kontinuerliga uppföljningsmöten och bibehålla en transparent organisation. Ravichandran et al. (2016) och Walls (2013) är eniga och menar att ett kollektivt fokus skall ligga på slutprodukten, istället för subprocesser eller enskilda funktioner. Ett sådant arbete gör att problem, lösningar, och aktiviteter kommuniceras ut till hela teamet och inte endast till den berörda avdelningen. Detta är något som P4 understryker och beskriver att man skall se produkten som helhet vilket bidrar till att avdelningar dels får respekt och dels mer förståelse för varandra.

Kommunikationen skall enligt Bass & Riggio (2006) bedrivas på ett stimulerande sätt för att öka teamets förmåga att prestera och vidare utveckla individernas ledarskapsförmågor.

Empirin visar att F4 ger teamen friheten att själva bestämma hur kommunikationen skall bedrivas. Det enda kravet är att det skall ske på ett effektivt och fördelaktigt sätt. Genom att arbeta utifrån DevOps är det tydligt att kommunikation sker på flera olika sätt för att involvera och integrera alla i mjukvarulivscykeln. Empirin överensstämmer med litteraturen om vikten av att kommunicera gränsöverskridande.

5.2.2 Kontinuerlig återkoppling

Samtliga intervjupersoner ser gärna att problem som uppstår skall lyftas med hela teamet. Genom snabb återkoppling kan problem effektivt åtgärdas. Sökning och återkoppling bedrivs kollektivt för att motverka att dessa incidenter uppstår. Detta stöds av Kim et al. (2016) som menar att man skall arbeta tillsammans för att isolera problem och hindra dess eventuella påföljder. Återkoppling inom företagen sker antingen genom kommunikation, alternativt automatiserade verktyg som informerar samtliga om incidenter. Båda metoderna skapar ett effektivt kontinuerligt flöde av återkoppling som når ut till hela teamet. Även detta ligger i linje med Kim et al. (2016) som menar att alla skall bli informerade, speciellt individen som orsakade problemet, även om det åtgärdas tillsammans. Sådan återkoppling kan bidra till lärande inom teamet, vilket gör att arbetet och slutprodukten ständigt förbättras.

Kontinuerlig återkoppling gör det möjligt att identifiera och åtgärda problem i ett tidigt skede, innan det blir för dyrt och komplext (Kim et al., 2016). F2 implementerade en modell med rotering av personal för att skapa en cykel för feedback inom teamen. Resultatet blev att återkoppling började ske gränsöverskridande mellan teamen, vilket bidrog till att fler förbättringsområden kunde identifieras. Att dela med sig av information mellan teamen kan, enligt Davis & Daniels (2016), eliminera onödiga flaskhalsar och bidra till en förbättrad slutprodukt.

Kontinuerliga leveranser är en viktig beståndsdel för att bedriva DevOps, vilket även gör det möjligt för teamen att erhålla både intern och extern återkoppling (Sharma, 2014). F1, F2 och F4 placerar produktutvecklingen nära slutkunden med syfte att kunna få extern återkoppling direkt från slutanvändaren. Intern återkoppling kan fördelaktigt komma från alla olika nivåer i organisationen, då flera perspektiv kan tas i beaktning. Återkoppling från ledningen kontrollerar att utvecklingsarbetet ligger i linje med visionen, driften kan bidra med återkoppling som har en inverkan på prestandan för produkten, samt lägre nivåer som kan påpeka problem som en team leader inte har upptäckt.

5.2.3 Delning av kunskap och verktyg

Litteraturen är enhetlig om att kunskap och verktyg skall delas över hela organisationen vid tillämpning av DevOps. Alla företag förutom F3 bedriver någon form av kunskap- och verktygsdelning. Empirin tyder på att det bedrivs aktiviteter för att gynna detta och skapa ett lärande inom teamen, vilket ligger i linje med Perera et al. (2017) och Hüttermann (2012) som belyser vikten av att dela med sig av problem. P1 understryker att det är få som besitter kunskap inom både utveckling och driftsättning. Att inte ha kunskap inom båda områdena kan orsaka framtida problem för att arbeta med kontinuerlig driftsättning. Driftpersonalen hjälper därför utvecklarna med att komplettera deras kunskap när det är något de inte förstår. Detta sker främst för att de skall ha möjligheten att sätta upp egna produktionsmiljöer för att kunna arbeta mer effektivt.

Delning av kunskap sker inte bara för problem, utan även idéer som vidare kan förbättra arbetet. P5 understryker att idéer och inputs ofta kommer från nerifrån för att främja detta. Hüttermann (2012) och Perera et al. (2017) förespråkar att idéer och förslag delas för att förbättra samarbetet. Kommunikationskanaler och kontinuerliga möten används för att sprida såväl problem som idéer med hela teamet. Möten gör det möjligt för övriga medarbetare att ta del av återkoppling från både kund och andra avdelningar, vilket P4 poängterar. Vidare understryker P1 att gemensamma möten med hela teamet kan ses som en möjlighet för en bredare kunskapsspridning över flera avdelningar.

Rakowski (2015) hävdar att kunskapsdelning bidrar till att skapa acceptans och en fungerande DevOps-kultur. Ett sådant arbete bedrivs på F4 genom att främja återkoppling och transparens för att upprätthålla en öppen kultur. Vidare menar de att kunskapsdelning även kan baseras på individuella intressen, vilket kan underlätta acceptansen för nya lärodomar bland medarbetarna.

Verktyg delas per automatik inom F1 när nya projekt påbörjas. De har börjat använda gemensamma plattformar för att få en bättre insyn i hur utvecklarna arbetar. Att dela med sig av verktyg, tillsammans med kunskap, är något som Perera et al. (2017) och Hüttermann (2012) anser krävs för att upprätthålla ett effektivt samarbete som DevOps-team.

5.3 Innovativt och lösningsorienterat arbete

5.3.1 *Innovativt och kreativt arbetssätt*

Innovation och kreativitet är något som genomsyrar samtliga företag, vilket tydligt ligger i linje med litteraturen. Medarbetarna på F1 har friheten att agera efter förmåga vad gäller förbättringsarbete, där företaget aktivt hjälper medarbetare att lyfta och förverkliga nya idéer. När sådan motivation kommer ovanifrån, tenderar medarbetare att arbeta mer lösningsorienterat genom att se problem på ett mer innovativt och kreativt sätt (Bass & Riggio, 2006).

Empirin visar att innovation är en del av det dagliga arbetet för samtliga företag. Detta ligger i linje med Forsgren et al. (2018), som därtill menar att en innovationskultur kan bidra till lärande inom organisationen. Tidigare hade F4 endast ett enskilt team som arbetade aktivt med innovation, vilket orsakade att innovationen inte fick någon större spridning inom organisationen. Idag arbetar de istället med innovationsarbete inom alla team, och i allt de gör. F4 arbetar utifrån ett explorativt tankesätt som är värdegrundade för organisationen, vilket uppmanar medarbetarna att våga testa nya lösningar. Utöver detta, arbetar F4 utifrån Pareto-regeln som syftar till att medarbetare skall ägna 20% av sitt dagliga arbete åt utbildning eller innovationsarbete. De andra företagen ger inte uttryck för en lika tydlig struktur för innovationsarbete. Däremot påvisar F1, F2 och F3 explicit att ett sådant tankesätt uppmuntras och värdesätts.

Rakowski (2015) menar att misslyckanden måste accepteras för att kunna bedriva en innovativ DevOps-organisation. Detta bidrar till att medarbetare vågar utmanas för att hitta nya lösningar som kan förbättra och effektivisera arbetet, utan risken att känna sig begränsade. F3 uppmuntrar sina medarbetare att vara innovativa och utmana sig själva, så länge arkitekturen som helhet fungerar, vilket ligger i linje med litteraturen. Genom att uppmuntra nya lösningar och vara mottaglig för förändring, skapas team av högre flexibilitet

och anpassningsbarhet. Även F2 uppmuntrar teamen att implementera nya lösningar och idéer som kan gynna arbetet. De vill emellertid att implementeringen skall godkännas av en ledare för att uppnå konsensus, vilket kan begränsa medarbetarna och resultera i onödiga flaskhalsar.

Ett djupgående tankesätt kan bidra till ökat självförtroende, engagemang och ansvarstagande enligt Walumbwa et al. (2004) och Yukl & Kaulio (2011). Det bedrivs ett forum på F4 där medarbetare är välkomna att lyfta och sälja in sina idéer. Medarbetarna får resurser och huvudansvar för idén, vilket kan öka engagemanget ytterligare. Även F1 uppmuntrar ett sådant tankesätt, då det kan skapa mervärde i arbetet och gynna slutprodukten.

5.3.2 Lösningsorienterat arbetssätt

Att arbeta lösningsorienterat beskrivs av samtliga företag, vilket överensstämmer med litteraturen, dock sker det i olika utsträckning inom företagen. Rafferty & Griffin (2004) belyser vikten av att arbeta kreativt och lösningsorienterat för att effektivt lösa dilemman som uppstår. F1 och F3 uppmuntrar ett lösningsorienterat arbetssätt där kunden står i fokus för att kunna leverera värde. F1 kan medvetet rulla ut lösningar som saknar enstaka funktioner, då man inte vill låta kunden vänta. Vidare arbetar man hellre manuellt då man kan identifiera felaktigheter enklare. När produktsläpp närmar sig, händer det att F3 anpassar arbetssättet för att bara fokusera på att leverera funktionaliteter som kan generera värde åt kunden. Driftarbetet bör, enligt F4, tidigt involveras i mjukvarulivscykeln och skapa en bättre förståelse för hur det kommer fungera i produktionsmiljön. Detta för att enbart kunna fokusera på att addera värdeskapande funktionaliteter. Vidare beskriver P3 att företaget lägger stor vikt vid att vara öppen och adaptiv då branschen ständigt förändras.

Inom F4 arbetar man utefter värdeord som genomsyrar organisationen. De värdesätter att arbeta bland annat utefter smarta och enkla lösningar. Man arbetar tillsammans för att åtgärda de incidenter som uppstår. Enligt Ravichandran et al. (2016) skall ingen individ bära särskilt ansvar för problem som uppstår vilket överensstämmer med F4. Det finns även likheter i F1 där personal inom driftavdelningen kan hjälpa utvecklarna att felsöka deras kod. Detta ligger vidare i linje med Kim et al. (2016) som menar att man kontinuerligt skall arbeta med problem.

Med hjälp av automation kan F3 säkerställa kvaliteten genom att tidigt identifiera uppkomna problem. Kim et al. (2016) och Farroha & Farroha (2014) anser att man tidigt skall identifiera detta, då man lättare kan hantera liknande problem i framtiden om det återkommer.

5.4 Individuell omtanke

5.4.1 Förtroendekultur

Samtliga företag beskriver utifrån empirin någon form av arbete för att aktivt främja en förtroendekultur. Detta likt Walls (2013) och Davis & Daniels (2016) som hävdar att en stark förtroendekultur är en förutsättning för ett gränsöverskridande och innovativt DevOps-team. Vidare anser Davis & Daniels (2016) att man skall ge medarbetarna ett förtroende att själva bedriva sitt arbete. Det ges helt fria tyglar till team på F3 att ta ansvar för deras arbetsuppgifter och vad för typ av arbetsmetod de önskar att använda. Det går även hand i hand med att de själva blir ansvariga för problematik som detta medför. Likt F3, låter F2 och

F4 även deras team vara autonoma och ansvara för deras arbetsområde. Görs inte detta menar Davis & Daniels (2016) att delning av information kan utebli och understryker en negativ påverkan på produktiviteten.

DevOps förespråkar en kultur där man inte skyller ifrån sig eller pekar finger åt andra team eller avdelningar, vilket transformellt ledarskap kan understödja (Rafferty & Griffin, 2004). Detta är något som P5 beskriver ligger invävt i organisationens kultur. Farroha & Farroha (2014) menar att man kollektivt skall ta ansvar för allt som teamet producerar, oavsett resultatet. För att öka förtroende behöver samtliga få sin röst hörd om idéer vilket F1 och F3 belyser. De förespråkar att team lyfter funderingar och problem, vilket Ravichandran et al. (2016) tyder på respekt där arbete värdesätts. Ravichandran et al. (2016) beskriver även förtroende som att individer från driftavdelningen skall kunna bidra med återkoppling till utvecklarna. Inom F2 skall alla känna sig involverade och aktivt komma med inputs. De ägnar sig åt att rotera individer, oavsett roll, mellan olika team, vilket skapar förtroende och engagemang.

Walls (2013) beskriver att det skall finnas tillit från driftavdelningens sida i att utvecklarna arbetar utifrån slutproduktens bästa. Detta överensstämmer inte med vad F1 beskriver, då utvecklarna informerar vid nya kodändringar för att få ett godkännande från driftpersonalen om eventuella felaktigheter.

5.4.2 *Agera coach*

Alla företag, utom F3, beskriver att ledarskap inom teamen sker i en mer restriktiv bemärkelse. Likt Hussain et al. (2017) har ledarskapsattribut påvisats förekommande för medarbetare i DevOps-team. En sådan ledare betraktas mer som en mentor eller facilitator genom att vägleda medarbetarna snarare än lägga sig i arbetet. Vidare ligger fokus på att bevara en stark sammanhållning och engagera alla inom teamet. Genom att uppmärksamma och lyssna till alla medarbetare kan teamet enklare vägledas utifrån en gemensam vision och förmåga. Enligt litteraturen skall en ledare agera mer för att ge stöd åt medarbetarna. På F3 och F4 har teamen ett gemensamt ansvar att facilitera arbetet, där ledaren har en mer stödjande roll. F1 bedriver en roterande roll med syfte att agera coach och understödja utvecklingsarbetet. Empirin tyder således inte på att en ledare behöver kontrollera att arbetet följer en utarbetad plan, som Wiedemann (2018) antyder. Istället menar F3 att en ledare skall fokusera på teamets mognadsnivå för att avgöra huruvida teamen kan arbeta fullt autonomt, eller om mer handpåläggning krävs. En genomarbetad och understödjande kultur gör att team kan arbeta mer självständigt, istället för att DevOps-team skall styras ovanifrån. Det blir därför allt viktigare att förespråka samarbetsvilliga team, vilket även Wiedemann (2018) belyser.

Intervjupersonerna är eniga om att en ledare skall vara lyhörd och se till att alla får sin röst hörd, oavsett vilken nivå det kommer från. Yukl & Kaulio (2011) menar vidare att en sådan ledare ofta betraktas som en coach genom att stödja och uppmuntra medarbetarna. P1 menar emellertid att det är ett kollektivt ansvar att uppmuntra och berömma medarbetare för utfört arbete. Genom att uppmärksamma och stödja samtliga medarbetare kan förtroendet öka, vilket i sin tur kan bidra till fler vågar yttra sig om nya förbättringsområden.

5.4.3 *Minska beroenden*

Alla företag arbetar, mer eller mindre, aktivt för att minska beroenden inom teamen. I enlighet med empirin, betonar Farroha & Farroha (2014) vikten av att undvika en position där teamet är beroende av enskilda personer. Genom att komponera team som besitter tvärfunktionell kompetens, vill F4 att teamet skall ha ett kollektivt ansvar i att bedriva arbetet och hantera problem som uppstår. F1 och F2 använder en roterande position inom teamen som gör att teamet inte alltid består av samma personer, vilket även det minskar beroendet av enskilda medarbetare. Alla företag bedriver emellertid ett kollektivt ansvar för problemlösning, vilket gör att enskilda medarbetare inte bär skuld för problem som uppstår. Detta ligger i linje med Farroha & Farroha (2014) som menar att team skall uppmärksammas som helhet.

Att inte behöva förlita sig på kompetenser för enskilda medarbetare, kan vara en avgörande faktor för att lyckas bedriva en effektiv verksamhet (Davis & Daniels, 2016). Trots arbetet ovan nämnt, anser sig F1 fortfarande aningen beroende då det fortfarande förekommer team som besitter specialistkunskaper samt medarbetare som är direkt kopplade till produkten. Om sådana nyckelpersoner oväntat försvinner, kan det resultera i oanade konsekvenser för organisationen. För att minimera risken för det, försöker F1 hålla nere barriärerna för att öka insynen i arbetet.

6 Slutsats

Det huvudsakliga syftet med DevOps är att forma ett närmre samarbete mellan utvecklarna och driftpersonalen för att effektivt optimera hela mjukvarulivscykeln och därmed förkorta ledtider. Detta bidrar till snabba och kontinuerliga förändringar vilket i sin tur skapar en bättre återkopplingscykel, vilket bidrar till ett ökat kvalitetsarbete med en mer stabil och pålitlig infrastruktur. Ledarskap är en viktig beståndsdel för att kunna bedriva ett gränsöverskridande samarbete, där fokus skall ligga på att skapa en gemensam vision och målbild. Trots detta är ledarskap ett förbisett ämne inom just DevOps. Transformellt ledarskap har tidigare uppmärksammats inom agil mjukvaruutveckling, då litteratur har påvisat samband mellan ledarskapsstilen och effektiv mjukvaruleverans. Vidare behandlar litteraturen inte hur transformellt ledarskap bedrivs i relation till DevOps, vilket mynnade ut till vår forskningsfråga: *Hur bedrivs transformellt ledarskap i team som arbetar med DevOps?*

Genom vår studie har det påvisats att det transformella ledarskapet inte bedrivs utifrån en enskild ledare, utan är snarare en kollektiv ansats för hela DevOps-teamet. Empirin har konstaterat att ansvaret är utspritt över flera medarbetare som arbetar tillsammans för produktens bästa. DevOps-team sammanför aspekter av det transformella ledarskapet i kulturen som ligger till grund för allt arbete där visionen skapar en tydlig målbild. Teamen tillåts att vara mer självstyrande och ansvara för sin produkt genom hela livscykeln. Detta ställer emellertid krav att arbetet inom teamet sker effektivt och att man gemensamt ansvarar för att problem åtgärdas och att kunskap sprids vidare.

