

DESIGNPRINCIPER FÖR DIGITALA DEVOPS- BEDÖMNINGSMODELLER

Kandidatuppsats i Informatik

Tobias Sandberg
Tobias Svensson

VT 2018:KANI10



HÖGSKOLAN
I BORÅS

Svensk titel: Designprinciper för digitala DevOps-bedömningsmodeller

Engelsk titel: Designprinciples for digital DevOps assessment models

Utgivningsår: 2018

Författare: Tobias Sandberg, Tobias Svensson

Handledare: Hannes Göbel

Abstract

Today, there is a great need for IT businesses to work with continuous improvement to keep competitive on the market. An important part of the work of continuous improvement is to assess and evaluate the existing situation with a view for creating profitable actions. To enable this assessment, businesses can use standards and models for process assessment. In the IT sector, many companies try to improve their work processes and combine development and operations. The work of this interconnection is often referred to as DevOps. The problem we address is that there are no simple digital tools that are contextualized for activities that want to improve collaboration between these departments. The existing class of DevOps assessment systems contains inadequate models, thus not support development and operational players in their work to assess their business to provide a basis for improvements. In order to improve the opportunities for DevOps operations while creating new knowledge, we have designed and evaluated a digital assessment model that can be used in practice. To fulfill the purpose, we have used the research method Action Design Research, which is a particularly suitable method of creating IT-related models in a real context. The result confirms that existing assessment models are not sufficient and that the problem is generalizable as a class of problems. An operating digital model will also be presented with the purpose of assessing activities from a DevOps context. In developing the artifact, three general design principles have also been identified which developers and practitioners should follow for design of future assessment models for DevOps. The principles imply that, in the creation of a DevOps assessment model in an IT context, (i) a grading scale should be divided into four capacity levels, (ii) statements used in the model should be changeable and adaptable as organizations are unique and (iii) should be developed so that development and operation can perform the evaluation together.

Keywords: DevOps, Action Design Research, Design artifact, Process assessment, Assessment model

Sammanfattning

Det finns idag ett stort behov för IT-verksamheter att arbeta med ständiga förbättringar för att hålla sig konkurrenskraftiga på marknaden. En viktig del i arbetet med ständiga förbättringar är att bedöma och utvärdera den befintliga situationen i syfte att skapa bra åtgärder. För att möjliggöra denna bedömning kan verksamheter använda sig av standarder och modeller för processbedömning. I IT-sektorn anstränger sig många företag med att förbättra arbetsprocesserna och brygga samman sina utvecklings- (eng. Development) och driftsavdelningar (eng. Operations). Arbetet med denna sammanlänkning benämns ofta som DevOps. Problemet som vi adresserar är att det saknas enkla digitala verktyg som är kontextualiserade för verksamheter som vill förbättra sitt samarbete mellan dessa avdelningar. Den befintliga klassen av system för DevOps-bedömning innehåller modeller som är otillräckliga och stödjer därmed inte utvecklings- och driftsaktörer i deras arbete att bedöma sin verksamhet för att ge beslutsunderlag för förbättring. I syfte att förbättra möjligheterna för DevOps-verksamheter och samtidigt skapa ny kunskap har vi designat och utvärderat en digital bedömningsmodell som kan användas i praktiken. För att uppfylla syftet har vi använt oss av forskningsmetoden Action Design Research som är särskilt lämplig metod vid skapande av IT-relaterade modeller i en verklig kontext. Resultatet bekräftar att befintliga bedömningsmodeller inte är tillräckliga och att problemet är generaliserbart som en klass av problem. En operativ digital modell kommer även presenteras med syfte att bedöma verksamheter ur ett DevOps-kontext. Vid utveckling av artefakten har även tre generella designprinciper identifierats vilka utvecklare och praktiker bör följa vid design av framtida bedömningsmodeller för DevOps. Principerna innebär att vid skapandet av en bedömningsmodell för DevOps i en IT-kontext bör det (i) användas en betygsskala uppdelad på fyra kapacitetsnivåer, (ii) påståenden som används i modellen bör vara förändrings- och anpassningsbara då verksamheter är unika, samt att (iii) modellen bör utvecklas så att development och operations kan utföra utvärderingen tillsammans.

Nyckelord: DevOps, Action Design Research, Designartefakt, Processutvärdering, Bedömningsmodell

Innehållsförteckning

1	Inledning.....	- 1 -
1.1	Bakgrund.....	- 1 -
1.2	Problemformulering.....	- 2 -
1.3	Syfte och frågeställning.....	- 2 -
1.4	Förväntat resultat.....	- 2 -
1.5	Avgränsning.....	- 2 -
1.6	Disposition.....	- 3 -
2	Relaterad litteratur.....	- 4 -
2.1	DevOps.....	- 4 -
2.2	Generella bedömningsmodeller.....	- 6 -
2.2.1	CMMI.....	- 6 -
2.2.2	ISO/IEC 15504.....	- 7 -
2.2.3	Sammanfattande kritik till generella modeller.....	- 8 -
2.3	Specifika bedömningsmodeller för DevOps.....	- 8 -
2.3.1	DevOps maturity model.....	- 9 -
2.3.2	Improve.....	- 10 -
2.3.3	Solinea DevOps Maturity Model Matrix.....	- 10 -
2.3.4	The Continuous Delivery Maturity Model.....	- 12 -
2.3.5	The LeaseWeb maturity model for DevOps teams.....	- 14 -
2.3.6	Sammanfattande kritik till specifika modeller.....	- 15 -
2.4	Behov av digital bedömningsmodell för DevOps.....	- 17 -
3	Metod.....	- 18 -
3.1	Metodval.....	- 18 -
3.2	Action design research.....	- 19 -
3.2.1	Problemformulering.....	- 20 -
3.2.2	Bygga, intervention och utvärdering.....	- 20 -
3.2.3	Reflektion och lärande.....	- 26 -
3.2.4	Formalisering av lärande.....	- 26 -
4	Resultatanalys.....	- 27 -
4.1	Mål med IT-artefakt.....	- 27 -
4.2	Beskrivning av IT-artefakt.....	- 29 -
4.3	Beskrivning av designprinciper.....	- 31 -
4.4	Utvärdering av IT-artefakt och designprinciper.....	- 33 -
4.4.1	Utvärderingsepisod 1.....	- 33 -
4.4.2	Utvärderingsepisod 2.....	- 35 -
4.4.3	Utvärderingsepisod 3.....	- 37 -
4.4.4	Utvärderingsepisod 4.....	- 39 -
4.4.5	Sammanfattning av utvärdering.....	- 42 -
5	Slutsats.....	- 44 -
5.1	Bidrag i form IT-artefakt.....	- 44 -
5.2	Bidrag i form av designprinciper.....	- 44 -
5.3	Reflektion och fortsatt forskning.....	- 45 -

Figurförteckning

Figur 1: DevOps maturity model (Mohammed 2015, s. 53).....	- 9 -
Figur 2: Solinea DevOps Maturity Model (Parks 2016).....	- 11 -
Figur 3: The Continuous Delivery Maturity Model (Rehn, Palmborg, Boström 2013)	- 13 -
Figur 4: Problemgraf	- 17 -
Figur 5: Utvärderingsepisod 1.....	- 22 -
Figur 6: Utvärderingsepisod 2.....	- 23 -
Figur 7: Utvärderingsepisod 3.....	- 24 -
Figur 8: Utvärderingsepisod 4.....	- 25 -
Figur 9: Målgraf	- 28 -
Figur 10: Uppfyllande av mål i verktyget Improve	- 30 -
Figur 11: Uppfyllande av designprinciper i verktyget Improve.....	- 32 -

Tabellförteckning

Tabell 1: CMMI Kapacitetsnivåer (eng. capability levels).....	- 7 -
Tabell 2: ISO/IEC 15504 Kapacitetsnivåer (eng. Capability levels)	- 8 -
Tabell 3: The LeaseWebs maturity model for DevOps teams (Poelwijk 2016)	- 15 -
Tabell 4: Utvärderingsepisod 1 - Delning av erfarenheter.....	- 34 -
Tabell 5: Utvärderingsepisod 2 - Helhetsperspektiv	- 36 -
Tabell 6: Utvärderingsepisod 4 - Helhetsperspektiv	- 40 -

1 Inledning

Syftet med kapitel ett är att ge en beskrivning till den bakgrund som uppsatsen vilar på. Bakgrunden kommer mynna ut i en problemdiskussion som i sin tur leder vidare till en forskningsfråga samt en syftesbeskrivning, vidare kommer avgränsning beskrivas samt det förväntade resultatet för uppsatsen innan kapitlet avslutas med en disposition av uppsatsen. Kapitlet fortsätter med 1.1) Bakgrund, 1.2) Problemformulering, 1.3) Syfte och frågeställning, 1.4) Avgränsning och 1.5) Disposition.

1.1 Bakgrund

Det finns idag ett stort behov för organisationer att arbeta med ständiga förbättringar för att hålla sig konkurrenskraftiga på marknaden. I arbetet med ständiga förbättringar är det ofta fokus på organisationers processer vilka måste förbättras då dessa påverkar och styr affärsverksamheten (CMMI Team 2010). Det är även av stor vikt att förbättra processer ur ett tjänsteperspektiv då både det populära ramverket *ITIL* och den internationella standarden för ITSM *ISO/IEC 20000* påpekar vikten av processförbättring för bättre IT-tjänster (Shrestha, Cater-Steel, Toleman, Tan 2014). Det existerar idag flera olika ramverk som syftar till att stödja arbetet med att förbättra processer som exempelvis *ISO9000*, *TQM*, *Six Sigma* och *Lean*. Tidigare forskning visar att användning av mjukvara för att tillämpa dessa metoder påskyndar processanpassning och förbättring (Shrestha et al 2014), samtidigt visar studier att verktyg som hjälpmedel för bedömning av mjukvaruprocesser påverkar kostnad och tid (Adalı, Özcan-Top, Demirörs 2016). För att identifiera förbättringsmöjligheter av processer används processbedömningar (Shrestha et al 2014), vilka ofta är baserade på modeller som exempelvis *CMMI* och *ISO 15504* vilka är de mest populära process-bedömningsmodellerna för organisationer som hanterar mjukvara (Adalı, Özcan-Top, Demirörs 2016).

I IT-sektorn finns ett framväxande problem inom mjukvaruutveckling och drift som innebär att många befintliga IT-system brister i sin effektivitet och därmed inte uppfyller de värden som avsetts (Kavis 2014). Problemet beror bland annat på att verksamhetens avdelningar inte kommunicerar med varandra och arbetar i så kallade *silos*. Begreppet silo kommer från att de olika avdelningarna som arbetar med utveckling (development)¹ respektive teknisk drift (operations)² är helt avgränsade och oberoende av varandra (Kavis 2014). Bristen på kommunikation gör att Dev ofta saknar de miljöer och verktyg som de behöver för att vara produktiva, vilket leder till att Ops får tillhanda programvara med bristande kvalitet vilket kan leda till att de blir svåra att underhålla (Kavis 2014). Det finns ett gap mellan Dev och Ops i många organisationer idag vilket utgör ett stort hinder för snabba och frekventa leveranser av mjukvara. En anledning till att det finns gap mellan avdelningarna är att de saknar gemensamma mål, processer och synsätt (Wettinger, Breitenbucher, Leymann 2014). Å ena sidan beskriver Hüttermann (2012) att Dev har ett behov av förändring i form av nya funktioner, buggfixar och annat arbete baserat på förändringsbehov. Å andra sidan finns ibland en rädsla för förändring hos Ops eftersom den kan påverka det stabila tillstånd som försöker upprätthålla på systemen. För att lösa ovanstående problem har en rörelse växt fram som kallas DevOps (Kavis 2014). DevOps har som mål att genom samarbete inom verksamheter brygga samman de isolerade silorna Dev och Ops (Lwakatere, Karvonen, Sauvola, Kuvaja, Olsson, Bosch, Oivo 2016). Att dessa avdelningar eller funktioner har ett samarbete är av stor vikt för att snabbt och frekvent kunna utveckla och leverera mjukvara till kunder (Lwakatere et al 2016).

¹ Hädanefter kommer roller som berör utveckling benämnas som *Dev*.

² Hädanefter kommer roller som berör drift benämnas som *Ops*.

1.2 Problemformulering

Likt arbetet med processförbättringar i verksamheter behöver även samarbete mellan Dev och Ops utvärderas och bedömas för att förbättras (Riley 2015). I syfte att stödja samarbetet mellan avdelningarna har olika ramverk för DevOps presenterats (Parks 2016; Rehn, Palmborg, Boström 2013; Poelwijk 2016). Dessa ramverk eller modeller hjälper Dev och Ops att implementera och bedöma olika egenskaper ur ett DevOps-kontext. Problem som identifierats med dessa modeller är att de saknar en teoretisk grund och transparens i hur de utformats, de har baserats på CMMIs mognadsnivåer, de har inte tillräckligt stark koppling till DevOps underliggande principer, och slutligen att de inte har implementerats i ett digitalt verktyg. Ovanstående problem kan tillsammans leda till att bedömning av verksamheter genomförs på ett bristfälligt sätt. Vi menar att de problem, som beskrivits ovan, har uppstått då det saknas designkunskap för hur digitala bedömningsmodeller för DevOps bör utformas, vilket är det huvudsakliga problemet som adresseras i studien.

1.3 Syfte och frågeställning

För att möjliggöra för verksamheter att bedöma sin DevOps satsning på ett korrekt sätt, baserat på forskning snarare än individuella praktikers erfarenheter, har en bedömningsmodell skapats. Vi förespråkar att en sådan bedömningsmodell kan hjälpa verksamheter i deras arbete med att kontinuerlig förbättra sina DevOps-relaterade processer. Vi definierar en bedömningsmodell enligt Albliwi, Antony och Arsheds (2014) definition där de beskriver en mognadsmodell som ett verktyg som används för att utvärdera styrkor och svagheter i verksamheter i syfte att ge vägledning till förbättring. Modellen utvärderar kvalitetsstandarder och best practice i verksamheten vilket även jämförs mot andra organisationer. Vårt syfte är även att skapa designkunskap som kan användas vid framtida utveckling av bedömningsmodeller som skall hantera samma problemområde som adresserats. Att skapa en sådan designkunskap är viktigt då det hjälper utvecklare att basera design på grundade principer istället för gissningar (Hevner, March, Park, Ram 2004). För att skapa en sådan artefakt och designkunskap behöver följande forskningsfråga besvaras:

Hur bör en IT-artefakt som möjliggör bedömning av DevOps utformas?

1.4 Förväntat resultat

Vi förväntar oss att vårt resultat skall vara till nytta för både forskningssamhället och praktiker. Resultatet till forskningssamhället förväntas i form av ett antal designprinciper som kan förändra eller komplettera existerande principer för modeller som används för att bedöma verksameters DevOps. Designprinciperna förväntas därmed kunna användas vid framtida utveckling av IT-artefakter som syftar till att hantera samma klass av problem/lösningar som beskrivits i bakgrunden och problemformuleringen. Vad en designprincip är beskrivs vidare i metodkapitlet under rubriken *Action design research* (se sektion 3.2). Vi förväntar oss även att resultatet kommer vara till nytta för praktiker då vi skapar en digital bedömningsmodell som kan användas av verksamheter som vill utvärdera och förbättra sina DevOps-relaterade processer.

1.5 Avgränsning

DevOps kan användas inom flera områden och verksamheter (Hüttermann 2012), vi har dock enbart utvärderat artefakten i samspel med IT-sektorn som är mer välbekanta med begreppet vilket underlättat utvärdering och återkoppling vid utveckling av modellen. Studien har även

avgränsats från processen att skapa de påståenden (se bilaga 1) som använts i artefakten för att bedöma verksamhetens DevOps. Artefakten kommer istället att utgå från de påståenden som skapats av forskningsgruppen InnovationLab på Högskolan i Borås i deras forskningsprojekt i datadriven innovation. Studien kommer även att avgränsas från utveckling och utvärdering av verktyget Improve där artefakten implementerats. Exempelvis när artefakten utvärderas huruvida den är enkel att använda och förstå utvärderas inte gränssnitt eller interaktion. Verktyget avgränsas då syftet med forskningen har varit att utveckla en modell och utforma bedömningsnivåer, inte att utveckla ett digitalt verktyg för modellen.

1.6 Disposition

Genom att presentera uppsatsens disposition önskar vi ge en ökad förståelse för studiens uppbyggnad och konstruktion.

Kapitel 1 - Inledning

Här presenteras bakgrunden för området som sedan resulterar i en problemformulering som leder till vårt problemområde. Därefter presenteras studiens syfte och forskningsfrågor. I kapitlet redogörs även det förväntade resultatet av vår studie samt avgränsning.

Kapitel 2 - Relaterad litteratur

I relaterad litteratur granskas och definieras DevOps i syfte att skapa kontext för bedömningsmodellen. Vidare kommer generella bedömningsmodeller granskas vilka kan ses som en samling eller klass av lösningar snarare än enskilda instanser då modellerna anses vara breda och applicerbara på flera problemområden. Därefter kommer enskilda instanser av lösningar för specifika bedömningsmodeller för DevOps att analyseras. Modellerna kommer även kritiserats vilket leder till identifiering av behov för utveckling av en digital bedömningsmodell för DevOps.

Kapitel 3 - Metod

I metodkapitlet motiveras och beskrivs den metod som valts för att besvara forskningsfrågan. Samtliga processer som genomförts vid utvecklandet av artefakten beskrivs utifrån metodens fyra steg för att ge en tydlig bild över studiens tillvägagångssätt. Då utveckling av artefakten skett iterativt har kapitlet strukturerats efter de utvärderingsepisoder som genomfördes.

Kapitel 4 - Resultatanalys

I resultatanalysen presenteras de mål som skapats för artefakten samt hur dessa använts vid utvärdering av IT-artefakten. Den IT-artefakt som skapats under studien för att lösa problemområdet presenteras, samt hur införskaffad kunskap har hjälpt oss identifiera designprinciper. Här presenteras också den utvärdering och sammanfattning av resultaten från den iterativa utvecklingsprocessen för IT-artefakten och designprinciperna.

Kapitel 5 - Slutsats

I slutsatsen argumenteras för hur studien lyckats besvara frågeställning genom att presentera en artefakt och tre designprinciper. Avslutningsvis diskuteras och kritiserats metodval vilket leder till förslag för vidare forskning.

2 Relaterad litteratur

I kapitel två kommer inledningsvis DevOps granskas och definieras med syfte att skapa kontext för bedömningsmodellen. Vidare kommer generella bedömningsmodeller beskrivas vilka kan ses som en klass av lösningar snarare än enskilda instanser då modellerna anses vara breda och applicerbara på flera problemområden. Därefter granskas enskilda instanser av lösningar för problemområdet, det vill säga specifika bedömningsmodeller för DevOps. Vidare kritiserar modellerna vilket leder till identifiering av behov för utveckling av en digital bedömningsmodell för DevOps. Kapitlet innefattar 2.1) DevOps, 2.2) Generella bedömningsmodeller och 2.3) Specifika bedömningsmodeller för DevOps, 2.4) Behov av digitala för bedömningsmodeller.

2.1 DevOps

För att besvara forskningsfrågan behöver begreppet DevOps definieras då det skapar en kontext för bedömningsmodellen. Historiskt sett har systemutveckling skett i separerade silos i verksamheter uppdelade på kompetens, dessa silos benämns ofta som Dev och Ops (Hüttermann 2012). Dev specialiserar sig på kod vid systemutveckling medan Ops fokuserar på IT-systemets funktion i verksamheten och sköter dess underhåll (Hüttermann 2012). Vardera silon har sina egna mål baserat på arbetsuppgifter. Dev's framgång mäts efter deras förmåga att snabbt leverera nya system medan Ops mäter sin framgång utifrån IT-systemets stabilitet (Hüttermann 2012). Då Dev och Ops arbetar i separerade silos uppstår kommunikationsbrister mellan avdelningarna som i sin tur påverkar verksamheten på flera nivåer (Hüttermann 2012). Denna kommunikationsbrist resulterar i att (i) Dev och Ops inte delar samma mål för verksamheten, att (ii) det inte finns tydliga tillvägagångssätt för hur Dev och Ops skall hantera systemförändring, lanseringar eller underhålla dem samt (iii) att de använder olika verktyg för att utföra sina uppgifter (Hüttermann 2012). Följderna blir att det skapas en verksamhetskultur där utvecklare saknar rätt systemmiljöer och verktyg vid utveckling, vilket hindrar effektivitet och produktivitet (Kavis 2014). Samtidigt tvingas Ops underhålla IT-system som "kastats över väggen" vilket innebär att de saknar insikt för att kunna underhålla IT-system på ett korrekt sätt (Kavis 2014).

Som en motreaktion till ovanstående problem med långsamma systemleveranser och ostabila IT-system har en rörelse växt fram av praktiker kallat DevOps (Kavis 2014). DevOps har sina rötter i agila systemutvecklingsprinciper vilka verkar för snabba leveranser och en god design som stärker anpassningsförmågan i verksamheter (Verona 2016; Beck et al 2001). Idag finns ett behov för verksamheter att vara flexibla för att klara av en konkurrerande miljö. Bass, Weber och Zhu (u.å.) styrker detta och menar att DevOps främst skapades utifrån behovet av att snabba leveranser anses vara kritiskt för att driva en effektiv verksamhet. The agile manifestos skrevs då flera individer ville förbättra det rådande tillvägagångssättet att utveckla system och såg ett nytt sätt att arbeta med systemutveckling i praktiken. Ett utkast ur det Agila manifestet (Beck et al 2001) ser ut som följande:

Individer och interaktioner framför processer och verktyg
Fungerande programvara framför omfattande dokumentation
Kundsamarbete framför kontraktförhandling
Anpassning till förändring framför att följa en plan
Det vill säga, medan det finns värde i punkterna till höger,
Värdesätter vi punkterna till vänster mer.