Studien har grundat sig på fyra huvudområden, med underliggande aspekter, för hur det transformella ledarskapet bedrivs i en DevOps-kontext. Det har påvisats att företagen arbetar frekvent med aktiviteter för att främja ett gränsöverskridande samarbete. DevOps har emellertid definierats på olika sätt inom företagen, vilket kan vara underliggande orsak till att aktiviteterna har varierat.

Litteraturen belyser vikten av att DevOps-team skall använda sig av gemensamma mål och mätetal för att främja ett gränsöverskridande samarbete. Trots att företagen aktivt arbetar utifrån en delad vision, förekommer både mål och mätning även på enskild team-nivå. Anledningen är att team har olika mognadsgrader och använder sig därför av olika mätetal som speglar de individuella målen. Teamen ser själva till att mål uppfylls, och det blir således ett gemensamt ansvar att se till att både mål och mätning sker i linje med den övergripande visionen.

Transformellt ledarskap kan motivera och inspirera medarbetare genom öppen och tydlig kommunikation. Alla inom DevOps-teamen engageras i kontinuerliga möten där utvecklarna och driftpersonalen involveras tidigt i varandras arbete. Detta skapar transparenta DevOps-team där problematik, lösningsförslag och aktiviteter aktivt kommuniceras ut, både internt och externt. De studerade företagen använder sig av egna metoder för att så effektivt som möjligt bedriva kontinuerlig återkoppling, och ser positivt på att återkoppling kommer från alla nivåer i organisationen. Studien tyder på att sådan involvering och kunskapsdelning kan bidra till ökad motivation och acceptans bland medarbetarna. Det förekommer att team själva

får avgöra hur kommunikationen skall bedrivas, vilket återigen tyder på att ansvaret sker på lägre nivåer i organisationen.

DevOps-teamen erbjuds en större frihet i att arbeta och implementera nya idéer och lösningar. Flera företag saknade emellertid en tydlig struktur för innovationsarbetet, där medarbetare behövde godkännande för att implementera nya lösningar. En tydlig struktur kan motverka att flaskhalsar uppstår så att arbetet i DevOps-teamen kan fortskrida mer effektivt, vilket kan spegla teamets förmåga att hantera problem och arbeta lösningsorienterat.

De studerade företagen arbetar explicit med att öka förtroendet i DevOps-teamen för att kunna minska beroendet av enskilda individer eller team med specialistkompetens. Trots detta, förekommer det fortfarande beroenden bland företagen. Det har observerats ett samband mellan beroende och förtroende, där förtroendet ökar i takt med att beroendet minskar. För att kunna minska beroenden kan det bli allt viktigare att nya faktorer tas i beaktning redan vid sammansättningen av DevOps-team. Genom att komponera team som besitter en bred, tvärfunktionell kompetens, kan gapet mellan nyckelpersoner minskas. Vidare assisteras DevOps-teamen kollektivt genom stödjande coachning istället för att enskilda ledare styr och ställer, vilket bevisligen skapar förtroende.

6.1 Vidare forskning

Denna studie har ämnat åt att undersöka hur transformellt ledarskap bedrivs i team som tillämpar DevOps. Vi har valt att avgränsa oss från huruvida storleken på teamet påverkar förmågan att bedriva ledarskapet i denna kontext. Vidare forskning kan således undersöka om storleken på teamen kan påverka arbetet.

7 Bilagor

B1 – Svensk intervjuguide

Bakgrund

1. Vad är din roll i företaget?
 - a. Vad har du för bakgrund tidigare?
2. Hur länge har du arbetat som ...?
3. Vad har du för arbetsuppgifter och ansvarsområden?
4. Hur ser organisationsstrukturen ut?
 - a. Hur ser ditt team ut?
 - b. Vilka roller finns i teamet?
 - c. Har någon ett övergripande ansvar? Vad har denne för roll?

Vision

5. Hur arbetar ni med mål?
 - a. Har ni gemensamma eller individuella mål?
 - b. Vem ser till att mål efterlevs? Hur sker uppföljning?
6. Använder ni någon form av mätning?
 - a. Har ni gemensamma måttal för både utveckling och drift?
7. Anser du att alla i teamet delar samma visioner och värderingar?

Kommunikation

8. Hur ser kommunikation ut inom teamen/avdelningarna? T.ex. möten, vilka är delaktiga?
9. Det som tas upp på exempelvis möten, sprids det vidare ut till alla berörda?
10. Hur arbetar ni med feedback från någon form av pipeline?
11. Hur ser arbetet och kommunikationen ut mellan dev och ops?
12. Har utvecklarna någon direkt kundkontakt? Varför/varför inte?
13. Hur involverade är driftpersonal i utvecklingsfasen?

Innovativt och lösningsorienterat arbetssätt

14. Förespråkas det att vara kreativ och arbeta med problemlösning? Hur?
15. Uppmanas personal att lösa problem på egen hand eller förväntas de att få ett godkännande först?
16. Om fel upptäcks, hur hanteras dessa? Vems ansvar är det?
 - a. Delas lärdomarna med teamet?
17. Hur mycket av processerna är automatiserade?
 - a. Finns det några nackdelar med att ha färre manuella kontroller?

Individuell omtanke

18. Förespråkas det att alla kommer med nya förslag på förbättringar? T.ex nya verktyg.
Får alla sin röst hörd?
19. Hur går ni tillväga när det kommer förslag på förbättringar?
20. Hur mottagliga är ni för förändringar?

Övrigt

21. Vad skulle du säga gör just er till en DevOps organisation?
22. Har det någon gång uppstått situationer där ett DevOps-arbetsätt varit ett problem?
 - a. Om ja, hur löste ni det?
 - b. Om nej, tror du det kan komma att uppstå för problem i framtiden?
23. Någon annat du skulle vilja tillägga?
24. Går det bra att vi mailar dig om vi kommer på något ytterligare fråga som kan vara relevant?

B2 – Engelsk intervjuguide

Background

1. What is your role in the company?
 - a. What is your previous background?
2. How long have you been working as...?
3. What is your main tasks and responsibilities?
4. How does the organization structure look like?
 - a. What does your team look like?
 - b. What are the roles within the team?
 - c. Has anyone some kind of responsibility? What is their role?

Vision

5. Are you working with goals?
 - a. Is it collaborative or individual goals?
 - b. Who are responsible for that goals are met?
6. Are you using some kind of measurements?
 - a. Are those common both for operations and development?
7. Do you consider that the whole team share same visions and values?

Communication

8. What does the communication look like within teams and the departments? E.g meetings, who is involved?
9. Topics that come up on example meetings, are they further shared along the people?
10. Are you working with some kind of feedback from a delivery pipeline? How is that processed?
11. How does the work and communication look like between development and operations?
12. Has the developers any direct contact with customer/end-user? Why/Why not?
13. How involved are operations in development?

Innovative and solution-oriented working

14. Is it advocated to be creative and working with problem solving? How?
15. Are staff encouraged to solve the problem on their own or are they expected to get any confirmation first?
16. If errors occur, how are they handled? Whos is responsible?
 - a. Are learning shared within the team?
17. How much of the processes are automated?
 - a. Are there any pros and cons with having less manual controls?

Individual consideration

18. Advocates new suggestions on improvements? e.g new tools. Is everyone getting their voice heard?
19. How do you proceed when new suggestions on improvements comes up?
20. How receptive are you to change?

Other

21. What would you say makes you a DevOps organization?
22. Have any situations arisen where working with DevOps have been a problem?
 - a. If yes, how did you resolve it?
 - b. If no, do you think there could arise any problems in the future?
23. Is there something else you would like to add?
24. Would it be okay if we contact you if we would come up with any additional questions that could be relevant for the topic?

B3 – Transkriberingsprotokoll P1

Information	Färg efter huvudområde
Verksamhet: Mediabransch	(V) Vision = röd
Intervjuobjekt: P1	(K) Kommunikation = grön
Tid och plats: 17-04-2019 via personligt möte	(A) Innovativt och lösningsorienterat arbetssätt = blå
Medverkande: P1, Eric Magnusson och Julia Trulsson	(O) Individuell omtanke = gul

Rad:	Information:	Aspekt:
1	Vad är din roll i företaget?	
2	Jag jobbar som DevOps och har min bakgrund mer på Ops sidan än Dev. Det känns som de flesta av mina kollegor kommer mer från Dev-sidan. Primärt stöttar jag de utvecklingsavdelningarna som finns här i Malmö. Men jobbar egentligen över hela organisationen.	
3	Du nämnde att du främst har tidigare bakgrund inom Ops mer, vad har du haft för roller då?	
4	Jag har egentligen haft lite olika spridda roller. [...] I samma veva flyttades jag bort från IT, och då insåg man att det behövdes den här DevOps-rollen ändå och jag tog hand om det här då ensam för Y. Formellt så tillhörde jag ju avdelningen i Stockholm, men jobbade i princip helt ensam. Eftersom man från början inte ville införliva oss med X så jobbade vi helt annorlunda, med helt andra plattformar, så det var som skilda världar. Sen var det ju P2 som vi tog in lite på IT-sidan, och som visade intresse för de här bitarna. Det gjorde att han sen också blev överflyttad till DevOps-avdelningen då. Så redan under förra året försökte jag få in honom rätt mycket i det här arbetet. Men egentligen var det väl nu vid nyår som han formellt tillhörde avdelningen.	
5	Du pratade innan om att du stöttar utvecklingsteamerna, hur då?	
6	Om man tittar på det vi kallar för NYP-teamet här, som egentligen handhar vår huvudsakliga webbplattformar för Sydsvenskan och HD. Så har de, vad jag tycker då, egentligen en kille som är hyfsat kunnig på Ops och är vass som utvecklare. Så han skulle man ju kunna säga är mer DevOps som sådan, men sen behöver han också hjälp med just rena Ops-bitar som inte han kan. Det är väldigt få som är extremt duktig på både och. Men sedan finns det ju bland utvecklarna de som är back-end och front-end och några som är mer åt formgivningshållet och så. Men de räcker ju inte riktigt till på Ops-sidan alls då. Så på det viset har det varit min del i det hela, att göra den biten.	K3
7	Så när utvecklarna känner att de inte har tillräckligt med kunskap om Ops-biten, är det då du som hjälper dem?	

8	Ja, och sen är det även löpande saker som ska fixas och donas. [...]	K3
9	Vissa säger att man kan se DevOps som en roll eller avdelning medan andra säger att det är mer ett typ av arbetssätt där delar upp sig i team och att man är med mer från start till slut av produkten. Hur skulle du själv definiera DevOps?	
10	För mig är DevOps mer att alla personer behöver inte vara från hela spännvidden men man är team där man gör allt från operations till utveckling och så vidare. Sedan är frågan, var drar man gränsen? Operations, vad är då det kan man också fråga sig. [...]	O3
11	Hur involverade är utvecklarna i driften och vice versa?	
12	<p>De flesta av våra utvecklare de mer eller mindre bara skriver sin kod, och sen när de har gjort commit på koden och pushat den till GitHub så blir den ju automatiskt deployad. Och de vet inte alltid hur. De gör de stegen, och sen kommer det ut på sajten eller vart det nu ska. I de allra flesta fall så har de inte någon aning om hur de här sakerna fungerar. Men det finns ju då framförallt en kille bland utvecklarna som förstår det tänket. Annars är det ju vi som hjälper till. Sedan använder vi ju oss av Amazon, cloud formation för att sätta upp miljöerna, och sen har vi ju code pipeline, och de här sakerna för att ta hand om koden och köra ut den. Och i den mån är det ju en del kodknackande för mig också.</p> <p>En del jag har pratat med och försökt diskutera "vad är DevOps och vad är inte DevOps" har ju sagt att man håller på mycket med kod också. Och ja, även om det inte är direkt en huvudsyssla som bygger själva sajten. Det är ju ganska mycket andra jobb som snurrar för att sköta saker med operations där vi också knackar kod för att få de sakerna att funka som vi vill.</p>	K1, K3, O3
13	Så när utvecklarna skriver kod och deployar den, har de i driften ingen direkt kommunikation med utvecklarna om vad som händer?	
14	<p>Jo men det har vi. Och det är därför jag tycker att vi jobbar som DevOps ändå. För jag sitter ju t.ex. mitt bland utvecklarna, mitt i smeten. Jag är ju med på utvecklarnas veckomöten varje vecka så jag har ju god insyn i vad de gör och de vet vad jag gör. Så jag känner mig som en i gänget här.</p> <p>Mina kollegor i Stockholm har ju suttit som en avdelning så och har inte haft så tight integration i utvecklingsteamet men de börjar nu fundera om det inte är rätt så som vi gör här. Men där är de ju fler utvecklingsgång än vad det finns anställda på DevOps här. Men bara en sån sak som att vara med på de olika utvecklingsavdelningarnas möte och snappa upp saker tycker jag gör att det skulle kunna bli bättre.</p>	K1, K3
15	Det du får ta del av på de mötena, är det något du förmedlar vidare till annan personal i driften?	
16	Ja, nu har man ju ändå tyckt att nu måste vi lite mer gå mot att vi använder samma plattformar. Och vi måste mer kunna lära av varandra. Så nu gör jag mycket mer jobb åt utvecklingsgången i Stockholm. Jag behöver komma in och förstå hur de har satt upp saker. Om de använder open shift som plattform säger du en sak, men bara för att man kan open shift betyder det inte att man exakt hur de har satt upp saker och så. I den processen jag befinner mig just måste jag försöka förstå bättre hur de har satt upp saker. Naturligtvis är inte allt dokumenterat, även om mycket är det. Målet är väl också att mina kollegor där uppe ska kunna hjälpa till här nere också. Så vi försöker ju föra runt kunskapen. Och det är ju en nackdel då med att om varje person på DevOps blir utplacerad med ett utvecklingsgång kan det vara svårare ibland att hålla oss ihop och hålla oss uppdaterade. Rätt eller fel, det är inte lätt alltid. Det finns en uttalad strategi om att vi	K1, K3, V1

	<p>ska försöka gå mer åt att använda samma tekniska plattformar. Sedan finns det ju alltid motstånd mot det också. [...] det är ju alltid enklare att arbeta om man sitter fysiskt nära.</p>	
17	<p>Den bilden vi har av DevOps är väl ändå att utvecklare och driften ska samarbeta. Tror du det blir svårare att praktisera DevOps om man outsourcar vissa delar?</p>	
18	<p>Ja, jag upplever det som att fysiska avstånd och ibland också kulturella avstånd och språk, allting sånt gör ju det svårare. Det är inte omöjligt men den absolut lättaste avdelningen jag har att samarbeta med är de utvecklarna som sitter vid skrivborden runt omkring mig. Bara det att kunna sitta och höra vad de pratar om, och vad de har för problem just nu. Man snappar upp så mycket som man inte gör annars. Det är min erfarenhet av det hela.</p>	K1
19	<p>Arbetar ni mer särskilda mål för Dev och Ops eller har ni gemensamma mål?</p>	
20	<p>Utvecklarna har ju mål på lång- och kort sikt, de försöker ju koppla upp sig till företagets mål givetvis. Men de varierar ju också. Organisationen har ju försökt med lite olika saker, som att centralisera vissa saker. De utvecklingsgång som sitter här skulle ju kunna sitta i Stockholm, men det har man ju insett att "nej det funkar ju inte". För att tidningarna som sådana har olika mål, även om både Y och Z är del av organisationen. Så betyder det ju inte att de olika redaktionerna har samma mål eller vill samma saker. Så man vill att utvecklingsteamet sitter väldigt nära redaktionerna. För att de ska ju också stödja vad redaktionen vill. Så ofta kan man säga att utvecklarnas mål styrs väldigt mycket av vad redaktionen vill prioritera nu. Vill redaktionen att vi ska jobba mer mot små orter, och inte bara huvudorterna Malmö, Helsingborg och så, kanske det blir projekt i form av att vi ska kunna ha lokala nyheter lättare osv. De projekt som hela tiden startar är oftast spunna ur vad redaktionen för närvarande vill göra. [...] Den typen av mål kan utvecklarna inte alltid sätta själva, utan det styrs utifrån vad företaget vill, vad redaktionen har beslutat sig för och hur ska man jobba.</p>	V1, K2
21	<p>Det har ju med planeringsarbete att göra. Om vi tar ditt exempel att ni ska nå till mindre orter, hur skulle du säga mer att ett sådant planeringsarbete går till? Är det redaktionen som sätter sig ned med utvecklarna och berättar vad de vill ha?</p>	
22	<p>Jag får känslan av att här, iallafall på den här tidningen, att ledningen för redaktionen och ledningen för utvecklarna de är rätt så tigha. Och det funkar även ner på de övriga medarbetarna på redaktionen och enskilda utvecklare. Det verktyg som skribenterna använder sig för att skriva, det byttes ut för att det är en extern produkt som införlivas och den uppgraderades och då skulle det komma till lite nya saker. Då skulle redaktionen idag ha en möte kring det, och direkt i den kanalen som utvecklarna använder "vi ska ha ett möte kring det här, vilka vill komma?". Och då var direkt några utvecklare som sa "vi kommer". Så det känns som att det faktiskt funkar bra. Och den biten hade inte funkat att göra det på distans.</p>	K2, K1
23	<p>Sätter ni upp olika team för olika projekt eller har ni ändå era fasta avdelningar? Hur fördelar ni arbetet?</p>	
24	<p>Nu är vi ju inte så många här nere, vi är ett väldigt litet team, och de är väldigt kopplade till just vår produkt. Uppe i Stockholm finns det vissa team som är mer specialiserade. Och eftersom vi nu knyts närmare så får vi också besked på att vissa saker ska vi utnyttja de centrala resurserna och inte lägga tid på själva. Ett exempel är att man samlar ihop data både från oss och från alla produkterna inom familjen. Kunskap om kunder, kunskap om läsare osv. Det är mycket GDPR inblandat där. Som en enda stor kunskapsbank, och sen så är det ju fritt för alla att också då dra nytta av den. Det är ju en sån grej som vi får klara</p>	O3, K2, K3

	direktiv om att det ska vi inte pyssla med själva. Det hade vi inte riktigt kunnat klara av heller med den personalstyrkan. [...]	
25	Vilken roll skulle du säga att driftarbetare har i detta?	
26	De är ju ofta med. Men vi är ju inte så många, det känns som det är väldigt platt struktur på det viset.	K3
27	Så ni kan göra det medvetet att ni släpper kod som ni vet inte är tillräckligt bra?	
28	<p>Om ska släppa något nytt som ska ha fem features, en feature är inte färdig men vi släpper den iallafall med de här fyra featuren. Man behöver inte vänta på att vi har allting helt klart. De behöver inte betyda att man är medveten om buggar, men att det är en feature av fem som hade varit gött att ha med också - men det är viktigare att få ut de fyra än ingen alls. Det är viktigare än att vänta, att få ut det i tid.</p> <p>Det är ju för det mesta helt odramatiskt att släppa något nytt, man gör det hela tiden. Och det tycker jag är trevligt.</p>	A2, O1
29	Dessa releaser ni gör, är de någon del av ett planeringsarbete som ni gjort tidigare eller arbetar ni hela tiden med det som kommer från redaktionen?	
30	<p>En del saker är ju tidsbestämt åt sådant som vi inte beror på. Vi kanske ska göra någon speciell valsajt, vi kan ju inte skjuta på valet. Valet blir ju ett visst datum och då måste det vara klart. Då blir det ju mycket mer planering för att få klart allt i tid och så. Ibland kommer det ju bara idéer, typ "det skulle vara väldigt trevligt att göra någonting sånt här". Och då kanske det är något mer som utvecklarna har lite vilande när de har tid över. Då blir det mer att vi släpper det när vi känner för det.</p> <p>Då har jag märkt att man som avdelning kanske planerar in, tex efter jul och nyår, vad har vi som vi ska göra under året. Vad hinner vi med före sommaren? Så lägger man kanske upp en tre-fyra större grejer som ska hinnas med. Och vissa saker är ju mer att de behöver vara klart vissa datum. Och så vet man att de här smågrejerna, det kommer tillkomma grejer, så man är väl rätt så agil på det sättet att man tillåter att det ploppar upp idéer.</p>	A1, V1
31	Så det utsatta målet kan förändras hela tiden?	
32	Ja men det gör det.	V1
33	Det du pratade om innan, att ni kan hitta nya saker som hade kunnat vara nice att ha i produkten, är det något som organisationen i sig förespråkar? D.v.s. att ni tänker lite kreativt och gör saker som kan gynna produkten i slutändan.	
34	Ja men det känns som det är så vi jobbar. Det uppmuntras att man tänker till. Det är en sån grej som jag kan märka, och som flera har sagt också, att externa utvecklingsteam och framförallt i andra länder med annan kultur, då är det väldigt sånt att man får exakt vad man ska göra och då gör man exakt det. Även om man vet att det här kommer att skita sig så säger man ingenting för man kör efter kartan - det är det här vi ska bygga och då bygger man det. Och så är det definitivt inte in house här. Jag kanske inte alltid kommer så ofta med saker för sajts funktion och så, även om det händer också. Men jag kan ju komma till utvecklarna och säga "det här håller inte, ni måste tänka om. Så som ni bygger nu så orkar snart inte databaserna med. Titta här. Ni måste ha gjort något	A1, K2, A2

	<p>feltänk här. Lasten bara sticker iväg här nu sen ni släppte de här nya grejerna”. Och då kan jag ju få säga till utvecklarna, “tillbaks till ritbordet här nu. Vi måste jobba tillsammans för att lösa det här”. Så jag kommer kanske mer från det hållet, för jag håller ju koll på pengarna också. Vad kostar det utvecklarna att sätta upp nya saker.</p>	
35	<p>Så är det din roll lite att övervaka lite och se till att rätt grejer görs och att det är inom rimliga gränser?</p>	
36	<p>Jag håller ju koll på driftspengarna om man nu kan säga så. Och det är ju inte alla pengar, men det kan ju vara så att vi ska sätta upp ett antal nya maskiner eller vad det nu är. [...] Och då måste jag ju stämna av med utvecklarna, om vi reserverar x antal instanser av den här typen under ett år. Hur ser det ut, vad kommer ni göra, kommer ni förändra någonting. Så det är ju en del av det här att ha lite känsla för vart är utvecklingen på väg. Om jag reserverar upp ett antal instanser för att sätta upp ett antal Postres-databaser eller MySQL-databaser och så visar det sig “nej vi ska byta från det. Vi ska inte ha sådana databaser. Produkten kommer se helt annorlunda ut, vi kommer bygga det på Solar instanser eller något annat”. Och det funkar ju inte, det måste till den här dialogen. Och den tycker jag är viktig.</p>	K1
37	<p>Så du är även med och pratar med utvecklarna, även om du är mer ansvarig för driften?</p>	
38	<p>Absolut. Vi jobbar som ett team ihop. Sen försöker jag hålla lite koll på pengarpåsen och driftmiljön och allt sånt håller sig och så.</p>	K1
39	<p>Men ändå har utvecklare och personal fritt ansvar att agera om de hittar förbättringsområden, eller måste de stämna av med dig först?</p>	
40	<p>Jag skulle säga att vi har väldigt fria tyglar. Vi har ju kommit så pass tight att utvecklare kommer till mig också: “jag har en idé här. Skulle vi kunna göra såhär?”. Så tittar vi på det tillsammans. Det tycker jag är jättebra.</p>	A1
41	<p>Om problem skulle uppstå vid driftsättning, hur går ett sådant arbete tillväga? Får utvecklarna ta del av det som gått fel?</p>	
42	<p>Det är utvecklarna som har ansvar för det de släpper ut. Vi tar inget ansvar för det de släpper ut. Släpper de ut kod fredag eftermiddag så är det de som får jobba på helgen. Sen är det ju inte alltid så enkelt naturligtvis. Och inträffar ett fel så är det ju inte alltid man från början att det beror på en viss deploy, eller om det beror på att det som finns i back-end gör att det fallerar. Det är inte alltid man kan säga det. Och i de situationerna är det guld värt att vi inte har de här höga murarna emellan och att vi är ett team, vi jobbar ihop. Och det funkar faktiskt.</p>	O3, K3
43	<p>Är det ni i driften som upptäcker felet och kommunicerar ut det?</p>	
44	<p>Om vi släpper en ny kod och det skiter sig, så ibland kan det vara jag som upptäcker det. För att ibland blir det väldigt hög last eller jag ser att klustret helt plötsligt bara skalar upp, det är något som inte stämmer. Det kan vara utvecklaren själv för de har ju lite koll på om koden de släpper verkligen funkar. De kan ju också vara någon från redaktionen hör av sig och säger att något inte ser rätt ut. Det kan komma lite överallt ifrån.</p>	K2, O3
45	<p>Så där är det lite mer gemensamt ansvar att se till att problem upptäcks?</p>	

46	<p>Ja, eller den som upptäcker att något är knas hojtar. Men utvecklarna har ju fortfarande ansvar för sin kod. De kan ju även backa koden och lägga på tidigare versioner och så. Sen är det ju bitar som faller mer på mig, men det är ju inte så mycket ut på sajten. Men vi använder ju en del andra verktyg som Prometheus, Grafana, och sådana saker för att hålla koll på statistik och så. Och det är ju inte kod som jag skriver, mer anpassar det. Och skiter det sig där någonting då får jag ju utvecklarna på mig "kan inte se längre". Så har jag släppt ut något som är skit får jag ju ta ansvar för det. Men det är ju inte ut på sajten som så.</p>	O3, V2, K2, O3
47	<p>Du nämnde också att ni övervakar arbetet, sker det någon typ av mätning också? Hur systemet mår, eller hur många deployments ni gör per dag eller vecka.</p>	
48	<p>Mina kollegor i Stockholm kan nog se precis hur många deploys de gör, undrar om vi kan se det så enkelt. Här nere har vi en dashboard där vi kan se alla deployer som håller på just nu, det är ju ofta flera deployer som går samtidigt. Så vi har det för att se vad som rullar ut och om det går bra eller inte.</p>	V2
49	<p>Har alla tillgång till dashboarden?</p>	
50	<p>Jaja, den sitter på stora skärmar. Likadant när jag övervakar certifikat och en del andra saker så har jag också det på en stor dashboard som alla kan se. Så visar den tex rött är det något som inte är som det ska. Så det är inte bara jag som kan se det.</p>	V2, K3, K2
51	<p>Är det för att alla ska kunna ta del av arbetet som sker?</p>	
52	<p>Ja. Men just hur sajterna mår övervakas ju på massor av mätpunkter, det finns också uppsatt på stora skärmar så att alla kan följa det. Och de är ju de vi valt ut som alla ska se, sen kan det ju vem som helst gå in i vår Grafana och själv plocka fram och göra sina egna grafer på det man är intresserad av, eller just sina applikationer. Men i de fall Grafana tar mätvärden från t.ex. Amazon, som de släpper från sig, eller vad som ligger i Prometheus. [...] Prometheus är ett vanligt verktyg att använda, där man kan fråga en applikation om statistik. T.ex. "hur många anrop har du haft den senaste...?", "hur mycket minne tar det...?", och bla bla. Men då måste ju applikationen som ska släppa ifrån sig de här mätvärdena hålla reda på det, och det är ju upp till utvecklaren då. Har de gjort sin kod, så får de ju lägga in så att den håller reda på de här sakerna.</p> <p>Så att Prometheus kan ställa de här frågorna och få något vettigt tillbaka. För Prometheus kan ju inte trolla. Så har bara utvecklarna gjort det de ska, och de sakerna som de är intresserade av att få ut mätvärden på. Då blir det i stort sett per automatik tillagt så att Prometheus tar ut de värdena, och så kan man få grafer på det och man kan gå in och söka och fråga efter saker. Så det finns ju alla möjligheter att ta fram all möjlig statistik. I 9 fall av 10 är det ju utvecklaren själv som vet vad som är intressant att fråga den applikationen om. Det är ju inte alltid jag vet vad som är intressant i en specifik applikation. Jag kan ju fråga efter "hur mycket trafik har den haft?". Men sedan alla små mätvärden kanske inte jag direkt kan avgöra om det är högt eller lågt eller normalt.</p>	V2, O3
53	<p>Använder ni er av någon form av belöning? Dvs hur uppmuntras personal när de gjort något bra?</p>	
54	<p>Inget formellt så, men det tycker jag. Det kommer ibland chefer från andra avdelningar och säger "det där ni gjorde, det blev jättebra". Det finns en ganska bra stämning på det viset. Och inom gruppen kan vi ju också ge beröm. Det är ju oftast vårt mycket mer än om man får någon formell gratulation. [...] En sådan grej var ju som vi kallar för "Slumpis". Det debatterades i många år, och såklart fortfarande, om det här med betalväggar på sajten. Ska man betala för att se innehållet eller inte. Och om man ska betala, vad ska man betala för. Jag minns inte exakt, men jag tror idén startade hos en av våra utvecklare.</p>	O2, O1, A1

	<p>Visst material kanske vi vill ha bakom betalvägg, och visst material ska kanske aldrig vara det. [...] Men de flesta artiklar är slumpade. Det är en väldigt enkel algoritm egentligen. Och då startade man med det väldigt enkelt och det blev ju väldigt uppskattat, bara som ett exempel. Det har man ju jobbat vidare med och algoritmen kan ta hänsyn till flera saker.</p>	
55	Du nämnde att arbetet kanske inte blir lika bra alltid, är det ett problem?	
56	<p>Det kan vara ett mindset. Det kan bli lite mer slapt. Men tekniskt annars, att ha deployer ofta, ser jag inte som ett stort bekymmer. Det finns så mycket teknik kring det där som underlättar. Men det kan ju vara mer mentalt att man kanske inte alltid bryr sig lika mycket. Ibland kan det vara en fördel också, att vi behöver inte bry oss om den här femte featuren. Det är både positivt och negativt.</p>	A2
57	Tror du att det traditionella projektledarrollen kommer att ersättas eller försvinna mer om man arbetar med DevOps? Du säger ju att arbetet och roller är mer flytande och att man har väldigt fria tyglar och så vidare.	
58	<p>Det har man ju. Bland våra utvecklare här nere finns det ju en som är chef ändå. Han har ju många gånger sista ordet, men han väljer att inte alltid lägga sig i. Vilket är bra. Men han kan ju gå in och säga "nej, nu får alla teamen släppa det ni har i händerna för den här grejen måste ut till nästa vecka". Men det gör han ju väldigt restriktivt, han försöker ju inte styra och ställa. Det känns som de här veckomötena varje måndag, där styrs väldigt mycket sånt upp. Alla får en känsla för vad det är som gäller. Jag tror att den rollen kommer ändå behövas, någon som håller ihop gänget. Jag tror inte att en sådan här avdelningar ska vara för stor heller, för då tror jag att det försvinner. Det blir svårt att jobba på det här sättet om avdelningen är för stor, det tror jag. Det här är ändå en avdelning där alla syns, man kan liksom inte gömma sig.</p>	O2, K1, V1, O1
59	Tycker du att det är viktigt när man arbetar med DevOps, att alla syns och alla är med?	
60	<p>Ja, på något sätt spelar det nog inte så stor roll hur man jobbar, det är alltid viktigt att alla syns. För i väldigt stora organisationer är det personer som inte syns och inte blir sedda, mår oftast inte speciellt bra heller. För de enskilda individerna är det viktigt att bli sedda. Och mår individerna bra, så blir det också bättre. Sen om det också är för stort så kan det också bli lite spretigt, även om individerna tycker att de gör bra ifrån sig och att de jobbar hårt så kanske det blir att man inte är med på de gemensamma mål man har. Det är mina erfarenheter.</p>	O1, O2, V1
61	Finns det något du skulle vilja tillägga när man ser till ledarskap och ledningsarbete i en DevOps-kontext som vi inte redan tagit upp?	
62	Nej, jag tycker vi har gått igenom mycket. Jag kommer inte på något mer.	

B4 – Transkriberingsprotokoll P2

Information	Färg efter huvudområde
Verksamhet: Mediabransch	(V) Vision = röd
Intervjuobjekt: P2	(K) Kommunikation = grön
Tid och plats: 17-04-2019 via personligt möte	(A) Innovativt och lösningsorienterat arbetssätt = blå
Medverkande: P2, Eric Magnusson och Julia Trulsson	(O) Individuell omtanke = gul

Rad:	Information:	Aspekt:
1	Vad du har för roll här? Och vad är din bakgrund innan du kom hit?	
2	Rollen jag har för tillfället är DevOps, eller plattformsteamet. Jag är ganska ny i denna positionen då jag blev överflyttad från IT-driften i februari. Så 1,5-2,5 månad. Innan dess så satt jag på IT-supporten och innan jag dess så pluggade jag systemspecialist inom Linux. Så det är min bakgrund.	
3	Vad skulle du säga är största skillnaden på det jobbet du hade innan?	
4	Mer scriptning och programmering skulle jag säga. Mer kontakt med utvecklare.	K1
5	Hur ser ditt team ut? Vilka jobbar du nära?	
6	Först och främst P1 som ni träffade innan. Sen så har vi en plattformsteam uppe i Stockholm som sysslar med och DI, DN, Expressen, Mitt Media osv. Våra miljöer ser ju väldigt annorlunda ut för Stockholmsmiljöer. Men där försöker vi verkligen mixa så att vi verkligen jobbar emot alla miljöer.	
7	Alla dagar kanske inte ser likadana ut? Har du några typiska arbetsuppgifter? Några ansvarsområden?	
8	Ja, för tillfället denna vecka så är man något som heter att vara 'Hatten'. Då har man en slack-kanal där utvecklare skriver om problem och så hjälper man dom med dom problemen. Så gott man kan.	O2, O3 A2
9	Är det problem när man ska driftsätta eller under utvecklingen?	
10	Det kan vara allt möjligt, deras kod, deras containrar som har satt upp i ett Open-shift kluster som Stockholm har. Typiskt sådana frågor. Eller öppna nätverksvägar eller så.	