Enligt Verona (2016) har DevOps sitt ursprung ur den första av manifestets principer: individer och interaktioner före processer och verktyg. DevOps syfte är att bryta ner de silos som verksamheter idag har och istället skapa avdelningar i verksamheten som kommunicerar med varandra (Verona 2016; Smeds, Nybom, Porres 2015). Även om verktyg som används i verksamheter bidrar till effektivitet och ibland är kritiska för utförandet av processer, måste de användas på rätt sätt. Verona (2016) menar att individer och interaktioner bidrar till en mer agil arbetskultur då kommunikationen mellan verksamhetens funktioner effektiviserar de verktyg och processer de idag behandlar. DevOps är ett relativt nytt och ofta missförstått begrepp inom IT-branschen, att definiera begreppet kan därmed ses som särskilt viktigt (Kavis 2014; Smeds, Nybom, Porres 2015). Praktiker har tolkat termen på flera olika sätt: som en IT-roll, en hybrid av utvecklare och systemadministratör, en superadministratör som är specialist inom både utformning av system, utveckling och underhåll. Följderna av den sistnämnda tolkningen innebär att det skapas ytterligare en silo vilket sätter verksamheten i samma situation den tidigare befann sig i (Kavis 2014). DevOps är snarare en kultur, ett nytt sätt att tänka kring hur verksamheter utvecklar och levererar system. DevOps vill motverka bildningen av silos och fokuserar istället på samarbete mellan Dev, Ops, kvalitetssäkring, produkt och förvaltning (Kavis 2014). DevOps är en sammansättning av funktionerna Dev och Ops i en verksamhet (Hüttermann 2012). Ur ett utvecklarperspektiv förklaras Dev som mjukvaruutvecklare inkluderat programmerare, testare och kvalitetsgranskare och Ops som personal med fokus driftsättning och driver produktens infrastruktur, system administratörer, databasadministratörer och nätverkstekniker. DevOps exemplifierar de aktiviteter i verksamheter som effektiviserar leveranser av system med betoning på den interna kommunikationen och kunskapen för att förbättra leveranscykeln. DevOps berör även produktkvaliteten då system utvecklas med individuella krav och grundläggande villkor (Hüttermann 2012).

Inom DevOps rörelsen nämns ibland akronymen CAMS eller CALMS, vilket står för Culture, Automation, Lean, Measurement och Sharing. Syftet med CAMS är att skapa en inställning vid systemutveckling där kvalitetssäkring är beroende av ett fungerande samarbete mellan Dev och Ops (Kavis 2014). CALMS grundas i en; (C) kultur där människors kunskaper är en mer väsentliga än de verktyg och processer verksamheten använder samt att system utvecklas av och för människor; (A) Automatisering möjliggör direkt återkoppling och är därför en viktig del inom DevOps. (L) Lean innebär minimering och effektivisering av processer i verksamheten; (M) Mätvärden gör det möjligt att utvärdera processer i verksamheten i syfte att arbeta med ständig förbättring; (S) Sharing skapar en kultur där verksamheten delar idéer, processer och verktyg (Hüttermann 2012; Riley 2014). CALMS förändrar traditionella roller, det vill säga att det inte längre nödvändigtvis måste vara Dev som är ansvariga för kod, testare som är ansvariga för test och Ops som ansvarar för att underhålla systemet. CALMS står för att hela verksamheten ansvarar för de system som tillhandahålls inklusive mål, leverans och kvalitet (Kavis 2014).

2.2 Generella bedömningsmodeller

I syfte att strukturera problemet, identifiera lösningsmöjligheter samt vägleda designen av artefakten har de två generella bedömningsmodeller identifierats, *CMMI* och *ISO 15504*. Modellerna har valts då är de två mest förekommande för utvärdering av processer (Adalı, Özcan-Top, Demirörs 2016). *CMMI* anses vara en särskilt lämplig grund för att bygga modeller med syfte att förbättra processer inom DevOps-projekt, dock behöver modellerna anpassas för rätt kontext (Rong, Zhang, Shao 2016). De två modellerna är så pass generella att de kan se som en klass av lösningar snarare än enskilda instanser, då de anses vara breda och applicerbara på flera problemområden (Jokela, Siponen, Hirasawa, Earthy 2006). Klassen av problem som dessa modeller hanterar är processbedömning med syfte att identifiera förbättringsmöjligheter.

2.2.1 CMMI

CMMI är baserat på ett processororienterat synsätt som fokuserar på tre kritiska områden som en organisation kan fokusera på för att förbättra sin affärsverksamhet. Områdena är människor, procedurer och metoder samt verktyg (*CMMI Team* 2010). Hur dessa relaterar till varandra avgörs av de processer som finns i organisationen. Det är därmed processer som styr affärsverksamheten, möjliggör skalbarhet och införlivar förbättringsmöjligheter. Processer tillåter även organisationer att utnyttja resurser och undersöka affärstrender. Att arbeta med förbättring av processer är därmed viktigt för att verksamheter som vill hålla sig konkurrenskraftiga på marknaden. När en organisation utvärderar sina processer med hjälp av *CMMI* används så kallade *nivåer*. Nivåer beskriver en evolutionär väg som organisationer bör följa när de vill förbättra sina processer som används vid utveckling av produkter eller tjänster. Processerna kan utvärderas enligt två sätt och benämns av *CMMI* som *continuous*- och *staged representations*. *Continuous representations* resulterar i att verksamheter kan uppnå *kapacitetsnivåer (capability levels)*, *staged representations* leder istället till att verksamheten uppnår *mognadsnivåer (maturity levels)*. Det som skiljer *staged* och *continuous* är att *staged mognadsnivåer* bedömer det generella tillståndet för processer i en organisation ur ett helhetsperspektiv medan *staged kapacitetsnivåer* karakteriserar tillståndet hos en organisations processer inom ett individuellt processområde (*CMMI Team* 2010).

*CMMI*s *continuous representation capability levels* delas in i fyra kapacitetsnivåer (nivå 0-3), dessa är: *ofullständig (incomplete)*, *utförd (performed)*, *hanterad (managed)* och *definierad (defined)* (*CMMI Team* 2010) vilka framgår i tabell 1. För att klättra en nivå måste samtliga mål för nuvarande nivå vara uppfyllda. Nivå 0 (*ofullständig*) innebär att en process antingen saknas helt eller enbart utförs till viss del. Nivå 1 (*utförd*) uppnås när en process används och kan utföra det arbete som krävs för att producera arbetsprodukter och uppfyller de uppsatta målen för processområdet. Nivå 2 (*hanterad*) innebär att en process kan planeras och utföras efter verksamheten *polycys*, det finns tillräckligt med resurser för att producera kontrollerad output, den involverar rätt intressenter, övervakas, kontrolleras, granskas och utvärderas utifrån sin processbeskrivning. Nivå 2 hjälper till att säkerställa att aktiviteter genomförs på korrekt sätt även under stress. Nivå 3 (*definierad*) är den högsta nivån i *CMMI*s kapacitetsnivåer. Nivån innebär att en process är skräddarsydd baserad på någon av organisationens standardprocesser och att den följer uppsatta riktlinjer för processer. Processen har en beskrivning som underhålls och processrelaterad erfarenhet delas i organisationen. Den stora skillnaden mellan nivå två och tre är omfattningen på processernas standard, beskrivning, och procedurer. Nivå tre kräver mer standardiserade processer med möjlighet till individuella anpassningar i enlighet med organisationens riktlinjer snarare än att alla processer varierar i utformning. En annan skillnad på nivå två och tre är att den tredje nivån har en mer rigorös beskrivning över processerna som tydligt visar syfte, inputs,

startkriterier, roller, mätvärden, verifieringssteg, outputs och avslutskriterier. På nivå tre hanteras processer mer proaktivt med en förståelse för processers interrelationer och har tydliga mätvärden för både processer och arbetsprodukter (CMMI Team 2010).

Tabell 1: CMMI Kapacitetsnivåer (eng. capability levels)

Nivå	Namn
0	Ofullständig (Incomplete)
1	Utförd (Performed)
2	Hanterad (Managed)
3	Definierad (Defined)

En utvecklad beskrivning över CMMIs staged representations och dess mognadsnivåer kommer inte redovisas i denna studie då artefakten som utvecklades inte har baserats på dessa mognadsnivåer. Motivering till varför artefakten inte baserats på dessa mognadsnivåer ges i sektion 2.2.3 *sammanfattande kritik till generella bedömningsmodeller*.

2.2.2 ISO/IEC 15504

ISO/IEC 15504 som även kallas SPICE som står för Software Process Improvement and Capability Determination utvecklades på uppdrag av ISO (International Organisation for Standardisation) och IEC (International Electrotechnical Commission) med syfte att skapa en standard för förbättring, fastställande av förmågor och självutvärdering av processer (Adali, Özcan-Top, Demirörs 2016). Standarden utvecklades initialt inriktad mot processer för mjukvaruutveckling, men har nu blivit en generell processbedömningsstandard (Shrestha et al 2014). ISO/IEC 15504 använder precis som CMMI av Capability levels för bedömning av processer. För att uppnå en viss capability level på en process behöver processen först uppnått vad de kallar för processattribut, dessa attribut liknar de som används i CMMIs modell men är istället indelad i sex nivåer (nivå 0-5), dessa är: *ofullständig (incomplete)*, *utförd (performed)*, *hanterad (managed)*, *etablerad (established)*, *förutsägbar (predictable)* och *optimerad (optimizing)* vilka framgår i tabell 2. Nivå 0 (ofullständig) innebär att en process syfte och resultat inte uppnås, dessutom saknas tillräckligt med bevis för syftet med processen. Det finns inga processattribut på denna nivå. Nivå 1 (utförd) säkerställer att en process har ett syfte och genererar ett resultat. Nivå 2 (hanterad) innebär att processer har arbetsprodukts- och prestationshantering. Processerna planeras, övervakas och hanteras. Nivå 3 (etablerad) har uppnåtts när processer har definierats enligt en viss procedur och implementerats baserat på denna definition. Nivå 4 (förutsägbar) ser till att processers prestanda är förutsägbara. Denna förutsägbarhet uppnås genom fokusering på kvantitativ data runt processen. Nivå 5 (optimerad) riktar in sig på att arbeta med ständiga förbättringar på processer. Arbetet med ständiga förbättringar innebär att processer optimeras för att uppfylla de behov som ställs på en verksamhet.

Tabell 2: ISO/IEC 15504 Kapacitetsnivåer (eng. Capability levels)

Nivå	Namn	Attribut
0	Ofullständig (Incomplete)	
1	Utförd (Performed)	1.1 Process Performance
2	Hanterad (Managed)	2.1 Performance Management 2.2 Work Product Management
3	Etablerad (Established)	3.1 Process Definition 3.2 Process Deployment
4	Förutsägbar (Predictable)	4.1 Process Measurement 4.2 Process Control
5	Optimerad (Optimizing)	5.1 Process Innovation 5.2 Process Optimization

2.2.3 Sammanfattande kritik till generella modeller

Vid analys av de generella bedömningsmodellerna CMMI och ISO/IEC 15504 framgår det att modellerna ger en bra grund för konstruktion av en artefakt vars syfte är att utvärdera verksamheter ur ett DevOps-perspektiv (Rong, Zhang, Shao 2016). Samtidigt kan det riktas kritik mot CMMIs staged representations och dess väg mot högre mognadsnivåer som uppnås genom konfigurationer av flera komplexa organisatoriska och miljöförhållanden snarare än genom en linjär sekventiell väg (Lasrado, Vatrupu, Andersen 2016). Flertalet studier visar även att CMMIs högre mognadsnivåer (nivå 4-5) enbart kan uppnås genom användning av agila metoder (Rong, Zhang, Shao 2016), samtidigt påstår Fritzsche och Keil (2007) att några av CMMIs processområden på mognadsnivå tre, samt de flesta områdena på nivå fyra och fem inte går att uppnå utan att ta bort några av de agila grundstenar som metoderna vilar på då dessa grundstenar är i konflikt med CMMIs processområden. Borttagning av dessa grundstenar skulle innebära att de agila metoderna skulle försvagas och flera av deras fördelar skulle tas bort. Att påtvingat förändra de agila metoderna går även emot CMMIs syfte att förbättra processer genom att göra dem agila (Fritzsche, Keil 2007). Det går därmed att motivera att det inte är lämpligt att använda sig av CMMIs mognadsnivåer vid utformning av en bedömningsmodell för DevOps, då begreppet har grundats från de agila tankesättet (Verona 2016). Mognadsmodeller har även ständigt blivit kritiserade för sin oförmåga att ta hänsyn till resultat och prestanda vid utvärdering. Det vill säga att det är möjligt att avancera i mognadsnivåer utan att ha förbättrat affärsprocesser (Albliwi, Antony, Arshed 2014).