11	Vad är syftet med 'Hatten'?	
12	Det är han eller hon som har ansvar att hjälpa dom, utvecklarna då. Sen nästa vecka tar någon annan i teamet över. Sen försöker alla hjälpas åt när man har det relativt lugnt. Förutom det är Legacy-IT mitt ansvar. [...]	O1,O2, O3, K3
13	Kan man se det som ett övergripande ansvar eller mer som en coach?	
14	Det är mer som en support skulle jag säga. För inte för vanliga IT-frågor såsom hur jag loggar in på mailen. Lite mer tekniska.	O2
15	Arbetar ni med några mål, från ledning eller gruppen?	
16	Vi arbetar vad jag förstått det hittills att vårt mål är att det ska vara enkelt, smidigt. Typ från t.ex. utvecklare att skriva sin app, klicka på en knapp och publicera appen.	V1
17	Då tänker ni mer övergripande för både utvecklare och drift?	
18	För hela organisationen. Mer gemensamma mål.	V1
19	Det funkar bra?	
20	Ja. Man har ju alltid att säga till om. Jag tycker att vi skulle göra på det här sättet istället och är det då att det är en bättre ide så brukar det oftast gå åt det hållet. Alla i företaget får sin röst hörd.	O1, A1
21	Ni arbetar med olika projekt och så, hur ser det ut när ni ska starta upp ett nytt projekt? Vilka är med i uppstart och slutet?	
22	Man har en beställare, som oftast går till vår team-leader. Jag skulle vilja ha en server med en tjänst på. Då gör man en ticket på Github och så har vi morgonmöte varje morgon och så tar man upp det då. Den här personen skulle vilja ha detta, vem vill dra i det? Vem vill kolla hur mycket tid som behövs läggas ner på det? Och sen tar man ett beslut på det.	K1, O3
23	Så det är mycket eget ansvar om vem som är bäst lämpad för uppdraget?	
24	Ja, men sen så kan det också va att du har bäst koll på området och detta projektet relaterat till det området, så ta du det Kalle.	O1
25	Så ni handplockar lite efter erfarenheter?	
26	Ja	
27	Du nämnde team leader, vem är team leader? Vad har den personen för roll?	

28	Team leader är ju vår närmsta chef, vi vanliga anställda. Han är där för att guida oss, vägleda, men sköter också mycket ekonomi. Sen så har ju han en chef.	O2
29	Men är det han ni vänder er till om det är någonting?	
30	Ja, precis. Sen så har vi en tech leader som har bäst koll på de tekniska lösningarna. Oftast han man kan rådfråga vid speciella frågetecken.	K3
31	Är det några mätetal ni brukar ha koll på efter sträva efter, hur ofta ni deployar tex? Eller hur arbetar ni? Är det mer ansvarig som har koll på de delarna?	
32	Nej, vi har ju en gemensam kanal för teamet i slack där vi skriver nu deployar vi detta här för det ändamålet. Har någon något emot det om något skulle vara ändrat? Alla lär sig.	V2, K1, K2, O1, K3
33	Hur är det om det uppstår problem i koden eller driftsättningen? Hur ser arbetet ut där, är det den som har kodat som har ansvar eller gemensamt?	
34	Det är ett gemensamt ansvar att den här personen som stött problemet lär sig och fixar det. Jag som är väldigt ny ställer ju frågor 20 ggr om dagen nästan. Men då är det oftast 2-3 personer som försöker vägleda en. "Du har fel i den raden eller gör såhär istället"	O3, O2, K3
35	Just för att skapa lite mer lärande inom organisationen?	
36	Ja precis. Och sen någon annan i teamet som inte alls har koll kan följa diskussionen och lära sig.	A2, K3
37	Så det är ett mer övergripande tankesätt? Inte bara för att du är ny?	
38	Det är samma för alla. Vi försöker ju lära utvecklarna också. T.ex. att öppna upp port 80 på nätverket för deras projekt. "Ja men såhär gör du" okej, jag gör det och förklarar stegvis.	V1, K3
39	Finns det någon form av mätning? Har ni några gemensamma mätetal på både utveckling och drift?	
40	Nej.	V2
41	Är driften involverade i utvecklingen och vice versa? Eller är det att utvecklarna gör sitt jobb och sen passar över till driften?	
42	Ja. Nae, som sagt så har vi ju miljöer för utvecklarna där dom själva kan göra väldigt mycket så att de slipper komma och säga "vi behöver detta" så får vi sätta upp det. Och det är det här Open-shift klustret som jag prata om. Sen är det annorlunda här i syd. Här kan man ju få en fråga, "vi behöver en sån här stor maskin för detta ändamålet" och då blir det mer som ett ärende. Men det är mer sällan.	K1

43	Det är mycket på eget bevåg också att göra saker och så inom teamet. Om ni märker att ni arbetar med ett projekt och tycka att vi borde göra såhär istället för ett bättre resultat. Har du tillåtelsen att göra det själv eller krävs något godkännande att göra ändringar och så?	
44	Jag är ju den sortens person som bollar allt med min chef.	
45	Är det mer från din sida att du vill ha bekräftelse?	
46	Det är nog från min sida. Så slipper man göra fel och göra om allt igen.	
47	Men det uppmuntras att du gärna kommer med idéer och så?	
48	Ja, det skulle jag säga.	A1
49	Vi hade en fråga till P1 om det finns någon form av belöning när ni gjort någonting bra?	
50	Det är i så fall vad jag sett hittills i vår kanal att man har hittat ett problem och lyckats åtgärda det. så skriver man det i kanalen så att alla ska se. Och då brukar det vara respons på det.	A2, K1, K3, K2
51	Då kan man lösa problem som andra har orsakat?	
52	Ja, eller att flera är och kollar på det samtidigt.	A2
53	Arbetar ni aktivt för att upptäcka fel?	
54	Ja, absolut. Just när man gör manuella jobb att man ser någonting som inte stämmer. Eller att utvecklare hör av sig att "nja, det gick inte att deploya en container" "klustret skalar inte upp" Ja men då är det en config-fil någonstans.	A2, K1
55	Du nämnde innan att ni hade de här morgonmötena. Vilka är med på dessa?	
56	Det är hela teamet.	K1, V1
57	Och vilka består det av?	
58	Det består av mig och P1 och teamet i Stockholm, team leader och tech leader och våra andra kollegor.	
59	Då är det ni som har titeln DevOps och inte development eller operations specifikt?	
60	Nej, utan där diskuterar vi vad vi ska göra under dagen och hur brådskande vissa problem är. Lite delegering.	K1

61	Den informationen ni tar del av på morgonmötet, är det något ni delar med er av till utveckling eller operations?	
62	Ja, det kan det mycket väl vara om man har problem med att hjälpa en utvecklare så kan man ställa frågan på morgonen och kanske få en lösning och då delar man ju med sig av det till utvecklarna.	K2, K3
63	Hur ser arbetet ut mellan utvecklare och drift? Ni är ju mer mellanhanden som kan lite både och, men har dom avdelningarna nära kontakt?	
64	Nu har vi ju inte drift just här utan allt är i tech-desk som de kallar det för. Det är väl mer IT-support. Men innan så kan det ha varit lite samarbete mellan dessa. T.ex. när det kommer till legacy-IT eller att man behövde öppna upp en port inifrån huset eller datahallarna någonstans. Annars väldigt lite om man jämfört med DevOps och utvecklare.	K1
65	Hur nära i sig arbetar utvecklare med kund?	
66	Väldigt nära skulle jag säga. Ett team av utvecklare som ska driftsätta en app t.ex. kan ju ta emot förslag och problem från kunden direkt.	K1, K2, A2
67	Finns det någon del av att driftpersonalen är involverade i utvecklingsuppgifter?	
68	Nej inte på så sätt. Utan det är just då om utvecklaren får problem i miljöerna så att vi hjälper till med det. Nu pratar jag mycket från Stockholmsperspektiv.	K1, K3, O1
69	Om vi tänker DevOps i det stora hela, vad skulle du säga gör er till en DevOps-organisation?	
70	Nej men just det att man har både bitarna, man har utvecklingserfarenhet att man kodar och programmerar lite, men också operations. Man sätter upp miljöer fast på ett utvecklarsätt. Det skulle jag vilja säga är DevOps. Stockholm säger att vi inte är DevOps utan kallar det plattformsteamet istället.	O3
71	Har du någon gång uppstått detta arbetssätt varit ett problem?	
72	Nej, jag har bara tyckt det varit roligt. Inte stött på något.	
73	När man utvecklar i en DevOps-organisation, tar man mer hänsyn till operations då?	
74	Det gör man nog ja. För man måste tänka på hela bilden istället för bara att sitta och utveckla och sen är det någon annans problem att driftsätta.	V1
75	Något annat du känner du vill tillägga, hur ledningsarbetet fungerar?	

76	Nej, just den biten är svår. Tror jag nämnt allt så som jag uppfattat det hitintills. Ja det är vår team leader som har den övergripande bilden så att säga eller som tar emot, delegerar om man kan säga så.	
77	Om det inte fungerar som det ska?	
78	Då brukar vi ofta få frågan om det inte är så att de själv inte ser något ser något fel i sin kod.	K2
79	Då kör ni en dialog med kund?	
80	Nej, med utvecklarna. Vi har väldigt lite kontakt med kund.	K1
81	Så ni frågar utvecklarna om de upptäckt fel?	
82	Oftast så brukar det vara den här kanalen där 'Hatten' får komma in och så skriver utvecklarna att de försökt deploya en applikation men det gick inte. Då är det en ganska ny utvecklare om han säger så. Men då går vi in i projektet i open-shift, kollar i loggarna, och ser kanske att det är ett syntaxfel i koden. Detta får du lösa då det inte är vi som byggt applikationen, det är dem. Eller för lite minne i klustret, okej då får vi öka det.	O2
83	Så ni hjälper dom att felsöka problemet?	
84	Ja det kan vi mycket väl göra.	A2
85	P1 nämnde någon dashboard?	
86	Ja, dashboard så används det för sidvisningar bland annat. Men det är för hela organisationen här. En annan dashboard är Prometheus som används för att övervaka våra olika tjänster och skapa applikationer som man kallar det för. Att den hämtar in information om t.ex. hur hårt applikationen jobbar, eller om den har kraschat. Om det har kraschat så varnar Prometheus så tar alert-manager och skickar meddelande i slack. Eller så har man Grafana så visar den mer grafiskt.	V2
87	Det senare du pratade om, är detta något alla tar del om?	
88	Ja, i Stockholm är det så att skaparen utvecklar ett projekt så får han automatiskt en Prometheus-setup och Grafana-setup så kan han bara logga in i tjänsterna och kolla status.	K3, K1
89	Kan det vara något som en team leader kan gå igenom för övervakning?	
90	Nej, men däremot kan det komma en hel del larm, att applikationen eller containern har problem med att komma igång. Du undersöker man ju varför.	K2
91	Vem är det som får alarmet?	

92	Alla i teamet. Sen har vi öppna kanaler också där det mer kommer generella larm tex om en container har svårt att starta så beror det ofta på klustret och då är det i teamet som ska ta hand om det.	K2
93	Så när ni får larm, vem ligger ansvaret på att lösa?	
94	Det är den som känner sig mest mogen.	O1, O3
95	Då tror jag vi har fått med allt. Du känner inget mer att tillägga?	
96	Nej, inte mer än att jag kan säga att DevOps, även om det är ganska nytt yrke, är det väldigt spännande och skoj. Bra variation. Man behöver absolut större övergripande, T-shaped knowledge.	O3
97	Även om man jobbar inom operations?	
98	Ja, det är mycket man kan göra med kodning. En utvecklare som inte kanske har koll på Prometheus och Grafana så kan man hjälpa till att skriva exports som läser av applikationen.	O3, K3

B5 – Transkriberingsprotokoll P3

Information	Färg efter huvudområde
Verksamhet: Affärssystem	(V) Vision = röd
Intervjuobjekt: P3	(K) Kommunikation = grön
Tid och plats: 24-04-2019 via digital telefoni	(A) Innovativt och lösningsorienterat arbetssätt = blå
Medverkande: P3, Eric Magnusson och Julia Trulsson	(O) Individuell omtanke = gul

Rad:	Information:	Aspekt:
1	What is your role in the company and what was your previous background?	
2	My role is like an entrepreneur so basically there is this one corporation that I seek out somebody who has ever build business before. They are kind of looking for to find another department like a digital division, so it's like a start-up within the corporation.	
3	So, what's your responsibilities or tasks do you have?	
4	Basically everything, I kind of build everything from scratch. It's basically a digital division so I have to come up with software as a service products. So, I manage a team where we are building an ERP system. And my background, I have been in software development mostly. So, I have technical backgrounds, and project management, and in the last five years I have been in start-ups.	
5	Do you work in any team?	
6	We are a 20+ people right now.	
7	What are the different roles in this team?	
8	We have functional, product, developers, automation engineer, and DevOps. That's all, I think. And designer.	K1
9	Does anyone have more responsibility, a leader or something?	
10	Basically, I have struct the team into several structures. So, I have an ERP team, with its own leader. And the is another HR team with its own leader. And DevOps with its own leader. So basically, three large teams.	K1

11	Are you working with some kind of goals?	
12	Yes, as I mentioned before we are supposed to launch a ERP product. And device bias, and we actually hold the intellectual property of it.	V1
13	So, do you have any goals that are collaborative or is it individual goals for your team?	
14	It is for the whole team actually.	V1
15	Who is responsible for the goals, that you are achieving them?	
16	We have larger goal, like there is the ultimate goal to launch the product. And then of course I have divided into several sub goals for each of the teams. Like for ERP, they have different goals. And for DevOps they have different goals as well because they use very different metrics in each team.	V1, V2
17	What kind of metrics?	
18	For DevOps, usually like SLA and number of defects.	V2
19	Do you measure something that is specific for ops or dev?	
20	Yes, I try to make everything measurable so I can track its performance.	V2
21	Do you believe the whole team share the same visions and values to reach these goals?	
22	Yes, supposedly.	V1
23	How does the communication work between the teams? Do you have any daily meetings? Who is involved?	
24	Yes, we implement HL software development specifically Scrum. And in Scrum we have daily meetings. Back when I was still in Jakarta, we had refinement meetings every week with the board of directors.	K1
25	Are people from lower levels at these meetings as well? Like developers and workers from the operations team.	
26	Yes, basically I try to involve all the team leaders, but I have this one spot that is dedicated for any levels. I want them to feel engaged in the team, so I try to rotate the one who fills that position every week. It does not matter whatever role they have.	K1, O1, O3
27	The team leaders who are at these meetings, do they share the information with the rest of the teams?	

28	Of course.	K3
29	So, you think it is important that everyone gets all the information?	
30	Yes definitely, that is why I have that specific slot.	K3
31	Are you working with any delivery pipeline?	
32	We have several deliveries within each week, that we release to the production.	
33	The feedback that you get from this pipeline, how is information that processed?	
34	We have our customer service portal, and we usually have some complaints or defects or whatever through those portals.	K2
35	How does the work and communication look like between the development and the operations?	
36	Basically, DevOps consists of those people who are specifically placed in DevOps, in the permanent DevOps teams. But I also place other team members like from the development and I rotate them every two weeks, just to have this temporary position in DevOps. So, I rotate every team from the ERP and HR to act as DevOps every two weeks.	K1, O3
37	So, it is not always the same people who are in the DevOps team?	
38	Yes	O1, O3
39	Has the developers any direct contact with the customer or end user?	
40	Not the developers, but the product team has.	K2
41	Are they sharing the information to the developers what changes need to be done etc?	
42	Yes	K1
43	How involved are the operations team in the development?	
44	Well, they basically do not have any say on the product roadmap, but they have inputs that can contribute to the product backlog. But it has to go through the person responsible for that first. So, they do not actually say anything that actually affects the product backlog, they give inputs.	K2
45	What kind of inputs could that be?	

46	It is related to the product performance. So, when they develop something, you actually get the result later after it is launched in production. So, for example, how does the code perform, does it perform well? Or whether it is refactored well. So, it takes less time to load pages and basically how the product performs. And when it is not performing that well, they give inputs to the development teams to try to improve it.	K2, A2, O1
47	Are people in the teams encouraged to be creative and work with problem solving?	
48	Yes, definitely. We try to involve everyone, so everyone has a say in everything. Like in every meeting, everyone is allowed to voice their concerns and inputs.	A1, O1
49	When someone comes up with a new idea or discovers a problem somewhere, can they solve it on their own or are they expected to get any type of confirmation first from a leader or something?	
50	Each team member is assigned to one Scrum team. Every decision is based on whatever that Scrum team achieve. If that person has some kind of input, he can voice it out during the daily meeting and it will be discussed and decided by the whole team.	O1, A1
51	If some errors occur, who is responsible for solving it?	
52	It depends, if it is a production defect it is of course DevOps team responsibility. But if it is like in the development server, then it is the development team.	A2, O3
53	Those learnings, either who is responsible, are those shared with the rest of the teams?	
54	Yes, that is why I rotate them every two weeks. I want the knowledge to be shared, that is the idea.	K3
55	Do you have a lot of automated processes?	
56	Yes, we have QA automation implemented.	
57	How is that working?	
58	The DevOps manage it. Most of it is created by the development team, based on whatever enhancements they make. And then it is handed over to DevOps and they manage everything from there, because it helps them to anticipate defects early before it is being launched.	A2
59	We spoke about being creative, innovative and working with problems, how do you advocate new suggestions on improvements? Such as new tools to use etc.	
60	I give the autonomy to the teams, so they can actually implement whatever they would like. But of course, there are some kind of hierarchy, so we have like technical lead. So,	O1, A1, O2

	it has to be approved by them. So, everything has to reach some kind of a consensus, before it is actually being implemented.	
61	How receptive are you for change?	
62	Of course, in technology everything changes fast. So, you have to be aware of every changes that might become the dominant process in the market. So yes, we are open to any kind of change.	A2
63	What you say makes you a DevOps organization?	
64	Every company has a DevOps because it acts as an interface to your customers. So, it acts like an interface between your product and your customers. But the different thing is between customer service is that DevOps actually is rotated between the development teams, so they have like feedback cycle.	K2
65	Have any situations arisen where working with DevOps have been a problem?	
66	Before I came up with this model, the two weeks rotation. I mean I learned it from my previous experience, they had a very rigid structure between development and operations. So, whatever is handled by the operations does not go back to the development team, so the line is cut and there is no improvement between those two. And the communication is not going very well, and everybody blame at each other.	K2, K3, K1
67	Is there something else you would like to add? What would you say is important when it comes to leadership in a DevOps context?	
68	Make sure the communication is happening in between those two departments.	K1

B6 – Transkriberingsprotokoll P4

Information	Färg efter huvudområde
Verksamhet: Byggbransch	(V) Vision = röd
Intervjuobjekt: P4	(K) Kommunikation = grön
Tid och plats: 25-04-2019 via personligt möte	(A) Innovativt och lösningsorienterat arbetssätt = blå
Medverkande: P4, Eric Magnusson och Julia Trulsson	(O) Individuell omtanke = gul

Rad:	Information:	Aspekt:
1	Jag är anställd som konsult, IT-konsult-arkitekt på X. Och är just nu placerad hos Y på deras IT-avdelning där vi bygger Y.com.	
2	Kan du berätta lite mer om din roll och dina arbetsuppgifter och så?	
3	Ja men vi kan väl starta så. 2006 bestämde jag mig för att börja jobba och då gjorde jag den klassiska amerikanska, började på den lägsta nivån och jobbade mig uppåt. Lägsta nivån betyder då IT-support via Academic Work och dåvarande AU-system nuvarande Cybercom i Malmö mot Telenor.[...] Vi kommer över som UI-expert första året, bytte över då och då under X. Åtagande var X-lett och jag var där som sub-konsult, underkonsult. UI-expert och gick över till X fick mer lön. Blev snabbt Lead-utvecklare på frontend-biten så Y.coms webb. Höll på med det i 2–3 år, det projektet lades ner, ett nytt projekt skapades för att, vad jag kan säga där är att det projekt jag gick till 2013 var ett monolitiskt system, att det var ett system och det var ju ett state of the art på den tiden men 2–3 år därefter skiftade marknaden över till ett micro-service orienterat system och det var föråldrat. Så nu gör vi en transaktion från monolitiskt till micro-service och cloud arkitektur. Det är det nya, nuvarande där jag är i form av Software Engineer och jag kommer till varför jag inte säger utvecklare senare då det handlar om arbetsmetodik. Jag har även roller som cutover-architect och cutover-manager vilket betyder när vi lanserar marknader så hanterar jag den sekventiella lanseringen. Vilka system går först osv.	
4	Jag antar att du har arbetat i team eller liknande, hur ser ditt team ut? Vad finns det för roller?	
5	Vad är intressant här? Mitt projekt-team eller mitt team på X? För det är olika.	
6	Kan du berätta skillnaden om dom så vi förstår lite mer?	
7	Så, hos min nuvarande arbetsgivare som är X så jobbar jag mer som, jag är arkitekt på X som min profession och där är jag i form av arkitekt och jag profilerat mig inom cloud och agil, cloud och DevOps/agil så jag eftersträvar att vara arkitekt och agil lead/agil coach eller så arkitekt inom DevOps och coach inom agila metoder, agil metodik och arbetsmetodik och nu närmast så är det SAFE 4.6, så Scale Agile som är Spotifys variant på uppskalad agil metodik inom större organisationer. Det är väl liksom vad jag gör på X och om ni tar mitt projekt-team så där bygger vi betalningslösningen för Y.com. Vi är ett	K3

	team av, jag tror vi är uppe i 5–6 utvecklare, 3 testare, 2 BA, 2 UX, 1 product owner, 1 vice product owner och en Scrum master. Så arbetsmetodiken är Scrum och det är team-arbetsmetodiken och de övriga är då Spotify-engineer model eller numera kallad Scrum at Scale.	
8	Men vad är din mer specifika roll i specifika projektet då?	
9	Så min roll är utvecklare, cutover architect och cutover manager på Y. Scrum at Scale heter det. Nu kan jag inte svära om det är Spotify som har namngett det eller om det är X alternativ till Spotifys modell.	
10	Hur arbetar ni med mål inom projekt, när ni sätter upp de och så?	
11	Ja vi är ju uppdelade i Scrum-team, där varje Scrum-team har en Scrum-master, en PO som sätter scopet, back-logen, kommer ifrån POn då det är han som prioriterar den. Nu så råder det lite, det sker en transaktion från SAFe till Scrum at Scale, eller Spotifys modell. Tidigare så var det ju lett mer överordnade, en Kanban där vi har epic-stories och det är så vi tar tåget fram om man ska prata Scrum/SAFe. Känner ni till SAFe?	V1
12	Ja, vi har kikat lite på det som ramverk.	
13	Ja då har ni sett release-tåget som tuffar på och varje iteration ska leverera någonting. Det var det Y gjorde förr. Nu har det ändrats och jag kan det teoretiskt, Spotify alltså, men det är fortfarande nytt för mig [...] men inte inom Scrum at Scale. Men det är ju fortfarande på epic-nivån på feature-nivån och back-logen till att features ägs nu av PO och han samverkar med andra POs, han samverkar med mig som cutover arkitekt så vi har ju block över kalenderåret där vi antingen ska ut en ny marknad eller en ny funktion i en marknad, en ny betalningstjänst eller vad det nu kan vara. Då ska det anordnas i alla team genom hela organisationen även liksom här där vi sitter nu med varuhuset att de kan hantera och leverera.	K1, V1
14	Så du nämnde något att ni använder Kanban?	
15	Nej, när vi körde SAFe så var det Kanban ovanpå. Om du tar fram SAFe blueprint så kommer ni se massa Kanban-boards ovanför. Teamen själva använder antingen Kanban eller Scrum eller Lean, vilken metod de vill, det är upp till dom. SAFe satte ju Scrum, nu har vi mer öppet att man kan köra Kanban, man kan köra Scrum. Vi kör Scrum just nu.	O1
16	Men vem ser till så att målen efterlevs, att de följs upp?	
17	Teamet. Vi ju en prioriterad lista från PO vid varje sprint så commitar vi till en sprint-backlog och den efterföljs av teamet. Scrum-master faciliterar Scrum, ceremonierna och vi har ju demo efter varje sprint och då vet vi om vi har uppnått allting eller om vi har något slippage. Vi har ju en break down och allting. Det är så vi följer upp.	V1, O3
18	Har ni någon form av mätetal för utveckling och driftsättning?	
19	Mätetalen är ju vår brandanchart och vi har ju story-points där vi har estimerat allting och vi följer det efter sprintens längd kontra vår kapacitet. D.v.s. hur många utvecklare, hur	V2

	många testare osv. När vi väl har slängt upp något i produktion så är det ju KPI som conversion-rates osv. T.ex. då gick vi upp i ordrar, eller ner.	
20	Är detta något hela teamet tar del av, eller är det specifikt för utvecklare?	
21	Vi har olika radiatorer som visar olika KPI:er. Conversion-rates kommer ju direkt till PO för han har ju andra stakeholders som tittar över Y.com som en helhet, även marknaderna. Så varje marknad äger sin egen webb men vi har ju en global upprullningsenhet med Y-IT som; här är lösningen, marknaden tar över den, dom följer upp, dom kommer över till oss när det är incidenter osv.	V2, K2
22	Hur ser kommunikationen ut mellan teamen/avdelningarna? Är det möten och är alla delaktiga då?	
23	Nu får du gå tillbaka till de olika arbetsmetoderna där vi hade SAFe, så hade vi ju en PI-planning, project incremental planning och då är det att alla team, alla vertikaler eller vad vi ska kalla dom kommer ihop en till två dagars planeringspunkt där dom nya feature presenteras, eller de nya PI incremental planen presenteras, vilka features vi ser ska gå in i den här incremental-planen och vi swormar ju att sätta ihop en sådan plan och se om vi kan committa till den eller inte. Nu i och med att vi har så, vad jag tror personligen, var att det där sågs som en väldigt tung grupp och toppstyrd plan. Teamen, eller vertikala teamen, så är det ju Scrum-teamen och när jag pratar om vertikala så är det affärsområdena som shopping, produktgalleri osv. Då är det är även team i back-end osv. D.v.s. hela kedjan för att uppnå t.ex. att addera en produkt till carten, shoppingbagen. Och switchen som vi har gjort är att dela upp oss i affärsenheter och jag jobbar i shopping [...].	K1, K3, A2
24	Och det som tas upp på planeringsmöten, är det sånt som delas med ut vidare?	
25	Ja vi har ju utrullningsplaner där jag i form av cutover arkitekt och cutover manager är, så jag tittar då över hela organisationen eller hela arkitekturen som en unik [...] att så här skulle vi kanske rulla ut klarna som en betalningsmöjlighet och då ser jag till att sekvensera planen så att vi har backend på plats, vi har APIer på plats, vi har content på plats och när saker ska göras i planen så att vi uppnår måltiden om vecka 25 går vi live men det här och det hur vi skulle komma över det. Vi har även marknadscoordinatorer om det är Klarna i Danmark, om vi tar det som ett exempel, så pratar dom med Danmark att ja vi ska ju uppnå vecka 25 för Klarna, är ni beredda? Vad behöver ni i form av verifikation, test och vad gör er nöjda okej, tummen upp vi går ut med det här.	O2, K1
26	Så det är väldigt viktigt att alla är med mot samma mål?	
27	Så vi koordinerar på olika nivåer, vi har ju en en country-rollout plan där vi rullar ut e-commerce på olika marknader. Alla Y-marknader har inte e-handel utan har då inspirationssidor vilket betyder samma sak i butiken här [...]	V1
28	Använder ni någon pipeline när ni levererar nya features eller ändringar?	
29	Jaja, så självklart så har vi olika maturity level i de olika teamen. Det är många team och vi har olika maturities i de olika. Vissa är full DevOps, d.v.s. helt automatiserat, andra har mer handpålägg för att gå ut och det är väl där koordinationen behövs mellan teamen och det är det jag gör på halv eller fulltid ibland, att se till att men här har vi ett team som behöver lite mer handpåläggning och här har vi ett team där jag behöver bara veta att det är på gång, att det är klart. Så lösningen som det ser ut just nu är multi-cloud baserad, vi har	O1, O2

	saker i Amazon, vi har saker i Google Cloud, vi har saker on-prem vilket betyder kundens datacenter [...]	
30	Och feedback som ni får, då det triggas igång enhetstester, hur arbetar ni med det?	
31	Så i mitt team så kommer feedback direkt till slack som ett meddelande att något gick fel och det tar teamet hand om.	K2
32	Är det vem som helst eller den som känner sig mest manad för att ta det?	
33	Japp, har vi ett moget team så är det vem som helst. Man är som en scout-maner att ta hand om det. Sen beror det också på om hur närmre vi kommer till en release desto mer av vår kapacitet har vi lagt undan för support osv. Så då går vi helt enkelt in i ett Kanban-läge där vi bara har en inbox och en output och hanterar därefter vilket vi är i just nu.	O1, O3, A2
34	Hur sker mer feedback mellan kund och utvecklare? Har dom nära kontakt?	
35	Dom har väldigt nära kontakt men det är också fasat som jag sa, vi når ju ut till marknaderna och kollar beroende på det är som kommer ut. Säg att en marknad, om vi tar den stora eller den lilla, säg att en marknad ska få e-handel så är det mycket mer tight samarbete mellan oss för att dom går ju från ingen e-handel till e-handel. Är det en marknad som redan har e-handel så är det mindre releaser alltså look-and-feel har ändrat en funktion som dom vill ha som liksom adderar mervärde eller värde överhuvudtaget och då är det från ingen alls, där vi har trust o.s.v., till dom testas och meddelar att det funkar bra. Den nivån eller att vi får flyger dom över hit och har sessioner osv. Så det beror på magnituden av vad vi gör.	K2, O1
36	Jag är lite nyfiken, om man bara ser till utvecklare och driftpersonal, hur ser samarbetet ut där emellan? Är de involverade i varandras arbete?	
37	Ja alltså, vi kan säga klassiskt så har det varit ganska silo som att vi slänger över en het potatis över staketet men absolut så har det varit ett skift till agilt och DevOps där teamen tar mer ägandeskap och ansvar och som vi sa där är det olika mogenheter inom teamen. Vi har ju personal på Ops-sidan men numera när vi tänker så är det operativa att se till att IP-adresserna och nätverkstekniken är på plats men i och med att vi har Amazon och Google så är det ganska hands-off och undanskymt då det är köpt som en tjänst. När det väl är uppsatt är det mer att konfigurera om att ja ni hade ingen e-handel och nu har ni en e-handel och det är väl den operativa delen som är kvar då. Allt annat är ju pipeline:at och klart.	K1, O3
38	Kan driftpersonal ge inputs till utvecklarna också?	
39	Ja absolut, men det är mer satelliter och supportpersonal. Tidigare så var det utstakat i silona att här har vi en produkt över till Ops:en som skickar ut den till produktion, marknaden testas, vi går tillbaka från silo till silo. Nu är det mer hela den här kedjan har strypts och nu har vi ett team som tar hand om allting i sann DevOps.	K1, O3
40	Och när det uppstår problem, förespråkar du att man ska vara kreativ och arbeta med att försöka hitta problem och försöka lösa dom?	

41	Ja, men alltså det är väl nu vi går till "what is true agile" och det är "when shit hits the pan" och nu löser vi det för att addera värde och fungerar ingenting så adderar vi inget värde och det är "all hands on deck" att fixa det så vi levererar värde och värde är slutligen kronor och ören och det är det som är det viktigaste. Nu så sitter vi i en position där vi har ett par problem med Spanien-marknaden och vi har ett helt team som är committade att jobba över helgen för att lösa dom mest kritiska incidenterna som vi har. Så det är teamet har ansvar och teamet levererar och också får cred för det.	A2, O1, O3
42	Vad är det för typ av cred?	
43	Om vi kollar på det klassiska då vi inte hade så mycket agilt och DevOps när vi hade vattenfall så var det att vi hade hjältar. Vi hade Eric där vi gick till som hade koll på den biten och kunde lösa det. Nu vill vi inte ha några hjältar, utan vi ska jobba agilt och teamet tar ägandeskap, teamet levererar och teamet som beslutar vad som behöver göras, när och hur för att nå de värde vi eftersträvar.	O3, K1
44	Jag tänkte även, vi snackade om problem uppstår, får dom lösa det på egen hand eller måste ni gå till en PO och rådfråga först?	
45	Scopet ägs av PO. Committed scope ägs ju av teamet och det är en så som vi internt kallar, majorbusiness critical incident, vilket innebär att vi helt enkelt förlorar pengar. Det är den högsta incidenten vi kan ha, vi kan inte ta betalningar på nätet, det är det högsta jag kan tänka mig. Vi ligger helt enkelt nere. Då är det självklart teamet som det berör gör ju komponenter och varje team äger sitt affärsområde, mitt är ju shopping - cart, checkout och payment. Är det något som inte fungerar där så får vi en incident från marknaden det berör eller är det globalt så får vi en hel incident från alla marknader. Och det kommer ju gå till teamet och PO och då får vi planera om och få in det i sprinten och det är ju teamets ansvar och PO ansvar att omprioritera och få in det så att vi har den arbetstid vi behöver för att lösa det. Men vi har i varje sprint en buffert för sådana här saker och den är 20% av kapaciteten och kan den inte hanteras inom det får vi gå över till vår PO som äger vår kapacitet och meddela att vi får prioritera om för att det här ska komma in.	O3, K2, O2
46	Du nämnde också att Amazon var det automatiserat och Google delvis. Tror du att det finns nackdelar med att ha färre manuella kontroller?	
47	Det är en svår fråga. För vad det beror på är ju att har du ett automatiserat för automationens skull eller har för att du inte ska slösa tid på manuella saker, repetitiva saker, och är det de senare, Ja, jag ser inga problem med att automatisera så mycket som möjligt men är det det första så har du förlorat hela poängen att gå DevOps. Hela idén med DevOps att köra mer Dev än Ops, eller låt mig omformulera att Ops-delen inte ska vara i silo utan ska vara ägt av teamet och ger du en utvecklare i uppgift att göra en sak två gånger så är det automatiserat på den tredje gången, eller förhoppningsvis efter den första och då kan man leverera mer värde då det är ingen administration, ingen byråkrati, det är hanterat och automatiserat samt förmodligen också mätt så att vi vet att det får plats.	V1, K1, A1, V2
48	Nya förslag på förbättringar om någon i teamet skulle nämna det, är det något som ni vill att dom ska ta upp eller har ni det strukturerat?	
49	Nej, stacken (utvecklingsstacken) ägs av teamet, det gör även produkten sammanägd med PO. Nej, jag skulle säga att det är teamet som äger sin tekniska stack och teamet vill byta till något annat som ger mervärde så ja. Våra team behöver inte koda i samma språk mellan teamen p.g.a. ett verktyg. De flesta kodar i React och i Javascript men vi har team som kodar helt i Go eller Java. Så det är upp till teamen så länge arkitekturen fungerar som helhet så är det okej. Vi integrerar på API-nivåer.	A1, O3