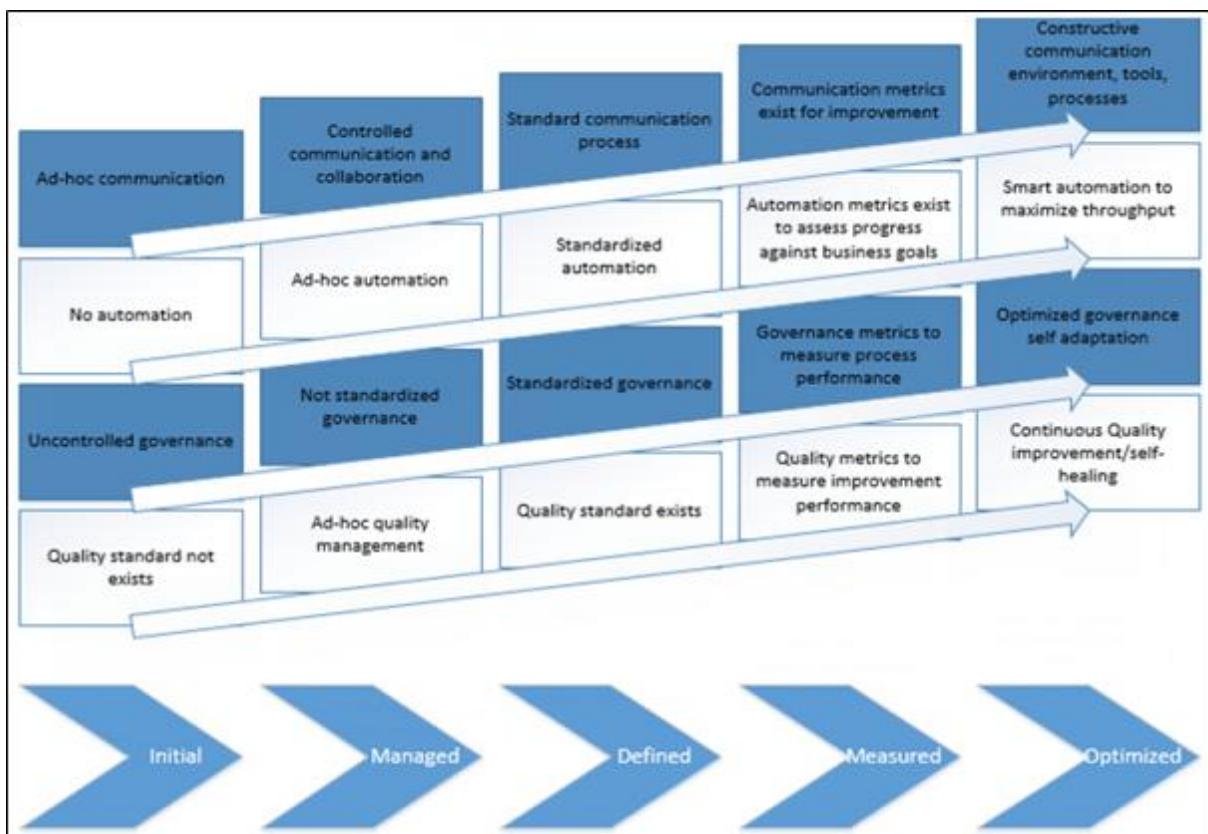
2.3 Specifika bedömningsmodeller för DevOps

I syfte att strukturera problemet, identifiera lösningsmöjligheter samt för att vägleda designen av artefakten har enskilda instanser av lösningar för specifika bedömningsmodeller för DevOps identifierats. Det finns idag ett flertal modeller vars syfte är att bedöma verksamheter ur ett DevOps kontext. Studien har identifierat två modeller eller verktyg som har skapats utifrån forskning inom ämnet: (i) DevOps maturity model (Mohamed 2015); och (ii) Improve (Cronholm 2018). Studien har även identifierat tre bedömningsmodeller som saknar en transparens i sin utformning då de istället för forskning baserats på praktikers erfarenheter, dessa modeller är: (iii) Solinea DevOps Maturity Model Matrix (Parks 2016); (iv) The

Continuous Delivery Maturity Model (Rehn, Palmberg, Boström 2013); och (v) The LeaseWeb Maturity model for DevOps teams (Poelwijk 2016). Även om modellerna har baserats på praktikers erfarenheter snarare än forskning, har de varit relevanta att granska då de är modeller som används och finns på marknaden.

2.3.1 DevOps maturity model

Det förekommer modeller som är grundade från litteratur, en sådan har skapats av Samer Ibrahim Mohamed (2015). Mohamed (2015) förespråkar att nyckeln till en lyckad implementering av DevOps och kontinuerliga leveranser är kvalitet, automatisering, samarbete och övervakning eftersom dessa element kan användas för att brygga samman Dev och Ops. Med kvalitet menar Mohamed (2015) att användning av lean ger snabbare leveranser till live-miljöer vilket underlättar omställningstiden och gör applikationen mer stabil med högre kvalitetsmål. Automatisering förbättrar leveranshastighet, produktivitet och återskapande. Samarbete säkerställer bättre kommunikation mellan olika team inom samma projekt. Med hjälp olika angreppssätt och verktyg som möjliggör kommunikation minskas risker, kostnad för omarbete, samtidigt som värdet för kunden maximeras. Övervakning kontrollerar hur dessa element samverkar sömlöst tillsammans för att uppnå DevOps. Grundat i CMMIs mognadsnivåer och ovan beskrivna element har Mohamed (2015) skapat en mognadsmodell för bedömning av DevOps, se figur 1.



Figur 1: DevOps maturity model (Mohammed 2015, s. 53)

2.3.2 Improve

På Högskolan i Borås pågår sedan 2016 ett forskningsprojekt i *datadriven innovation*. Projektet genomförs av forskningsgruppen *InnovationLab* som leds av Stefan Cronholm, professor i informatik. På högskolans hemsida beskrivs projektet enligt följande:

Projektet syftar till att stödja organisationer till att få förbättrad kunskap i datadriven innovation för att fullt ut kunna utnyttja de nya affärsmöjligheter som uppstår. Ett annat syfte är att hjälpa våra samarbetspartners att skapa konkurrensfördelar genom att integrera resurser som tillsammans kan skapa organisatoriska förmågor. Projektets övergripande forskningsfråga lyder: Hur kan sociotekniska resurser förbättra datadriven innovation i organisationer?

(Cronholm 2018)

Projektet består av tre mindre delprojekt där ett benämns som *Digital Platforms for Service Innovation*. I delprojektet har ett digitalt verktyg skapats kallat *Improve* med syfte att hjälpa verksamheter att förbättra sina processer ur ett tjänsteperspektiv. I verktyget har InnovationLab skapat ett antal påståenden (se bilaga 1) som ligger till grund för utvärderingen. Påståendena har formats utifrån tidigare forskning och utvecklats vidare i samarbete med praktiker. Totalt har tio organisationer varit involverade i utformning av påståendena som utvecklats i flera iterationer.

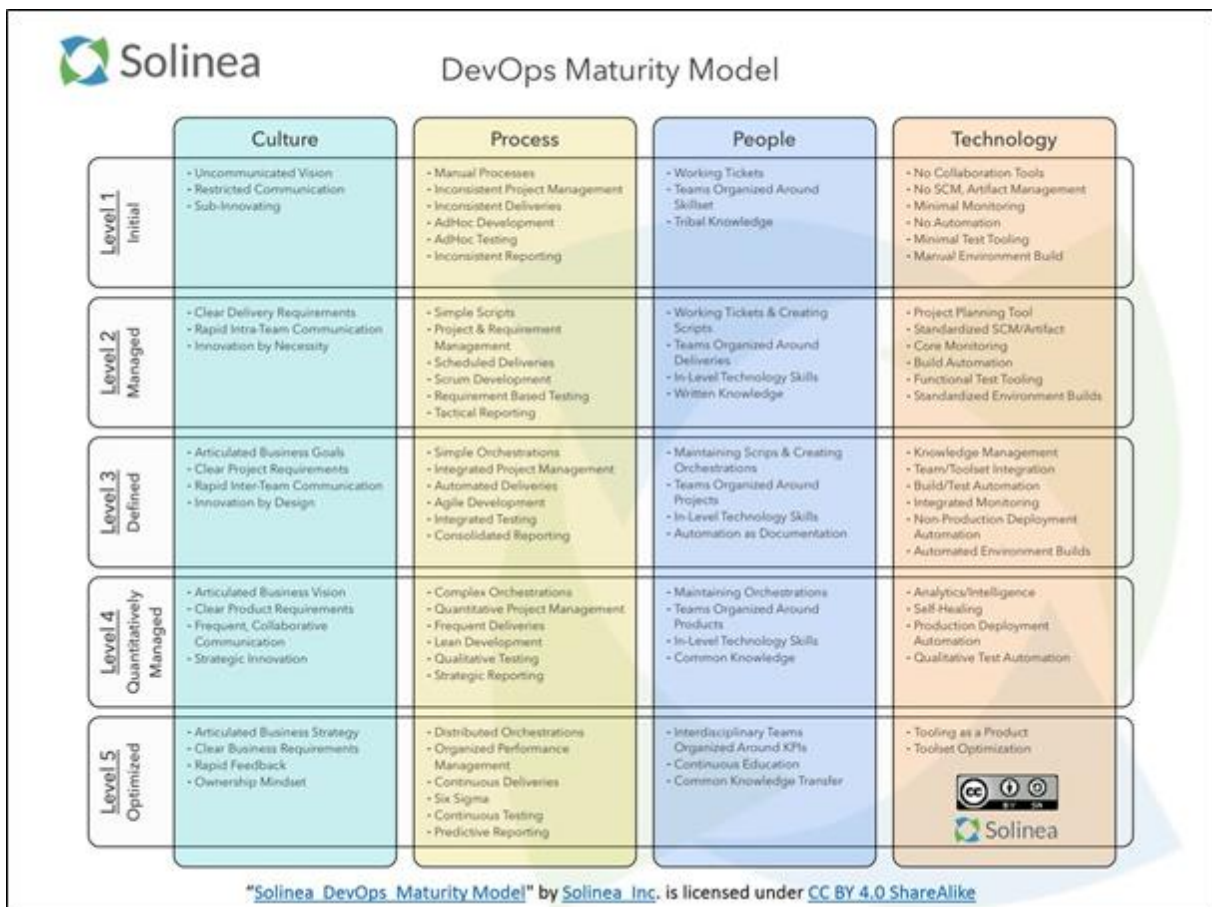
Improve används genom att verksamheter besvarar påståendena som kategoriserats på områdena: *samskapande av tjänst, resurser, automation, ständiga förbättringar* och *helhetsperspektiv*. Svaren befinner sig på en femgradig skala på noll till fyra där noll innebär att de inte alls håller med och fyra innebär att de instämmer helt. Dev och Ops svarar på påståendena separat för att sedan tillsammans genomföra en workshop där de båda avdelningarnas svar utvärderas för att identifiera eventuella gap, styrkor eller svagheter i syfte att förbättra tjänsterna. I Improve finns möjlighet att kommentera och dokumentera utvärderingen som även sparas. Nästa gång verksamheten genomför en utvärdering kan de titta på föregående sessioner för att se ifall de arbetar i rätt riktning och faktiskt förbättrar sina tjänster. Vid användning av Improve går det enbart att bedöma de olika påståendena enskilt, det saknas därmed ett ramverk eller en modell för att ge en helhetsbedömning när verksamheter använder verktyget.

2.3.3 Solinea DevOps Maturity Model Matrix

Konsultföretaget Solinea som under flera år har arbetat med att leda företag till en mer agil, säker och förändringsbar verksamhet uppmärksammade problematiken med att rama in definitionen för begreppet DevOps i verksamheter. De menar att ledningen (executives/managers) har svårt att inse hur DevOps påverkar verksamheter och att bristen på insyn kan hämma inställningen till förändring (Parks 2016). Baserat på tidigare erfarenheter i verksamhetsutveckling skapade de en modell som kan användas som en vägledning för att guida verksamheter i deras DevOps-adaption. Vid utformningen av denna modell identifieras fyra grundpelare som representerar aspekter i en verksamhetsförändring, dessa är: *människor (people)*, *teknologi (technology)* *processer (process)* samt *kultur (culture)* (Parks 2016). Solinea menar att en verksamhet måste minska sitt manuella arbete och anpassa sina processer för att främja automatisering (Parks 2016). För att möjliggöra automatisering måste människor i verksamheten ha kunskapen i att arbeta med automatisering på ett sätt som berör infrastruktur, automatiserad utveckling samt teknologier i verksamheten för att stötta dem i deras utveckling (Parks 2016). För att skapa dessa förändringar i verksamheten behöver verksamheten se över sin rådande verksamhetskultur (Parks 2016). De behöver förbättra sin

kommunikation för att på ett effektivt sätt redogöra för de hinder de har i sitt arbete (Parks 2016). Det finns tydliga beroenden mellan de identifierade aspekterna, där förändringar i en aspekt även påverkar övriga.

Historisk har ledningen för verksamheter använt sig av mognadsnivåer, vanligtvis CMMI, för att förändra sin verksamhet (Parks 2016). De menar att CMMI sätter verksamhetens förändringar i en kontext med mål att guida verksamheten mot process- och kvalitativa mål samt användas för bedömning och förändringsplanering (Parks 2016). Då CMMIs mognadsnivåer kan ses som ett gemensamt språk för ledningen i en verksamhet är Solinea ramverk uppbyggt utefter dessa nivåer för att vara mer begriplig när den implementeras i verksamheten (Parks 2016). I mognadsmodellen presenteras fem nivåer av mognad vilka kan ses i figur 2. För varje nivå förklaras påståenden som symboliserar mognad efter de fyra definierade grundpelarna. Nivå ett beskrivs som *initial (initial)*, nivå två *hanterad (managed)*, nivå tre *definierad (defined)*, nivå fyra *hanterad kvantitativt (quantitatively managed)* och nivå fem *optimerad (optimizing)*. Nivåerna består av påståenden inom aspekterna kultur, människor, processer och teknologi (Parks 2016).



Figur 2: Solinea DevOps Maturity Model (Parks 2016)

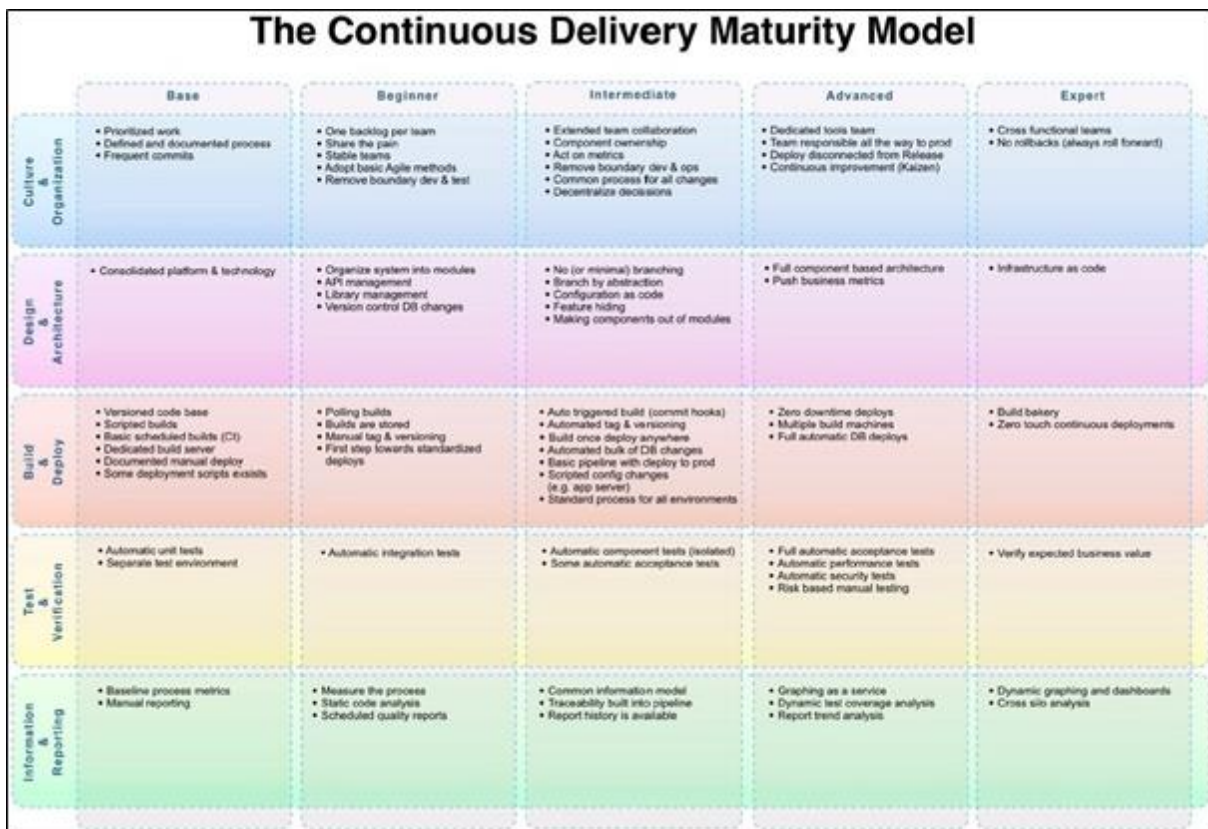
2.3.4 The Continuous Delivery Maturity Model

The Continuous Delivery Maturity Model används som ett verktyg vars syfte är att utvärdera processer vid utveckling och leverans av mjukvara ur ett helhetsperspektiv. Modellen har även som syfte att skapa en överblick i hur mogna verksamheter är i förhållande till modellens påståendesområden (Rehn, Palmborg, Boström 2013). Modellens mål är att ge struktur och förståelse för dessa påståendekområden vid implementering av kontinuerliga leveranser (*Continuous Delivery*) (Rehn, Palmborg, Boström 2013). Kontinuerliga leveranser inspireras av lean och agil systemutveckling med målet att snabbt kunna leverera och uppdatera programvara vilket går i linje med DevOps (Rehn, Palmborg, Boström 2013; Mohamed 2015). Agila metoder beskrivs som effektiva om de utvecklas inom verksamheten. Det kan dock bli problematiskt med agila metoder om delar av verksamheten inte är mogna nog för att anpassa sig till dessa metoder vilket kan hämma utvecklingen för resterande delar av verksamheten (Rehn, Palmborg, Boström 2013). För att undvika hämning av utveckling menar Rehn, Palmborg och Boström (2013) att verksamheter bör fastställa en solid plattform som skapar förutsättningar för verksamheten att utvecklas. Plattformen baseras på att verksamheten implementerar specifika verktyg, principer, metoder och praxis som grupperas i fem kategorier: *Culture & Organization*, *Design & Architecture*, *Build & Deploy*, *Test & Verification* och *Information & Reporting* (Rehn, Palmborg, Boström 2013). För att skapa en solid plattform som möjliggör förändring med hållbara resultat, anser Rehn, Palmborg och Boström (2013) att verksamheter behöver strukturera och anpassa sina kontinuerliga leveranser till deras mognadsmodell som är baserad på fem kategorier. Modellen hjälper även verksamheter att identifiera vilka inledande handlingar som ger mest effektiva resultat. Modellen har fått sin inspiration utifrån praktiken i form av tidigare implementeringsprojekt av kontinuerliga leveranser. Modellen grundar i litteratur som Jez Humble och David Farley's bok *Continuous Delivery* samt Eric Minick och Jeffrey Fredricks artikel *Enterprise Continuous Delivery Model* (Rehn, Palmborg, Boström 2013).

Modellen är uppbyggd på fem nivåer: *bas (base)*, *nybörjare (beginner)*, *mellanliggande (intermediate)*, *avancerad (advanced)* och *expert (expert)* (Rehn, Palmborg, Boström 2013). Dessa nivåer är baserade utifrån praktiken och beskriver hur verksamheter förhåller sig till kategorierna *Culture & Organization*, *Design & Architecture*, *Build & Deploy*, *Test & Verification* och *Information & Reporting* (Rehn, Palmborg, Boström 2013). Nivåerna är inte strikta och behöver därmed inte nödvändigtvis uppfyllas i följd utan bör snarare tolkas som ett utvärderings- och planeringsunderlag (Rehn, Palmborg, Boström 2013). Dock rekommenderas ett inkrementellt tillvägagångssätt då verksameters avdelningar kan (inom de olika kategorierna) bedömas till olika nivåer (Rehn, Palmborg, Boström 2013). Som tidigare nämnt är det ett problem om inte samtliga avdelningar är mogna för agila metoder då det kan hämma resterande delar av verksamhetens utveckling. Författarna till modellen menar att en jämn kunskapsnivå mellan organisationens avdelningar är av stor vikt för att bedriva utveckling samtidigt som större organisatoriska förändringar bör undvikas om kunskaps- och mognadsnivån varierar mellan avdelningar (Rehn, Palmborg, Boström 2013).

Modellens fem kategorier mäts genom de fem definierade nivåerna, se figur 3. Varje kategori har sin egen mognadsprocess men påverkar oftast övriga kategorier då kategorierna har en stark koppling (Rehn, Palmborg, Boström 2013). *Culture & Organisation* anses vara den kategori som påverkar en hållbar miljö för kontinuerliga leveranser mer än övriga kategorier. När de övriga kategorierna stiger i mognadsnivå kommer också kulturen och organisationen utvecklas (Rehn, Palmborg, Boström 2013). *Culture & Organisation* fokuserar på hur verksamheten dokumenterar sitt arbete för fortsatt utveckling och hur de främjar multipla team där kompetensen sträcker sig över hela systemutvecklingslivscykeln. *Design &*

Architecture fokuserar på hur de verktyg och tjänster verksamheten har samt påverkar möjligheten att anpassa sig till kontinuerliga leveranser. Om verksamhetens verktyg är utvecklat med principer som baseras på kontinuerliga leveranser kommer anpassningen till den nya kulturen ske mer effektivt (Rehn, Palmborg, Boström 2013). Build & Deploy berör automatisering och de verktyg som verksamheten har. Rehn (2013) menar att den högsta mognadsnivån av automatisering är när kontinuerliga leveranser kan ske fullt automatiserat. Likt Build & Deploy utvecklas Test & Verification genom verksamhetens användning av verktyg och automation (Rehn, Palmborg, Boström 2013). När en verksamhet blir mer automatiserad kan leveranser och ökad feedback av mätvärden ske mer frekvent och effektivt, det blir då naturligt att verifiera värdet för verksamhetens tjänster. Då mätning av processer möjliggör kontinuerligt utveckla sin verksamhet måste också informationen som flödar i verksamheten vara åtkomlig för intressenter. Information & Reporting fokuserar på att den information som företaget förfogar över måste vara åtkomlig för intressenter samt vara relevant och koncis för att kunna möjliggöra kontinuerliga leveranser (Rehn, Palmborg, Boström 2013).