50	Vad är fördelen med det?	
51	Det är ju en fördel och nackdel. Fördelen är att varje utvecklare, speciellt yngre, kommer vilja köra det nya häftiga och på ett sätt är det bra då saker förändras och blir bättre med tiden och man är ganska flexibel och adaptiv och kan vända med vindarna som råder som tex som när vi gick från monolitiskt till micro-service inom loppet av 3 månader. Det var det nya över hela världen. Om man tittar på Twitter så har dom gjort om sin back-end 3–4 gånger och det är för att uppnå värde snabbare. Nackdelen är att är det toppstyrt, att nu ska vi gå från Java till JavaScript så får man se över regionen där man finns. Kan man hitta rätt antal JavaScript-utvecklare med rätt kompetens? Så om vi tittar på Skåne så har de stora aktörerna varit Axis, Sony, Tetrapak och ett par andra som man kommer slåss om samma kompetenser.	A1
52	Kan det vara individuella fördelar med att programmera på det språket man vill? Att dom uppskattar att få skriva i det språk de är mest bekväma med?	
53	Ja man bygger ju en kultur och man ska fortsätta bygga en sund kultur och det kan man inte göra om man tvingar någon att göra något de inte vill. Men det är ju också en avvägning att varje utvecklare som kommer in har en nyanserad syn om vad som är rätt och fel. Det handlar även om en kompetens och erfarenhetsnivå. Det kan vara en utmaning men ingenting man ska se ner på men man måste se till att helheten fungerar.	O1, A1
54	Du nämnde att man är adaptiv, hur mottagliga är ni för förändringar?	
55	Om vi tänker på det som slagit hela Europa, så är det ju GDPR och slog de flesta aktörer men det var ju bara till att gilla läget och fixa det. Jag antar att det beror på vad det är för organisation man är i och för kultur och ledarskapstyp och styre som råder. Är de mottagliga och i vilken grad, vilken mognadsgrad? Är de rädda för en förändring? Om jag inte utgår från vart jag är just nu så ser jag att andra ville gå agilt men lyckades inte med det för att de inte köptes av organisationen eller arbetskraften som fanns tillgänglig. Så det är väl också att vara beredd på den. En sak som jag brukar i form av DevOps-evangelist så är det mottagligheten, det är inte bara en radda automatiska processer utan det är också en kulturbit. Hur beredd är organisationen att ta den, finns det buy-in från ledningen hela vägen upp? Köps den hela vägen ner till Agda som sitter i supporten? Ja, fine. Men utan det så har det bara en pipeline som sitter där. Jag har sett kunder som köpt enorma automationssystem för att man skulle ha det för att köra DevOps.	V1, A2, K1
56	Vad skulle säga gör er till en DevOps-organisation?	
57	Jag anser oss vara ledande inom DevOps och agila metoder. Nu vet jag inte om vi är topp 5, det brukar skifta om man kollar Gartner och så. Min del är ju en konsultfirma, vi levererar tjänster och måste leverera vad kunden efterfrågar. Där jag är just nu i vårt projekt så är det uttalat på CEO-nivå att vi ska gå agilt och DevOps att vi ska vara flexibla och snabba. Vi ska ha allt det där. Som jag sa innan så har vi testat SAFE och nu testar vi Spotifys metod och jag ser kulturen hamnar på plats och det jag kan säga är att skillnaden mellan SAFE och Spotify är att det passar bättre på företaget där jag är och det var ingen större skillnad i kulturen mot företagets värderingar till att gå över till Scrum at Scale och Spotify. Tittar man på SAFE så var det flera lager, det fanns en government och mer toppstyrt och för tungt. Ett av företagets grundord är tillsammans och Scrum at Scale funkar bättre med chapters och tribes osv.	V1
58	Om man ser mer till DevOps ur ett ledarskapsperspektiv, vad är det viktigaste då?	
59	Så på företaget, tanken är att efter detta projekt ska jag gå med i ett transformationsprojekt, där vi hjälper företag att gå över till agila eller DevOps-praxis och	V1

	<p>transformerar. Vi har en guide, en setup som vi följer och visar kund. Det första vi gör att vi gör en Agil och DevOps-assessment där vi intervjuar och kollar igenom vart ni är just nu och vart de vill nå. Jag tror de båda är viktiga och det jag försöker förändra inom det är vart tror ni att vi är och vart vi ser att ni är? En opartisk men också perceived perception, att det här säger att ni är men vad vi har märkt att ni är ungefär här och nivåerna betyder ingenting utan den faktiska poängen är att ni har en pipeline men ni har en 3-veckors connect mellan delivery och release. varför har ni den? Det hade vi tidigare, då får 3 år sedan tog en release för oss mellan 3-6 månader och nu har vi en när vi vill och det är det jag tror är syftet med DevOps. Först och främst är det kulturen för att uppnå full DevOps så måste man börja med kulturen så det är den som kommer ta längst tid. Att sätta upp en pipeline tar mig 2 timmar men vill ingen använda den så är den förlorad. Det är inte att automatisera releaser varje vecka utan det är release on-demand d.v.s. behöver vi release så gör vi det. Det finns ingen mening att release varannan timme om det inte levererar värde, utan väntar hellre då. Då väntar vi på kvalité och värde.</p>	
60	Sista frågan, om du har varit med att DevOps har varit ett problem?	
61	<p>Ja, flera som jag sa innan. Finns inte kulturen på plats så hjälper det inte, du kan köpa världens coolaste automationsverktyg men finns det ingen commitment både från golvet eller toppen så spelar det ingen roll. Det måste också vara linat över organisationen och inte bara ett ben av organisationen, mognad. Sen är det mer vad är det för bransch, varför, man kan göra DevOps utan agil men man kan inte göra agil utan DevOps och anledningen till det är att DevOps, [...] Man måste se över sin organisation vart man vill, att mäta sig med Spotify samtidigt som man har 500 000 medarbetare kan man göra och lyckas men man kanske ska titta på hur företag som hade en snarlik organisation gjorde för att lyckas och det är väl där jag tror att man får göra sin hemläxa och se vad som passar en. Om man drar en parallell, alla kan köpa systemkamera men ingen blir fotograf över en natt.</p>	V1
62	Jag tror att vi har fått med allt vi vill, är det något mer som du tror vi har missat?	
63	<p>Jag tror inte poängen är att man ska vara vad man vill. Poängen är att man ska ha den kulturen, friheten att självorganisera. Är du fast vid ett element och någon säger åt dig att vara agil så är det ganska svårt. Men man måste ändå se över att det kommer fungera. Varje kuggjul kan se unikt ut, men systemet i sig måste snurra runt. Så det handlar inte om att jag ska koda JavaScript för det är det jag gillar, utan det är mer att dom här verktygen är vad vi behöver för att vara så performance som vi kan vara och leverera så mycket värde så snabbt vi kan. Det ska man då framhäva och möjliggöra. Det ska inte vara en lekstuga, men samtidigt inte toppstyrt heller. Det är en avvägning och med tiden så kommer allt evolvera och man kommer hitta sig själv och se vad som passar som passar dom bäst. Det är väl det man ska promota. När det kommer till DevOps och agilt så är kulturen viktigast. Utan kulturen så kan du inte uppnå DevOps. Det måste köpas av ledningsgruppen och aktivt säljas in i DNAt av företaget. Det hjälper inte att någon skriker från ett torn att vi ska gå DevOps eller agilt, men alla andra tittar bort. Jag tror ägande jobbar för det, från CEO och neråt. Det kan lyckas som en grassroot, att utvecklingen skulle jobba själva så kan dom lyckas men där är en kostnad som måste till för att det ska uppnås. Det måste vara förstått att det kommer ta tid och finns det ingen i business som förstår att det kommer ge mervärde så kommer det trampas ner. Det är det som är den kulturella förändringen, att man äger det. Det är väl det jag kan sakna. Du kan inte komma med dubbla budskap, att det ena är ditt jobb och det andra är extrajobb. Så om du lyckas få fram kulturen att gör dig själv och bli bättre, och aktivt supportera det beteendet så kommer kulturen komma in att man itererar själv, evolverar och kommer fram till de mål man vill komma åt.</p>	V1, O3, O2

B7 – Transkriberingsprotokoll P5

Information	Färg efter huvudområde
Verksamhet: Byggbransch	(V) Vision = röd
Intervjuobjekt: P5	(K) Kommunikation = grön
Tid och plats: 07-05-2019 via personligt möte	(A) Innovativt och lösningsorienterat arbetssätt = blå
Medverkande: P5, Eric Magnusson och Julia Trulsson	(O) Individuell omtanke = gul

Rad:	Information:	Aspekt:
1	Vill du berätta lite om din roll på företaget och tidigare bakgrund?	
2	Absolut. Min nuvarande roll är Chapter Lead för DevOps engineering i den nya organisationen. Jag har varit här i på företaget sedan 2001, men även jobbat ännu längre inom IT så jag har varit med i gamet. Jag har jobbat med allt från PC tekniker, till databasadministratör, serveradministratör, jobbat mycket med Linux då jag varit Linux ansvarig här på företaget, även jobbat mycket med virtualisering, inom operations men också infrastruktur-automation som utvecklare och byggt portaler för det. Och så har jag jobbat som arkitekt och inom innovationsområden så har jag jobbat mycket med test-and-learn-hypotesen som vi kallar det. Man jobbar med Lean startupmetodologin, design thinking, prototyper, bygga MVP för att få snabbt resultat och feedback för att se om man går i rätt riktning eller inte. Så rätt bred bakgrund.	
3	Hur länge har du arbetat som Chapter Lead?	
4	Vi påbörjade hela den här stora omorganisationen förra året, så det är en del av vår transformation som vi gör på X. Vi jobbar väldigt mycket med varumärket och vi är den största franchisetagaren och strategisk partner till InterX. X är ju ett varumärke så x-gruppen heter vi. Vi promotar inte det så mycket, men det är ju för X vi jobbar. Vi finns i över 30 marknader, över 370 varuhus och citycenters. En stor apparat. Vi är i en jättetransformation i alla avdelningar för att slimma vår organisation lite mer, platta till den lite granna. Men också röra oss mer mot det digitala. Vi måste mäta kunderna, kundernas förändrade beteenden. Ni har hört talas om retail-döden i USA kanske? Den är ju rätt tydlig där. Så vi måste muskla upp lite och möta det på ett bra sätt. Och då handlar det om att vara convenient, möta kunden på ett bra sätt. Det ska vara enkelt att checka ut, vi ska erbjuda en inspiration till människan och hemmet, finnas på många touch points, inte bara på desktop utan överallt där vi kan. Men också synas, att vi inte bara är en traditionell retailer, utan även det digitala. Så vi gör en push mycket i den riktningen där. Så hela den rörelsen sattes igång förra året för gamla IT som nu är Digital, så vi rebrandade oss. Organisationsstrukturen sattes bara nu i februari i den nya rollen. Men inom DevOps-kontext satte jag igång ett arbete förra året, för jag visste att det är dithän vi måste röra oss för att vi ska få fart och effekt. Så jag gled in i rollen i februari och nu sitter jag här nu sedan dess i en rekryteringsprocess där jag har intervjuat massvis med människor. Vi ska bygga upp mitt Chapter då, och jag har satt en magisk siffra på 72 människor, så får vi se om det håller eller inte.	

5	Spännande. Vill du berätta lite mer om vad du har för arbetsuppgifter, mer än rekryteringen du sitter med just nu?	
6	Det är att ta fram <i>hur</i> vi vill jobba. Och det är ju satt på en bredare agenda att vi vill jobba produktorienterat, vi vill vara en produkt-centrisk organisation. Det är ett stort skifte att vi går från projektledad organisation till produkt, och mer långlivade produkter. Och de ska vara komposerade med krossfunktionell kompetens. Så vi har någon utvecklare, någon UX, och vad som behövs. Det är som sagt på övergripande nivå, men inte detalj <i>hur</i> . Så då har jag, tillsammans med andra, tagit fram en modell hur vi kan jobba med DevOps. Och DevOps för oss är allt från ideation-bitarna, vet inte hur jag ska översätta det på svenska men "idéverkstaden", till agil mjukvaruutveckling där vi till exempel använder Scrum eller Kanban-metoder för att styra, ha koll, och korta iterationer för utveckling. Men också drift, maintenance, och support. Det är hela DevOps-kedjan, end-to-end. Så där sitter jag mycket och jobbar med "hur kan vi göra det på bästa sätt, och hur kan vi stödja det här flödet?", med hjälp av verktyg, arbetssätt och metodik. En stor del av det är att ta fram metrics, där vi kommer jobba med att mäta en sorts mognadsnivå på teamen: "är ni gröna, orange, eller röda?". Sen är det upp till produktägaren i den här produkt-centriska organisationen, att sätta KPI:er på "jag vill att vi ska vara gröna". Men det är upp till dem. Mitt jobb är att tydliggöra, baserat på de här metrics, hur många releaser de gör i veckan och vad det håller för kvalitet. Då kan jag se var de ligger.	V1, O3, K1, V2, O1, O2
7	Så beroende på vad teamen har för mognadsgrad sätter man egna KPI:er?	
8	Ja, de sätter egna KPI:er som de måste uppfylla. Det är ju svårt för mig utifrån, och det är den gamla vägen "att jag sätter KPI:er på dig". Och vi rör oss mer mot OKR:er, Objectives and Key Results. Så vi sätter ett mål, och sen hur vi ska mäta det här och fylla upp det målet. Så det är ett mycket bättre sätt att jobba än bara med rena KPI:er.	V2, O1, V1
9	Du pratar om att ni bedriver både måluppsättning och mätning, är det gemensamma eller individuella mål?	
10	Det är på olika nivåer som vi sätter dem. På högsta nivå av Digital så har vi några OKR:er, sen trillar det ned. Men det är ju en cascading effect, så de är sammankopplade till huvudmålen men sen är det på individuell basis för ett visst produktområde eller produkt. Mitt chapter kommer alltid fram. Nu har jag inte gjort det ännu, men jag ska påbörja det arbetet och sätta några OKR:er. En klassisk är ju att "enligt de här fyra huvudmålen eller metrics så ska vi vara gröna, vi ska jobba för att få så många gröna som möjligt". Och då kan man mäta det med hjälp av de här OKR:erna.	V1, V2
11	Är det ni som ser till att de här målen efterlevs och följs upp?	
12	Det är upp till respektive produktägare eller Product Manager att jobba med teamen. Om man har ett team för "check out" till exempel, då kan jag ju önska att alla är gröna men då är krav på min rekryteringsfas och sånt. Att de jobbar för att hjälpa för att det ska skala DevOps ways of working på ett bra sätt. En bra mognadsnivå tyder på att man lyckats. Så jag kan ju ha som mål att jag vill att så många som möjligt ska vara gröna, men sen är det ju respektive PO som sätter sina KPI:er. Till exempel "det ska vara 80%, det är rimligt för oss".	V1, O2
13	Så man utgår utifrån vad som är bäst för sitt eget team?	
14	Precis. Så det är två separata med samma målbild, men som ser det på olika sätt.	V1