Figur 3: The Continuous Delivery Maturity Model (Rehn, Palmborg, Boström 2013)

2.3.5 The LeaseWeb maturity model for DevOps teams

En annan modell har utvecklats av LeaseWeb som har valt att dela med sig av sin modell via webbhotellet Github. Modellen är utvecklad och anpassad till LeaseWebs verksamhet samt deras DevOps-team (Poelwijk 2016). De anser därmed att modellen är utformad efter deras processer. Exakt hur och på vilka grunder mognadsmodellen har utvecklats eller hur den skall användas beskrivs inte av LeaseWeb. Däremot hänvisar de i sin modell till The Kniberg Scrum checklist vilket är ett verktyg som används i utvecklandet av den agila utvecklingsmetoden Scrum i en verksamhet. I Knibergs Scrum checklist förklaras det att verktyget inte skall tolkas som regler utan snarare riktlinjer i arbetssättet (Kniberg 2015). Det är inte meningen att påtvinga ett team att göra dagliga Scrum-möten utan snarare göra dem medvetna om varför det är viktigt att kommunicera. Om samma tankesätt används för LeaseWebs mognadsmodell går det att göra tolkningen att kraven i modellen egentligen inte behöver uppnås utan snarare har som syfte att öka medvetenheten i ett DevOps-team.

Modellen består av fyra kategorier som LeaseWeb benämner som krav (*requirements*), se tabell 3. I varje kategori måste ett team ta sig igenom fyra nivåer, *initial level*, *basic level*, *intermediate level* och *target level* (Poelwijk 2016). Nivåerna går från en initial nivå där inga krav uppfylls för att tillslut uppnå en målnivå där LeaseWeb menar att ett det råder ett idealt tillstånd för DevOps-team (Poelwijk 2016). Kategorierna är uppdelade i Kultur & Människor (*Culture & People*), DevOps Agility (*DevOps Agility*), Affärsvärde (*Business Value*) samt Automatisering & Verktyg (*Automation & Tooling*). Varje kategori innefattar flera områden som har underliggande påståenden, dessa områden och påståenden varierar beroende på vilken kategori som behandlas. För att uppnå grundnivån måste ett DevOps-team behandla informationsdelning och förståelse för verksamhetens processer samt att de uppfyller tre av tio påståenden ur The Kniberg Scrum checklist. Teamet måste också arbeta tillsammans för att skapa en god miljö där fel registreras i ett lärande syfte och fastställa gemensamma mål för sitt arbete (Poelwijk 2016). När grundnivån är uppnådd kan team fokusera på att öka förståelsen för policys som hanterar arkitekturen i verksamheten samt skapa övervakning av sina processer. På modellens grundnivå behandlas informationsdelning. Den mellanliggande nivån ställer högre krav på informationsdelningen som innebär att team delar information inom hela verksamheten. På nivån ställs även krav på att team har förståelse ur ett större perspektiv för de huvudmål som deras arbete bidrar med, samt att de tar emot feedback från kunder för att utveckla nya funktioner. Ur ett Scrum-perspektiv skall också teamet uppnått 10 av 10 påståenden ur The Kniberg Scrum checklist för att uppnå nivån. Den ideala nivån för ett DevOps-team benämns som målnivån. När ett DevOps-team uppnår denna nivå behandlar de ett proaktivt arbete genom att ta lärdom eller lära ut sitt arbetssätt i verksamheten, de granskar också sin egen prestanda och kommer tillsammans fram till hur de kan förbättra den. Teamet för också en aktiv dialog med intressenter i framtagningen av produkter samt följer upp hur nöjda de är med den levererade produkten. Verksamheter på denna nivå kritiserar inte misslyckanden utan menar att de är nyttiga för verksamheten då de skapar en kunskapsbas som kan användas för att undvika framtida fel.

Tabell 3: The LeaseWebs maturity model for DevOps teams (Poelwijk 2016)

Basic Level	
Happiness	We measure & share team happiness each sprint
Failing Fast	We register failures, so we can use this as an opportunity to learn
	We no longer deploy in a sprint after 2 failed deployments (error budget)
Transparency	We proactively share information with the whole team
Shared Responsibility	We understand each other's concepts, concerns and problems (of different expertise's)
	Our team is able to fully develop, maintain and support our products, applications and infrastructure (also in terms of skills & resources)
	We have our own EOD shifts to respond when services are down out of business hours.
Focus	We focus on our team goal & product vision
Self Organization	We plan our activities together with the PO
	Our team understands the boundaries in terms of resources, cooperation with other teams and departments, decision making policy and information flow
	We know the team KPIs

2.3.6 Sammanfattande kritik till specifika modeller

Det går att anse att dagens befintliga modeller för bedömning av verksamheter ur ett DevOps-kontext inte är tillräckligt väl utformade. För att stryka ovanstående påstående kommer studien presentera en del av de brister som identifierats när befintliga bedömningsmodeller analyserats. Bristerna är följande: (i) modellerna saknar en teoretisk grund och transparens i hur de utformats, de (ii) har baserats på CMMIs mognadsnivåer, (iii) de har inte tillräckligt stark koppling till CALMS, och slutligen (iv) att de inte har implementerats i ett digitalt verktyg.

Det har observerats att enbart ett begränsat antal av dagens mognadsmodeller som används i verksamheter för utvärdering och bedömning, har utvecklats och utformats utifrån forskningsbaserade riktlinjer. En majoritet av modellerna har istället utvecklats baserat på forskares praktiska erfarenheter vilket innebär att många mognadsmodeller saknar en teoretisk grund (Albliwi, Antony, Arshed 2014). Även Shrestha et al (2014) påpekar att processbedömningar oftast saknar transparens i sin utformning, vilket gör att bedömningarna ses som en svart låda där rationaliteten bakom bedömningsaktiviteterna inte förklaras. Samtidig saknas det idag vetenskapligt förankrade modeller för utvärdering av DevOps med syfte att hjälpa organisationer implementera och utveckla DevOps på ett stegvis sätt (Rong, Zhang, Shao 2016 se Feijter, Vliet, Jagroep, Overbeek, Brinkkemper 2017; Shrestha et al 2014).

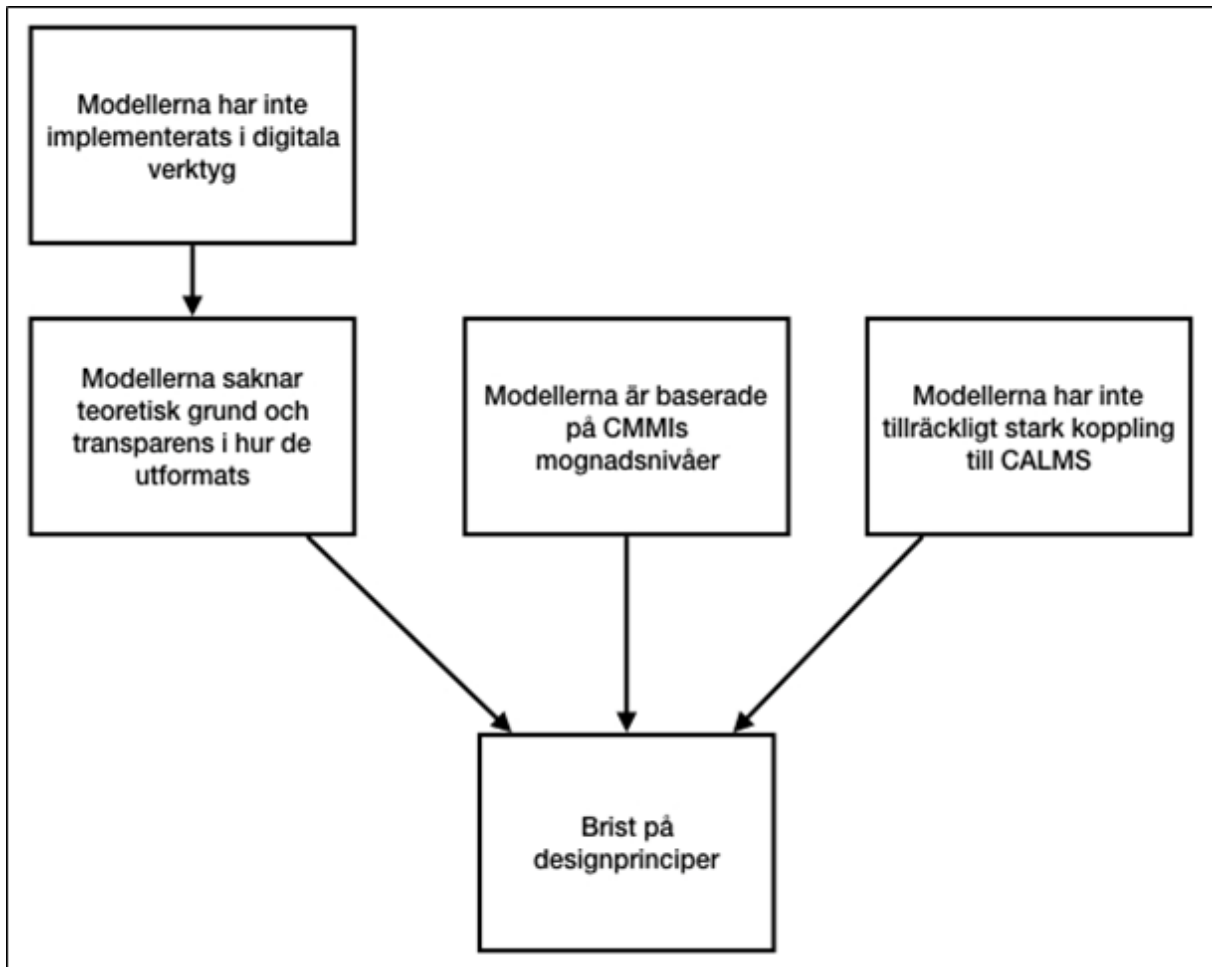
Samtliga modeller som analyserats har kategoriserat sina processområden på olika sätt. Mohameds modell (2015) är baserad på de fyra begreppen eller elementen kvalitet, automation, kommunikation/samarbete och övervakning. InnovationLabs processpåståenden

har kategoriserats på processområdena samskapande av tjänst, resurser, automation, ständiga förbättringar och helhetsperspektiv (Cronholm 2018). Trots att kategorierna skiljer sig till namn går det att urskilja att de till stor del förhåller sig till begreppen i CALMS som är en välkänd akronym och används av bland annat *The DevOps Institute* och *DevOps Agile Skills Association (DASA)* för att förklara sammanlänknigen av avdelningarna Dev och Ops i verksamheter (DevOps Institute 2017; DASA 2016). Mohammed (2015) beskriver elementet kvalitet enligt följande: kvalitet säkerställer snabbare och lean leveranser till autentiska miljöer vilket minskar omarbetstiden, gör applikationer mer stabila och höjer kvalitetsmålen på leveranser. Beskrivningen överensstämmer till stor del med definitionen över lean som är en del i CALMS (Hüttermann 2012). Samtidigt går det att koppla samtliga av InnovationLabs processpåståenden till CALMS (Cronholm 2018). De övriga modellerna som analyserats (The Continuous Delivery Maturity Model, The LeaseWeb maturity model for DevOps teams, Solinea DevOps Maturity Model Matrix) går också att koppla till CALMS även om de inte har lika stark koppling som i de tidigare modellerna.

Tidigare forskning visar att användning av mjukvara eller digitala verktyg som exempelvis affärsprocess-modelleringsverktyg kan påskynda implementering av eller förbättra processer (Shrestha et al 2014). Idag finns det en brist på verktyg för processbedömning vilket kan bero på att många bedömningsmodeller inte är transparenta i sitt utformade vilket gör det svårt att tolka modellerna (Shrestha et al 2014). Tidigare studier påvisar att de mjukvaruverktyg som finns idag främst utformats för praktiker som arbetar med processbedömning snarare än för verksamheter som vill ha möjlighet till självbedömning (Shrestha et al 2014).

2.4 Behov av digital bedömningsmodell för DevOps

Utifrån den sammanfattande kritiken till de generella och specifika bedömningsmodellerna går det att identifiera flera brister och problem som listats i en problemgraf vilken illustreras i figur 4. Problemgrafan visar inte bara vilka problem som identifierats utan visar även samband mellan problemen. Grundorsaken till alla problem som identifierats går att härleda till att det idag saknas designprinciper som utvecklare bör följa vid utformande av bedömningsmodeller för DevOps. Problemen visar även att det finns ett behov av utvecklandet av en digital IT-artefakt som kan användas av verksamheter med syfte att bedöma sina verksamheter ur ett DevOps-kontext.



Figur 4: Problemgraf

3 Metod

I kapitel tre motiveras och beskrivs den metod som valts för att besvara forskningsfrågan. Samtliga steg eller processer som genomförts vid utvecklandet av artefakten beskrivs utifrån metodens fyra steg för att ge en tydlig bild över tillvägagångssättet. Då utveckling av artefakten skett iterativt har kapitlet strukturerats efter de utvärderingsepisoder som genomförts. Kapitlet innefattar 3.1) Metodval och 3.2) Action Design Research.

3.1 Metodval

Det är forskningsfrågan som styr vilken forskningsmetod som bör väljas vid forskning (Johannesson & Perjons 2014). För att besvara vår frågeställning om hur en IT-artefakt som möjliggör bedömning av DevOps bör utformas, har vi till stor del inspirerats av forskningsmetoden *action design research (ADR)* som är en vidareutveckling av *design research (designforskning)*. Det som skiljer designforskning från traditionell empirisk forskning är att den inte bara har som syfte att beskriva, förklara och förutspå. Designforsknings syfte är även att förändra världen, förbättra den och även skapa nya världar. För att uppnå dessa mål används artefakter som kan hjälpa människor uppfylla sina behov, lösa problem och fånga möjligheter (Johannesson & Perjons 2014). Designforskning har sina rötter från ingenjörsvetenskap och är ett problemlösande paradig. I designforskning ses design både vara en process (samling av aktiviteter) som beskriver hur världen ser ut och en produkt (artefakt) som beskriver hur världen upplevs. Forskningen innebär kontinuerligt skiftande av perspektiv mellan designprocesser och designade artefakter för samma komplexa problem. En artefakt kan innebära flera olika saker: det kan vara en konstruktion (vokabulär och symboler), modell (abstraktioner och presentationer), metod (algoritmer och utföranden) eller instans (implementerade och prototypsystem) (Hevner et al 2004). I den här studien definieras artefakten som en modell som möjliggör bedömning av DevOps. Designprocessen är en sekvens av aktiviteter som producerar en innovativ artefakt. Utvärderingen av artefakten skapar en större förståelse av problemet vilket leder till att det går att förbättra både artefaktens kvalitet och designprocessen. Denna cykel genomgår oftast flera iterationer innan den slutliga artefakten är färdigställd. Under den här kreativa processen måste forskaren se utveckling av både designprocessen och artefakten som en del av forskningen (Hevner et al 2004). Kunskap som anskaffats för designprocessen kan användas för andra instanser av samma artefakttyp och benämns ibland som designprinciper (Sein, Henfridsson, Purao, Rossi, Lindgren 2011).

Inom designforskning är det oftast det påträffade problemet som driver utveckling av artefakter som senare utvärderas. När påträffat problem är den drivande faktorn för utveckling av artefakter så leder det till en cykel av "bygg först och utvärdera sen" (Sein et al 2011). Betoning inom designforskning ligger därmed på det ursprungliga problemet som identifierats i början av forskningsprojektet som gör att det blir obalans mellan kraven att (i) behandla problemen och att (ii) behandla dessa i en autentisk miljö. Separationen mellan utveckling av artefakter och utvärdering gör att det går att kritisera forskningen som otillräcklig vid skapande av artefakter. Även om designforskning har ett starkt stöd för abstraktion och innovation så anses den autentiska miljöns påverkan vara sekundärt (Sein et al 2011).

Det framgick att designforskning var en god grund för att besvara forskningsfrågan men att det samtidigt krävdes en kompletterande metod som går i linje med designforskning där det läggs större betoning på utvärdering av artefakten. För att identifiera och välja en sådan metod använde vi oss av Venable, Pries-Heje och Baskervilles (2017) artikel som redogör för hur

forskare kan välja forskningsmetod inom designforskning. Artikeln används för att hjälpa forskare välja mellan olika metoder inom designforskning beroende på vilket paradigm som forskaren förhåller sig till samt till vilken grad verksamheter involveras i forskningen. I artikeln delas tillgängliga metoder upp i två kategorier eller paradigm: positivistiska eller interpretativa metoder. För att besvara forskningsfrågan krävdes ett interpretativt förhållningssätt då problemformuleringen baserats på problem som identifierats i praktiken snarare än litteraturen vilket även kräver involvering av praktiker vid forskningsarbetet. Därmed uteslöts samtliga av de positivistiska metoderna i artikeln. För att välja mellan de interpretativa metoderna var nästa steg att granska huruvida verksamheter behöver involveras i forskningen. Artikeln föreslår att om det enbart finns en verksamhet som vill involveras i forskningen bör ADR väljas (Venable, Pries-Heje och Baskervilles 2017). Även om vi använt oss av mer än en verksamhet vid forskningen framgick det att det vara denna metod i artikeln som var bäst lämpad för vår forskning.

3.2 Action design research

För att besvara vår frågeställning har vi därmed utgått från traditionell designforskning men till stor del följt metoden action design research och dess steg och tillvägagångssätt som till skillnad från traditionell designforskning lägger större betoning på utvärderingen vid skapandet av IT-artefakter (Sein et al 2011). Metoden ger en uttrycklig vägledning för hur byggnad, inblandning och utvärdering bör genomföras i en samordnad forskningsansats. ADR hanterar två olika utmaningar, att (i) hantera problem som uppstår i en specifik verksamhet genom inblandning och utvärdering, samt att (ii) konstruera och utvärdera en IT-artefakt som kan hantera en samling problem baserat på den uppkomna situationen. Metoden består av fyra steg som i sin tur har ett antal principer. De fyra stegen är: problemformulering, “bygga, intervention och utvärdering”, “reflektion och lärande” samt “formalisering av lärande”. Principerna för problemformulering säkerställer att forskaren har som syfte att anskaffa kunskap som kan appliceras på en klass problem snarare än ett specifikt problem. Vid ADRs andra steg skapas en IT-artefakt som testas i verksamheten, efter varje test utvärderas artefakten som utvecklas iterativt. Utvärderingscyklarna i tidigare versioner är mer formativa för att i senare versioner bli mer summativa. Steg tre “reflektion och lärande” sker parallellt med de två första stegen och säkerställer att lösningar skall hantera en klass problem och lösningar snarare än enskilda instanser. Steget innebär även att forskaren kontinuerligt reflekterar över problemområdet, relaterad litteratur samt att det skapas ny kunskap. Det sista steget “formalisering av lärande” har som syfte att generalisera kunskapen som anskaffats. Sein et al (2011) förespråkar tre nivåer för denna process: (i) generalisering av problemet, (ii) generalisering av lösning, samt (iii) härledning till designprinciper utifrån forskningsresultatet.

Resultatet av designforskning är inte bara skapandet av innovativa artefakter, resultatet är även kunskap om hur det går att skapa andra instanser av artefakter som berör samma klass av problem, kunskapen kallas för designprinciper (Sein et al 2011). Designprinciper identifieras och formas i ADR under steget “reflektion och lärande” för att senare definitivt formuleras i det sista steget “formalisering av lärande”. Designprinciper är därmed baserade på de kunskaper som anskaffats för ett en viss klass av problemområde och går därmed att applicerade vid andra tillfällen då samma område berörs (Sein et al 2011). Syftet med designprinciper beskrivs av Gregor och Hevner (2013) som möjligheten att kommunicera designkunskap om hur IT-artefakter bör utformas för att hantera en klass av problem snarare än enskilda instanser. Vid utformning av designprinciper har vi utgått från van Akens (2004) struktur för designprinciper som ser ut enligt följande: *if you want to achieve Y in situation Z, then something like action X will help*. “Something like action X” betyder att åtgärden skall

användas som ett designexemplar. Ett designexemplar är en generell åtgärd som måste översättas till en specifik åtgärd för att lösa ett specifikt problem.

3.2.1 Problemformulering

Forskningsfrågan har grundats i att det idag finns ett stort behov för organisationer att arbeta med ständiga förbättringar av processer för att hålla sig konkurrenskraftiga på marknaden (CMMI Team 2010; Shrestha et al 2014). Likt arbetet med processförbättringar behöver även samarbete mellan Dev och Ops i verksamheter utvärderas och bedömas i syfte att förbättras (Riley 2015). Forskningsproblemet som motiverar vår forskning är att de modeller som finns idag med syfte att utvärdera verksamheters DevOps-satsning är bristfälliga då de: (i) saknar en teoretisk grund och transparens i hur de utformats, de har (ii) baserats på CMMIs mognadsnivåer, de har (iii) inte tillräckligt stark koppling till CALMS och slutligen att de (iv) inte har implementerats i ett digitalt verktyg. Ovanstående problem kan tillsammans leda till att bedömning av verksamheter genomförs på ett bristfälligt sätt. I enlighet med ADR skapades den initiala forskningsfrågan baserat på forskningsproblemet enligt följande: *Hur bör en digital artefakt som möjliggör bedömning av DevOps utformas?* Frågan är formulerad på följande sätt då den gick att generaliseras till en klass, snarare än en enskild instans av ett problem vilket är ADRs syfte (Sein et al 2011).

När forskningsfrågan hade utformats granskades relaterad litteratur då detta enligt ADR måste göras för att strukturera problemet, identifiera potentiella lösningar samt vägleda designen av artefakten (Sein et al 2011). Vi identifierade dels generella bedömningsmodeller vilka kan ses som klasser av modeller för utvärdering av processer, men även specifika instanser av bedömningsmodeller för utvärdering av DevOps. Modellerna granskades vilket resulterade i en problemgraf som blev underlag för de mål som sattes för artefakten. Målen har använts kontinuerligt för att vägleda designen och utvärdera artefakten.