15	Skulle du säga att PO har ett övergripande ansvar i teamen?	
16	Oh ja. Det är det som är det intressanta i den här produktcentriska organisationen. PO rollen är väldigt bred, och den har väldigt stort ansvar. Det vi brukar säga är "fully accountable", det är ju inte riktigt ansvar men accountable. Och teamet i sig är ju ansvariga då på vad de tar fram. Men den som är ytterst ansvarig, eller accountable, är PO. Vi går över mer till "minichefsroller". Det är något som jag tror McKinzy översatte Project Management-rollen till mini-CIO. [...]	O2, O3
17	Hur ser kommunikationen ut mellan och inom teamen? T.ex. vilka är delaktiga på möten och liknande?	
18	Det är en väldigt intressant fråga. Som ni vet i svensk kultur, eller svensk företagskultur, så har vi en consensus-grej som är väldigt stark. Där alla ska vara med på allting. Det sticker isär lite vad vi är ute efter, att vi vill delegera ansvar, vi vill vara snabbfotade, vi vill känna oss empowered. Så vi försöka bryta det här. Vi vill samarbeta, men vi vill inte mer egentligen utan vi vill samarbeta mindre, men mer effektivt. Så vi vill hålla det inom ett produktteam så mycket som möjligt, eller inom ett område. Kommunikationen däremellan i teamen kan de lösa lite hur de vill. Jag skulle inte rekommendera dem att göra mailkommunikation bara. Vi har en tanke att vi vill använda en IRC (Internet Relay Chat) -baserad slack för kommunikation. Vi har inte fått in det ännu men vi är just i görningen att sätta ett kontrakt med dem, så vi kan kommunicera mer effektivt och kollaborera kring saker. Vi har Confluence som ett verktyg från Atlassian, där vi dokumenterar och vi kan också kommentera och dela dokument. Vi använder Office365 för onlinedokument, som gör att många kan samarbeta. Så vi försöker bli mer effektiva i vår workspace productivity agenda.	O3, A1, K1
19	Är det viktigt för er att alla får ta del av allting som sägs och görs?	
20	Nja. Eller ja, iallafall att man är transparent i vad man gör. Men inte att man behöver gå runt och fråga alla. Mer transparensen, den är viktig. Vi får det här med accountability genom en transparens för annars blir det lite "vad gör de där i sitt hörn? Jag har ingen aning". Men är man väldigt tydlig och öppen med det man gör så. Vi är ute efter en väldigt öppen kultur. Så Open-allting: open collaboration, open source, open communication, det är några av ledorden som vi jobbar med.	K3, K1
21	Skulle du säga att det är viktigt att alla i organisationen delar samma värderingar och vision?	
22	Superviktigt. Och när jag är ute och pratar om DevOps i industrin, det riktar sig väldigt mycket åt tooling till industrin speciellt från företag och konsultfirmor, det handlar bara om det. Men DevOps är ju så mycket mer än tooling. Jag brukar sammanfatta det att det är en kombination av cultural mindset, practices, and tools. Och det sammanfattas i en akronym: CALMS. När jag sitter i intervjuer med folk brukar jag alltid utgå från den. Om folk känner till den och om de vet någonting om den. Det sammanfattar DevOps för mig på ett bra sätt. Det är culture, automation, sen går den in på Lean-metodologin och det är superviktigt. Från Lean får man optimering av flöden, vi får snabb feedback, vi reducerar waste som vi streamline:ar. Vi har även measurements, vi mäter allting vi gör för annars får vi inte den här transparensen och att vi kan få snabb feedback. Sharing är ju superviktigt i den. Från automation har jag dragit ut en ingenjörroll i form av DevOps Engineers, som kommer jobba mycket med automation: automation av flöden, CI/CD, IaC, så infrastruktur-automation ligger mycket där, i molnet, virtualisering och hela de bitarna. Men DevOps är mycket mer än så, men man kan dra ut en ingenjörroll ur det. Så kulturen är viktigt, och kulturen är något som vi gemensamt jobbar fram. X har ju en väldigt stark värdegrund, vi har 8 grundvärden som vi jobbar efter. Så alla principer som vi tillämpar är ju byggt ovanpå den. Så starka sådana som passar väldigt bra in i detta är Simplicity som vi jobbar	V1, K2, V2, K3, A2, K1, A1, O2

	<p>mycket med, att vi måste vara smarta och ha enkla lösningar. Togetherness är även väldigt starkt för oss, att vi gör det gemensamt. I den här kulturen jobbar vi i produktteam och vi gör allting tillsammans, vi delar uppgifter, vi är öppna med kommunikation så alla känner till vad vi gör i teamet. Så togetherness är starkt. Vi har en annan värdering som vi kallar Different with a meaning, och där kommer hela det här explorativa tankesättet in. Test-and-learn, eller Test-and-try som vi kallar det också. Vi har Leading by example, som är viktigt för mig inom mitt chapter. Om jag sätter en DevOps Engineer i ett team så vill jag att de sätter agendan: "Det här är ett bra sätt att jobba på". De gör det genom att visa - show, not tell. Det är hela den grejen, vi visar och vi gör och förhoppningsvis gör vi det riktigt bra så att andra tar efter. Och det är Lead by example.</p>	
23	<p>Du pratade om att ni arbetar med att våga testa, kan du berätta lite mer om det? Förespråkas det att tänka kreativt och arbeta med problemlösning och sådant?</p>	
24	<p>Precis. Vi har ju gått lite från att inte ha någon struktur på innovation överhuvudtaget. Till att ha en grupp som jobbade med det, incubation kallade vi det. Men då blev det lite skevt för då var det bara en grupp som jobbade med det. Och då fick vi ingen spridning och det skalade inte på ett bra sätt. Så det vi försöker nu är att vi försöker bygga in innovation i allting vi gör, nu blir det ju mer kopplat till inkrementell innovation och så att man gör det inom sitt område, men ändå. Och sen då försöker vi jobba med Pareto-regeln 80/20. Så att 20% av din tid ska gå till utbildning av dig själv, din tid i chapter, kanske något event med någon chapter lead eller att man bjudit in någon, eller att man jobbar för Lean startup, det finns de som är superintresserade av det. Vi är väldigt många människor så det finns säkert flera andra som också är intresserade av det. Eller är jag superintresserad av Scrum, och då finns det grejer kopplat till det. Så hur man bäst jobbar med retrospective. "Man kan ju kanske skapa en SIG så kan vi få så att alla gör likadant". Så det är den jag vill bygga. Fast den är organisk så det är ingen manager som säger "nu ska du göra det", utan det måste komma underifrån. Så jag försöker hjälpa organisationen med det, att sprida underifrån, alltså bottom-up. Den mest effektiva är väl en kombination av både en klar riktning och styrning från top management, men också en bullrande gräsrotsrörelse. Men så är det ju överallt i samhället också såklart.</p>	<p>A1, K1, K3, O2</p>
25	<p>Förespråkas det att vemsomhelst ska lyfta sina idéer och göra sin röst hörd? T.ex. om man har förslag på nya verktyg eller liknande.</p>	
26	<p>Ja, absolut. Är det på den nivån när det gäller verktyg och sånt så måste vi ju ta in input från industrin. Som traditionellt IT företag har vi ju tidigare jobbat väldigt mycket med standarder, "nu är det den här som gäller" och sen hänger man kanske inte riktigt med i industrin, eller folk som kommer från skolan eller andra företag "nej men det här är ju 10 år gammalt". Vi måste vara lyhörda och lyssna på utvecklingarna och hur de vill jobba. De kanske inte vill sitta i IDA utan alla vill röra sig mot det eller lättare editors, då måste vi ta den inputen och bygga några slags guidelines hur vi bäst gör det. Så vi måste vara lyhörda för arbetsmassan och hur de vill jobba. Men sen innovation kan vara brett, kopplat till tooling och automation, men också kopplat till nya produkter eller befintliga produkter. Och då har vi en enhet här på x som vi kallar Development and Innovation, som ett forum som man kan gå och pitcha sitt case. Så man kan få pengar för sin exploration. Då är det ju lite större initiativ, men om vi vill göra något i varuhuset kopplat till in store navigation, då är det typ ett sådant när vi behöver lite extra resurser och lite extra pengar och då pitchar man det till det forumet. Så lite så 'Dragons' Den'-aktigt.</p>	<p>O1, O2, A1</p>

27	Så om någon kommer på en bra idé kan det ses lite som en belöning då att de får ta hand om det projektet själva och styra det?	
28	Javisst. Och vi vill ju lyfta alla medarbetare. Så det är inte bara "Kalle" som får alla de här roliga uppdragen.	A1
29	Om man ser till team-nivå, när problem uppstår i någon ny release eller något, vem bär ansvar för det? Är det hela teamet eller den enskilde individen?	
30	Det är teamet. Det är kollektivt. Men fullt ansvar, eller accountable, är ju PO. Det är PO som är ansvarig för vad teamet gör eller inte gör. Det är en grundförutsättning som vi har. Själva support och maintenance och allting är teamet, och där försöker vi jobba på att hitta bättre sätt. Support inom IT kommer ju mycket från ITIL och nu kommer en version 4, men annars händer det inte så mycket i den arenan. Där försöker vi också jobba fram något bättre sätt att arbeta. Vi vill att teamet ska göra så mycket som möjligt, för att det DevOps-mindset men de kan ju inte bli nedringda. Det kan ju inte vara det enda de gör, att jobba med support. Så därför måste vi offloada de på något bra sätt också. Men mycket kommer att lösas sig med automation. Men är det något fel så är det gemensamt att vi löser det, och antingen rullar vi tillbaka det eller så jobbar vi tillsammans för att rulla framåt så att vi fixar buggen. Men vi inte har inför någon formaliserad "no blame", men det ligger invävt i kulturen. Vi pekar inte finger eller säger "det var Kalles fel".	O3, A2, O1
31	Att arbeta gemensamt med problem som uppstår, är det för att kunna lära av varandra och sprida kunskap?	
32	Absolut. Och det är så det skalar. Det är så man lär sig om sin produkt. Både nya grejer, att skriva kod är jätteroligt, men att ta hand om någon annans kod - det är då man lär sig, hur det funkar i verkligheten.	A2, K3, O3
33	Bortsett från PO, vilka andra delar är viktiga och ingår i teamen?	
34	Om man utgår från Scrum som merparten kommer jobba med, så är det tre roller: PO, teamet och Scrum master. Så vi har väl valt en liknande variant där PO är given, sen kommer teamet komponeras ihop av resurser från olika Chapters. Vi har lite olika: DevOps Engineering Chapter, ett stort Software Engineering Chapter, Data Engineering, UX, Scrum, och lite flera olika. Så beroende på vilken produkt det är så kommer vi komponera teamet. Software Engineers kommer alltid vara med. Team består av 7 +/- 2 personer, för att hålla nere antalet. Ni kanske har hört talas om Two-Pizza size teams från Amazon. Men det här med +/- 2 är för att vi vill hålla det udda för att få jämlikhet. Så mitt Chapter är superintressant för vi bara har utvecklare så producerar du kod, men hur får du ut den här om vi ska bygga en skalbar miljö för att stödja den. Och det är här vi kommer in. Vi har ingen separat Ops, utan vi har valt att bygga upp mer plattformstänk. Så vår operations-del försöker vi tänka att enable:a oss, self service, och infrastruktur. [...] Men annars är ju DevOps Engineer som en hjälp till sin Software Engineer för att sätta ihop sin solution design på teamnivå. Sen har vi ju en supportfunktion i form av arkitekter som tittar mer på övergripande saker, som domän-arkitekter. Men DevOps Engineers jobbar mycket för att sätta upp text stacken så vi får ett bra flöde och kan deploya om och om igen.	K3, A1
35	Skulle du säga att roller blir mer flytande genom att arbeta med DevOps?	
36	Ja. Egentligen den här DevOps Engineer-rollen, den måste försvinna. Men den fungerar som en katalysator nu under några års tid, tills vi kan bygga upp den här mognaden hos alla. Att man har cloud nativeness, man tänker "så här bygger vi för	O3, A2

	molnet och på molnet, och så här kodar vi på ett smart sätt, och så här jobbar vi med modern agil utveckling för att frikoppla kod från data, från konfiguration". Tillsammans bidrar det till en DevOps-kultur.	
37	Så det blir allt viktigare att utvecklare har koll på driften?	
38	Ja, det måste dem. Jag har sagt till Chapter Lead som rekryterar nu för Software Engineer, att de måste också vara en junior DevOps Engineer för de måste ha en känsla för hur molnet fungerar och hur jag kan få min kod ut där, och hur den kan skala upp till miljoner användare. Någon sorts känsla måste finnas. Sen kan de ju lära sig, så det är en transition-fas. Misstolka mig rätt, men andrahandsprioritering måste vara text stack. Målet för Software Engineer är affärslogiken och features, men andrahands måste vara text stack. Så alla måste känna till det. Och då kan DevOps Engineeren funka som katalysator, för du kan inte bara säga till fem stycken Node.js-utvecklare att nu äger du hela stacken. Så kanske det kommer vara om ett tag beroende på mognadskurvan, men i början måste vi ha hjälp. Där ser jag vår value proposition.	V1
39	Du pratar mycket om mognadsgraden, har det varit viktigt för organisationen tidigare också eller är det något som växt fram nu i samband med DevOps-transformationen?	
40	Jag tror det kommer mycket mer nu, för förr var vi mer bekväma. Vi hade en separat IT-funktion, som support-enhet, kostnads-del, det där gamla traditionella IT-tänket. Men nu när vi rör oss mot affären och vi försöker kombinera, så måste ju alla försöka stappa upp och bli mer mogna i hur man tänker. IT är inte bara en gammal dator någonstans, utan koden är viktig, affärslogiken är viktig, och det här stödet. Men vi måste se till att stödet passar på produkten. Så därför är mognaden superviktig, både när det gäller fart, och det är kopplat till den stora transformationen. "Hur möter vi kundens behov", jo det är genom att snabbt vara där och tillgodose behoven. Och då är ju time-to-market superviktigt. Om vi då har gamla processer och gamla datacenterlösningar och IT-tänk, funktionellt IT-tänk, då får vi ju inte den farten som behövs.	V1
41	Hur arbetar ni med feedback? Arbetar utvecklarna nära kund så de får ta del av det direkt?	
42	Alla måste ju bli mer customer-centric och kundmedveten och komma nära och sätta sig in och emphasize. Och jobba med personas och hela den grejen. Sen är det väl ingen direkt interaktion med kund förutom vid själva touchpointen, där vi ser deras beteenden, vi mäter deras beteenden, vi kan A/B testa, och vi kan se vad är conversion om vi gör så här respektive så här. På så sätt får vi ju feedback, men det sker ju mer mekaniskt och automatiskt snarare än personliga möten. Det personliga mötet är ju mer retail, och där är vi jätteduktiga. Men online är det en annan grej för där måste vi ta mätpunkter på det.	V1, K2, V2
43	Internt inom teamen, får medarbetarna ta del av feedback där också?	
44	Oh ja. Det måste till. Drift är ju ingen separat grej, om inte operations är med när man tar fram en ny digital produkt då är det inte DevOps. För då är det inte en del av eftertanken, hur vi ska drifva det. Problemet vi har är ju att det finns en kärnkonflikt mellan development och operations. Den ena vill bara pusha upp nya features så fort som möjligt, och de andra vill säga stopp för de vill ha en stabil och säker miljö, och vill istället dra ned på förändringstakten. Och med DevOps kan man jobba tillsammans. Det finns också en nedåtgående spiral som finns i alla industrier, speciellt inom IT och digital, och det är att man fokuserar för mycket på features. "Vi	V1, A2

	<p>ska bli så kundmedvetna så då måste vi lägga på det här och det här och det här". Och då missar man de icke-funktionella kraven som skalbarhet, resilience, och location - var är vi och var möter vi kunden? Där är det viktigt att DevOps och operations-tänket kommer in och brygger feature-jakten. Så att vi kan väga upp så att vi bryter den här nedåtgående spiralen. För om man bara lägger på features, features, features, så kommer snart korthuset att falla.</p>	
45	Hur mycket av era processer är automatiserade?	
46	<p>Vi är rätt tidiga på det är, och traditionella, så jag skulle säga inte alls. Teamen börjar jobba mer och mer Scrum-baserat, vissa Kanban. Men förr har vi varit väldigt hungriga inom ITIL, så då har vi haft alla dessa Change Management, Incident Management, Problem Management och alla de bitarna som ingår där. Väldigt tunga processer och väldigt manuella processer. Det försöker vi ju röra oss ifrån och vara mer lättviktiga och vi jobbar med tvåvägssituationer med andra verktyg som Jira och Confluence för att jobba med features och stories och sånt. Vi måste fortfarande jobba med support som är kvar i ITIL så vi måste brygga, och det är något som ligger mig varmt om hjärtat - hur gör vi detta. Så jag jobbar faktiskt med detta i en DevOps-kontext [...]. Men inget mer automatiserat i den aspekten, men i alerting och buggar är ju automatiserat i de nya teamen. Vi har inte så många ännu, i den här resan har vi ett tiotal som har påbörjat. Där försöker vi sätta upp alerting på allting. Är det något fel inom infrastrukturen eller kör man någon debugging eller så. Eller moderna verktyg som Stack driver eller på Google cloud så kan vi följa upp och mäta, och tracability i koden, anrop som går fel och vad som tar lång tid. Sådant är automatiserat.</p>	K2, V2
47	Har det någon gång uppstått situationer där DevOps har varit problematiskt?	
48	<p>Problemet kan vara att vi är lite för kvicka. Och det är kopplat till mognadsgraden där igen. Man måste skynda långsamt. 80% av vår organisation är i mindset av ITIL och traditionell IT. Så om jag springer iväg lite i förväg blir det lätt att man lägger skulden på DevOps och agila arbetssätt. Det är mer kopplat till hur vi internt är i resan, och hur vi kan brygga de här olika värdena. Man måste göra det på ett bra sätt. Annars har jag inte hört att det ska vara något hinder. Vissa tänker "ni springer för fort, ni gör ert eget". Och det är helt fine, för jag är ute efter något som jag kallar aligned atonomy. Vi har atonomi i teamen men den är baserad på att ha en stark alignment kopplad till accountability. Så att ha klara mål, klar accountability, och klar uppföljning. Det finns en modell som beskriver detta: tight-loose-tight. Där man har tighta mål och tight uppföljning, men du har loose means. Så du är fri att göra vad du vill, eller på ren svenska: frihet under ansvar. Och det är lite det DevOps handlar om också. Men då är det vissa som kan missuppfatta det och känna att de bara springer iväg och de planerar inte med det här agila myterna - att det finns ingen planering, det är bara ad hoc. Men det finns det ju, just agile är ju baserat på väldigt tydlig sprintplanering och retrospect kopplat till det, "vad var bra och vad var inte bra?" så man får den feedbacken. Jag skulle säga att mognadsnivån är en, utbildning och awareness. När jag försöker sprida detta försöker jag ändå utgå ifrån ADKAR modellen: att man börjar med awareness, vad är detta, vad är DevOps? Sen desirability: "det vill vi också ha, det låter ju superbra". Sedan en kunskapsnivå kopplat till det. Och sen de sista har jag inte kommit till ännu. Men jag tror det viktiga är att man bygger en awareness först och främst, det är ju där man stöter på motstånd för att folk inte riktigt känner till vad det är. När agile kom var ju folk "nej vad är det där". Men man behöver bara förklara tydligt att det handlar om att vi bryter ned arbetet, vi får feedback på det, och efter varje sprint så har vi producerat en artefakt kopplat till arbetet. Det betyder inte att det är färdigt, men den här iterationen är färdig. Om man bara sprider det och är tydlig med vad det är så slipper man många av de problemen. Många av problemen handlar ju om att man gör sin egen tolkning. Alla människor egentligen.</p>	V1, O1, K2, K3
49	Upplever du att ni stött på mycket motstånd från de anställda i den här transformationen?	

50	Nej jag tror att många ingenjörer, många på golvet, kan ju detta. Många är ju ändå måna om att läsa artiklar och att läsa på, och filar hela tiden på sin profession - "hur blir jag en bättre programmerare?". Många ingenjörer känner ju till det här. Jag skulle nog säga att mellanskiktet och kanske till och med toppskiktet är de som inte alltid hänger med i industrin och trenderna. Trenderna kommer ju ofta från ingenjörerna på golvet, det är där det kommer in och vi får input ifrån. Men om bara top management skulle sitta i möten och fundera på hur man bäst får ROI på den här eller revenue på det där, utan att på riktigt veta hur, då får man ju motstånd. "Vi ska göra så här som vi alltid har gjort". Det är i sådana förändringsarbeten man stöter på motstånd.	K3, K1
51	Så det är ofta mer motstånd uppifrån?	
52	Ja, det skulle jag säga. I vårt fall är det mellanskiktet, för där har vi fått in ett nytt ledarskap som är visionärer och tänker framåt men som inte har <i>hur:et</i> . <i>Hur:et</i> kommer ju från ingenjörerna, var det är vi ska. Men sen är det ju mellanskiktet, hur får man med alla på resan? Där har vi ju stött på problem.	V1
53	Vad är problemet med just mellanskiktet?	
54	Om man tittar på den mer humana aspekten så är det ju kopplat till förståelse, utbildning, och awareness. Sedan finns det ju aspekter i form av: "vad hände med mitt område? Jag har ju alltid styrt det så här". Span of control vs span of influence, så är det ju alltid. Har du någon form av maktposition vill du ju alltid behålla den, och om något göra den större. Och då är det svårt när någon kommer och säger att "vi ska jobba mer så här, så nu är det teamen som bestämmer".	O3
55	Om man ser ur ett framtidsperspektiv, hur tror du sådana mellanchefer kommer att fortsätta påverkas?	
56	De kommer att försvinna mer och mer. Vi ser ju på ledarskap på ett nytt sätt. Inom X jobbar ju vi med ledarskap på vårt sätt, både kopplat till hela X men också kopplat till oss som retailer och franchisetagare. Sen tar vi ju mycket input från industrin, så vi kallar det inte servant leadership men vi har många av de aspekterna i hur vi tränar upp nya ledare. Att vi ska facilitera, vi ska enable:a coworkers, de ska känna sig empowered och vi ska stödja och supportera. Men också aspekter av situationsanpassat ledarskap. Men det gamla management-spåret där man tror att man kan managera människor, det vill vi bort ifrån. Gamla Taylorismen, det vill vi bort från.	O3, O2
57	Så ledarskap ska istället finnas mer utspritt i teamen?	
58	Precis.	O3
59	Bortsett från PO, har ni att någon annan i teamet kan ta på sig någon form av coachroll? Att flera får testa på att ha övergripande ansvar.	
60	Ja, vi har en idé om att offloada PO genom en roll som vi kallar produktspecialist. Om man kollar på PO-rollen så är den väldigt bred, man måste kunna jättemycket, man måste ha färsk kunskap, och man måste jobba med stakeholder management, du måste jobba med budgetering och forecasting. Man måste jobba med roadmappning. Så det är jättemycket, och det klarar man inte själv. Så då har vi produktspecialistroll som är mer nischad mot produkt. Om man kollar på en T-shaped-modellen så är det på en övergripande nivå mer mot produkt. Sen har vi ju seniora mjukvaruutvecklare	O3

	<p>som kan ta en mer tech lead-position. Så vi försöker fördela det i olika områden, så att vi inte bränner ut PO.</p>	
61	<p>Så ni försöker sprida ansvaret mer?</p>	
62	<p>Ja, lite så. Chapter Lead rollen är ju också sån. Om jag ska anställa 75 personer, så kan ju inte jag sitta och jobba med det personalansvaret. Så får man försöka dela upp det så att PO, när de kommer till mig och frågar om jag har någon DevOps Engineer. Om jag har en som är ledig, så för jag över teamet. Så det tar budgeten och day-to-day ansvaret, medan jag har kvar vissa aspekter av det hela i form av utbildningar, resor, och sånt. Så det är inte den klassiska People management-rollen, och det är heller ingen Tech-roll. Det är något nytt och vi har väl inte riktigt landat i hur vi kommer jobba med den. Det kommer ju från gamla Spotify-rörelsen och är matris egentligen, men de insåg ju också att när man växer så får man ju ändra den. Och det är det jag tror är det viktiga i det hela, att det finns ingen 'silver bullet'. Det måste vara baserat på din kontext. CDON jobbar också med den här modellen, men mycket mycket mindre chapters. 8 personer i ett chapter, då kan ju Chapter lead också sitta och koda, men samtidigt jobba med rekrytering och utbildning one-to-one. Men när det är större organisationer som denna, vi har 3000 människor inom Digital och ett Chapter kan ju på sikt bli upp mot 1000. Då går ju inte det. Då får man istället kolla på hur man kan skala ner det, och bryta ned det till flera chapters baserat på affärsområde eller funktionellt område. Men vi är inte där ännu. Vi är i början av resan, men vi har börjat fundera på hur kan vi hantera detta på ett bra sätt. Men där tror jag att vi kommer att stöta på problem, och det är ju skalningsfaktorn. Hur kan vi hantera detta i takt med att vi växer?</p>	<p>O3, O1, A1</p>
63	<p>Tror du att det är viktigt att vara mottaglig för förändring då?</p>	
64	<p>Oh ja. Och vara lyhörd, både från människorna som jobbar i detta men också baserat på vad vi producerar och resultatet. För om kvaliteten blir sämre så tyder det ofta inte på teknik, utan det är människor. Hur de interagerar, kommunicerar, och så vidare. Så där måste vi vara väldigt lyhörda. Men jag försöker mitigera det redan nu, och ta fram en plan, men jag vet inte.</p>	<p>A1, O2</p>
65	<p>Vad skulle du säga är de viktigaste egenskaper för att kunna leda i ständig förändring?</p>	
66	<p>Det är just att vara lyhörd och lyssna. Och kunna rätta kursen efter vägen. Lyckas man inte förutse, så får man snabbt ändra kurs. Det gäller att vara väldigt lyhörd. Och tekniskt kunnig! Jag tror det måste finnas en aspekt av teknik och kunskap om området. Jag som Chapter Lead är ju också väldigt kunnig inom det här området, och har byggt upp erfarenhet både från utveckling och operations och automation. Vi måste sätta expertis på de här positionerna. Det finns ju lite olika modeller på hur man gör leadership, om man ska gå en teknisk väg eller liknande. Men det håller inte, vi kan inte separera dem.</p>	<p>O2, A1, K3</p>
67	<p>Så om man ser till den traditionella projektledarrollen så måste man ha mer tekniska kunskaper för att kunna leda i sådana här sammanhang?</p>	
68	<p>Oh ja. Det tror jag. Med lyhördhet och support. Och supporten bygger på meritbaserad, jag måste bevisa att jag kan supportera på ett bra sätt och inte bara att det känns bra i magen. Och då är det mer övergripande att vi vill vara informerade av data så att vi kan ta beslut baserade på fakta. Men i den här världen så har vi inte fakta, och då är det magkänsla, men om du inte kan backa upp den med någonting,</p>	<p>O2, K1, K3</p>

	då kan det ju gå åt vilket håll som helst... Så backa upp med merit och erfarenhet, för annars kan det gå åt vilket håll som helst. [...]	
69	Är det något du skulle vilja tillägga om man ser till ledarskap i en DevOps-kontext?	
70	Ledarskap är ju något jag kommer jobba mycket med här. Man måste jobba mycket med awareness, vad det är och vad vi är ute efter. Vad är målet med det här? Och inte bara för att man ska följa med i en trend. Utan att det finns faktiska mål med att jobba i den här kontexten. Att vi är ju hela idea, build, deploy, run-kedjan, och det är för att vi har full kontroll och kännedom om allt som händer. Vi är inte beroende av andra, och vi sätter oss inte i en beroende-position. Och det tror jag är viktigt att få fram, och det handlar om utbildning. Och att känna till målbilden, annars kommer man aldrig komma loss från det gamla IT-träsket. Om man inte förstår beroende-situationen är aldrig bra, och om man då är beroende av att ett datacenter som snurrar någonstans, att man följer en process, att man följer de här stegen - folk vänjer sig vid det. Så folk börjar ifrågasätta det själva, så det kvittar om jag står och skriker. Om de själva inte inser det, så kommer det inte lyckas.	V1, O3, K3
71	Avslutningsvis, vad skulle du säga gör just er till en DevOps-organisation?	
72	Jag tror det är starkt ledarskap och en vision, dels från mig som Chapter Lead men också från top management att vi måste transformera oss, vi måste bli mer digitala, vi måste jobba mer product centric. Det är den största grejen som gör att vi kommer få rätt förutsättningar. En annan grej är att vi rör oss mer mot moln-miljöer. Så vi går ifrån de här gamla, stelbenta, rigida IT-processerna med hårdvara. Vi blir högre upp i abstraktionsnivån av det vi använder: containrar, serverless-funktioner, off-the-shelf grejer. Det tror jag är en grundförutsättning. Så starkt ledarskap, vision, och tekniska förutsättningar.	V1, A1

Referenser

- Armenise, V. (2015). Continuous delivery with Jenkins: Jenkins solutions to implement continuous delivery. In *Proceedings of the Third International Workshop on Release Engineering* (pp. 24-27). IEEE Press.
- Avolio, B. J., & Bass, B. M. (1995). Individual consideration viewed at multiple levels of analysis: A multi-level framework for examining the diffusion of transformational leadership. *The leadership quarterly*, 6(2), 199-218.
- Bang, S. K., Chung, S., Choh, Y., & Dupuis, M. (2013). A grounded theory analysis of modern web applications: knowledge, skills, and abilities for DevOps. In *Proceedings of the 2nd annual conference on Research in information technology*(pp. 61-62). ACM.
- Bass, B. M. (1985). Leadership and performance beyond expectations. Collier Macmillan.
- Bass, B. M., & Riggio, R. E. (2006). Transformational leadership. Psychology Press.
- Beyer, B., Murphy, N. R., Rensin, D. K., Kawahara, K., & Thorne, S. (2018). The Site Reliability Workbook: Practical Ways to Implement SRE. " O'Reilly Media, Inc."
- Davis, J., & Daniels, R. (2016). Effective DevOps: building a culture of collaboration, affinity, and tooling at scale. " O'Reilly Media, Inc."
- de França, B. B. N., Jeronimo Junior, H., & Travassos, G. H. (2016). Characterizing devops by hearing multiple voices. In *Proceedings of the 30th Brazilian Symposium on Software Engineering* (pp. 53-62). ACM.
- Dyck, A., Penners, R., & Lichter, H. (2015). Towards definitions for release engineering and DevOps. In *2015 IEEE/ACM 3rd International Workshop on Release Engineering*(pp. 3-3). IEEE.
- Erich, F., Amrit, C. A., & Daneva, M. (2014). Cooperation between information system development and operations: a literature review. In *8th International Symposium on Empirical Software Engineering and Measurement, ESEM 2014, Torino, Italy* (pp. 1-1). Association for Computing Machinery.
- Farenhorst, R., Sens, M., Schukken, F., Kruitbosch, T., & Loader, N. (2018). DASA DevOps Fundamentals. Olingo Consulting AB.
- Farroha, B. S., & Farroha, D. L. (2014). A framework for managing mission needs, compliance, and trust in the DevOps environment. In *2014 IEEE Military Communications Conference* (pp. 288-293). IEEE.
- Forsgren, N., Humble, J., & Kim, G. (2018). Accelerate: The science behind DevOps: Building and scaling high performing technology organizations. Portland, OR: IT Revolution.
- Gmeiner, J., Ramler, R., & Haslinger, J. (2015). Automated testing in the continuous delivery pipeline: A case study of an online company. In *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 1-6). IEEE.
- Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley, 2013-01.
- Humble, J., & Molesky, J. (2011). *Why Enterprise Adopt Devops to Enable Continuous Delivery*. Cutter IT Journal, Vol. 24, No. 8.

- Hussain, W., Clear, T., & MacDonell, S. (2017). Emerging trends for global DevOps: a New Zealand perspective. In 2017 IEEE 12th International Conference on Global Software Engineering (ICGSE) (pp. 21-30). IEEE.
- Hüttermann, M. (2012). *DevOps for developers*. Apress.
- Jacobsen, D. (2002). Vad, hur och varför: om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen. Studentlitteratur
- Kim, G. (2013). Top 11 Things You Need To Know About DevOps. Tillgänglig online: <http://www.bogotobogo.com/DevOps/Puppet/images/DevOps/Top11ThingsToKnowAboutDevOps.pdf> [Hämtad 2019-03-27]
- Kim, G., Willis, J., Debois, P., Humble, J., & Allspaw, J. (2016). *The Devops Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution.
- Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015). Dimensions of devops. In *International conference on agile software development* (pp. 212-217). Springer, Cham.
- Perera, P., Silva, R., & Perera, I. (2017). Improve software quality through practicing DevOps. In *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)* (pp. 1-6). IEEE.
- Rafferty, A. E., & Griffin, M. A. (2004). Dimensions of transformational leadership: Conceptual and empirical extensions. *The leadership quarterly*, 15(3), 329-354.
- Rakowski, J. (2015). Keep CALM and embrace DevOps. AppDynamics. Tillgänglig online: <https://www.appdynamics.com/> [Hämtad 2019-03-26]
- Ravichandran, A., Taylor, K., & Waterhouse, P. (2016). *DevOps for Digital Leaders*. CA.
- Roche, J. (2013). Adopting DevOps practices in quality assurance. *Commun. ACM*, 56(11), 38-43.
- Rodríguez, P., Haghghatkah, A., Lwakatare, L. E., Teppola, S., Suomalainen, T., Eskeli, J., ... & Oivo, M. (2017). Continuous deployment of software intensive products and services: A systematic mapping study. *Journal of Systems and Software*, 123, 263-291.
- Sharma, S. (2014). *Applying Lean and DevOps for better business outcomes*. IBM. Tillgänglig online: <https://www.ibm.com/community/> [Hämtad 2019-04-04]
- Ståhl, D., & Bosch, J. (2013). Experienced benefits of continuous integration in industry software product development: A case study. In *The 12th iasted international conference on software engineering, (innsbruck, austria, 2013)* (pp. 736-743).
- van Kelle, E., Visser, J., Plaat, A., & van der Wijst, P. (2015). An empirical study into social success factors for agile software development. In *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering* (pp. 77-80). IEEE.
- Walls, M. (2013). *Building a DevOps culture*. "O'Reilly Media, Inc."
- Walumbwa, F. O., Wang, P., Lawler, J. J., & Shi, K. (2004). The role of collective efficacy in the relations between transformational leadership and work outcomes. *Journal of Occupational and Organizational Psychology*, 77(4), 515-530.
- Wickett, J. (2016). *DevOps core values: CAMS*. DevOps Foundations. [Video] Tillgänglig online: <https://www.linkedin.com/learning/devops-foundations/devops-core-values-cams> [Hämtad 2019-03-26]
- Wiedemann, A. (2018). IT Governance Mechanisms for DevOps Teams-How Incumbent Companies Achieve Competitive Advantages. In *Proceedings of the 51st Hawaii International Conference on System Sciences*.
- Yukl, G., & Kaulio, M. (2011). *Ledarskap i organisationer*. Prentice-Hall