En motivering till att använda sig av ADR som forskningsmetod var att forskningsfrågan krävde utvärdering av artefakten i verkliga miljöer och verksamheter. Enligt metoden måste därför verksamheters involveras tidigt i processen i syfte att säkerställa deras åtagande (Sein et al 2011). Vi kontaktade därför flera företag verksamma inom IT-sektorn som tidigare visat intresse för DevOps då de bland annat redan valt att delta i InnovationLabs forskningsprojekt inom datadriven innovation. Vi frågade verksamheterna om de hade möjlighet att delta vid en utvärderingsepisod där artefakten skulle testas i deras miljö och utvärderas. Två IT-organisationer av olika storlek och kunskap valde att delta i projektet och kommer att härnäst benämnas som Organisation A och Organisation B. Artefakten utvärderades även tillsammans med en praktiker som dels har goda kunskaper inom DevOps och även deltagit i InnovationLabs forskningsprojekt.

3.2.2 Bygga, intervention och utvärdering

ADR's andra steg sker i flera iterationer med syfte att utveckla en IT-artefakt som skall lösa det problem som initialt definierats i problemformuleringen (Sein et al 2011). Under steget utvärderas artefakten ständigt samtidigt som designprinciper växer fram som kan hantera en klass av samma problem. Samtliga delar i detta steg anses viktiga och ingen kan förbises vid utvecklande av artefakter.

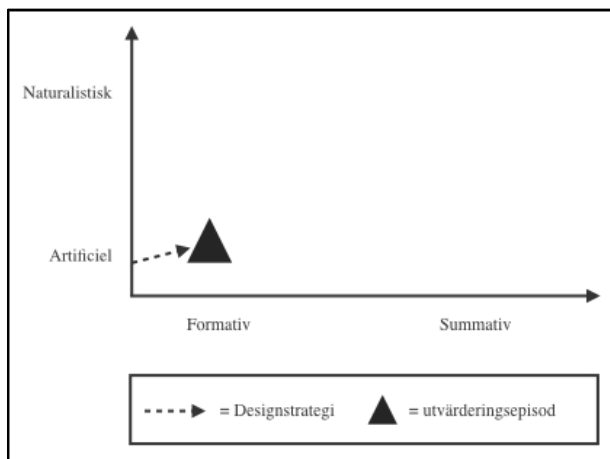
Den initiala designen av vår IT-artefakt har baserats på forskningsfrågan samt relaterad litteratur. De initiala påståenden som användes i modellen har hämtats från forskningsgruppen InnovationLabs delprojekt Digital Platforms for Service Innovation (Cronholm 2018). För att göra modellen digital lånades även verktyget Improve som skapats av InnovationLab.

Kapacitetsnivåerna som sattes för att bedöma verksamheterna har utformats baserat på CMMI och ISO/IEC15504s modeller för processbedömning. CMMIs och ISO/IEC 15504s modeller skiljer sig i antalet kapacitetsnivåer där CMMI har tre nivåer medan ISO-standarderna har fem (CMMI Team 2010; Adalı, Özcan-Top, Demirörs 2016). Båda modellerna har dock liknande beskrivning för vad som behöver uppnås på varje process totalt för att uppnå den högsta kapacitetsnivån i respektive modell. Vi valde att utgå från CMMIs tre nivåer då vi ansåg att det blir lättare för användare att förstå modellen om färre nivåer används då komplexiteten minskar, samtidigt som vi ansåg att det även passade bättre ut ett DevOps-kontext att använda sig av färre kapacitetsnivåer då det kan vara svårt att motivera varför en typ av processförbättring skall tillhöra nivå fyra respektive fem. Även om artefakten utformades utifrån CMMIs modell gällande antalet kapacitetsnivåer ansåg vi att den även gick i linje med ISO-standarderna gällande utvärdering av processer. Vid utformning av bedömningsmodellen identifierades även fyra huvudmål för att möjliggöra utvärdering och för att kunna avgöra när modellen är klar, se sektion 4.1. De mål som sattes var att (i) modellen skall vara enkel att använda och förstå, (ii) modellen skall följa en internationell standard, (iii) modellen skall gå att digitalisera, och slutligen (iv) modellen skall vara utformad efter akronymen CALMS.

Artefakten har byggts i fyra iterationer där samtliga iterationer innefattar en utvärderingsepisod (UE). Vid utvärdering av artefakten har vi följt FEDS-ramverket som skapats av Venable, Pries-Heje och Baskerville (2016) som ger en vägledning till hur forskare bör utforma utvärderingsaktiviteter vid designforskning. Ramverket utgår från två dimensioner: (i) det funktionella syftet med utvärderingen (varför), och (ii) utvärderingens paradigm (hur). Det funktionella syftet med utvärderingen delas in i formativ och summativ utvärdering. Formativ utvärdering har syftet att hjälpa till att förbättra det som utvärderas medan summativ har som syfte att bedöma huruvida utfallet matchar de förväntningar som ställs. Utvärderingens paradigm delas in i artificiell och naturalistisk utvärdering. Artificiell utvärdering innebär att artefakten testas mot teori genom exempelvis simulering, experiment, teoretiska argument eller matematiska bevis medan naturalistisk utvärdering innebär att artefakten testas i en autentisk miljö. Exempel på naturalistiska metoder kan vara fallstudie, fältstudie, undersökning eller liknande (Venable, Pries-Heje och Baskerville 2016).

Vid utveckling av IT-artefakter kan forskare välja att följa olika utvärderingsvägar beroende på vilka behov eller resurser som finns tillgängliga. Pilarna i figur 5 visar dessa utvärderingsvägar eller strategier, trianglarna symboliserar utvärderingsepisoder som genomförs under utvecklingsprocessen. Gemensamt för samtliga strategier är att de initialt har formativa utvärderingar för att senare i projektet bli mer summativa (Venable, Pries-Heje och Baskerville 2016). Vi har valt att utgå från en blandning av de strategier som författarna kallar *The human risk & effectiveness evaluation strategy* (HRE) och *the quick and simple strategy* (QAS). HRE inriktar sig initialt på formativa och artificiella utvärderingar för att snabbt utvecklas till mer naturalistiska och formativa. Utvärderingsepisoderna blir mer summativa i slutet av utvecklingsprocessen och fokuserar mer på hur effektiviteten av artefakten samt att den skall gå att implementera i organisationer. QAS förespråkar en snabb utvecklingsprocess med färre utvärderingsepisoder, strategin rör sig snabbt mot naturalistiska och summativa utvärderingar vilket gör det till en billigare och snabbare strategi i jämförelse med de andra (Venable, Pries-Heje, Baskerville 2016). För att visa vald utvärderingsstrategi visas figurer (baserad på FEDS ramverk för utvärdering i designforskning) vid varje UE. Figuren beskriver hur utvärderingen förhåller sig till det funktionella syftet med utvärderingen samt utvärderingens paradigm.

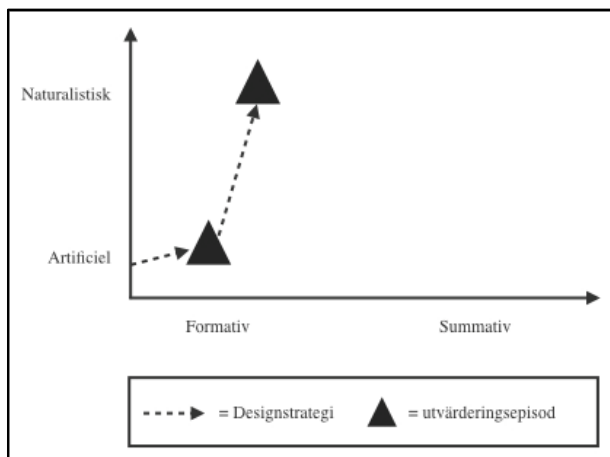
3.2.2.1 Utvärderingsepisod 1



Figur 5: Utvärderingsepisod 1

Figur 5 visar att UE1 som genomfördes var mer artificiell och formativ i sin utformning vilket stöds av Sein et al (2011) syn på hur tidiga utvärderingar av artefakter bör vara utformade. Episoden involverade inga potentiella användare, utan har genomförts av oss själva för att identifiera eventuella brister modellen. Då modellen hade skapats utifrån InnovationLabs påståenden för processer behövde vi initialt fastslå huruvida dessa påståenden var lämpliga att använda sig av i utformning av modellen, det vill säga ifall de är begripliga, ifall de gick i linje med CALMS samt ifall de gick att mappa mot CMMIs kapacitetsnivåer. Modellen utvärderades även utifrån de mål som identifierats som grundläggande kriterier för artefakten med undantag från målet att *modellen skall vara enkel att använda och förstå* då målet inte går att utvärdera på artificiell nivå.

3.2.2.2 Utvärderingsepisod 2

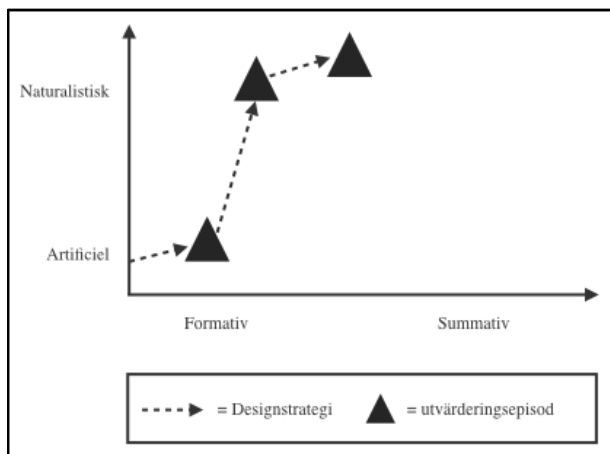


Figur 6: Utvärderingsepisod 2

UE2 som genomfördes var mer naturalistiskt utformad i jämförelse med UE1 då artefakten testades ute i en organisation. Frågorna som ställdes var fortfarande av formativ karaktär då även denna episod skedde i en tidig del av utvecklingsprocessen. Syftet var att utvärdera huruvida bedömningsmodellen ansågs lämplig att använda i organisationens verksamhet och att identifiera eventuella förbättringsmöjligheter. Figur 6 visar var UE2 befinner sig enligt FEDS-ramverk för utvärdering i designforskning.

UE 2 genomfördes ihop med organisation A där tre respondenter samlats för att utvärdera bedömningsmodellen. Inledningsvis fick respondenterna en beskrivning över hur utvärderingsepisoden var tänkt att genomföras. De fick information om att enbart ett påståendebeskrivande område i verktyget Improve skulle testas genom att Dev-representanter fick sätta sig separat och svara på de påståenden som finns i verktyget, samtidigt skulle Ops göra samma sak och svara på samma påståenden. Sen var tanken att genomföra en gemensam utvärdering av eventuella gap mellan Dev och Ops som identifierats baserat på de svar som framkommit i det tidigare skedet. För att starta igång grupperna gav facilitatorn en mer ingående beskrivning av verktyget, dess syfte samt en förklaring hur respondenterna gör för att svara på påståendena. Vid genomgången framgick det att respondenterna till stor del arbetade på Ops-sidan och saknade ett projekt där de representerat både Dev- och Ops-sidan vilket gjorde att utvärderingsepisoden behövde göras om och de fick istället svara på frågor kring de påståenden som finns i Improve-verktyget. Efter respondenterna fått uttrycka sig kring påståendena fick de avslutningsvis besvara på ytterligare frågor kring de kapacitetsnivåer som omfattar påståendena.

3.2.2.3 Utvärderingsepisod 3

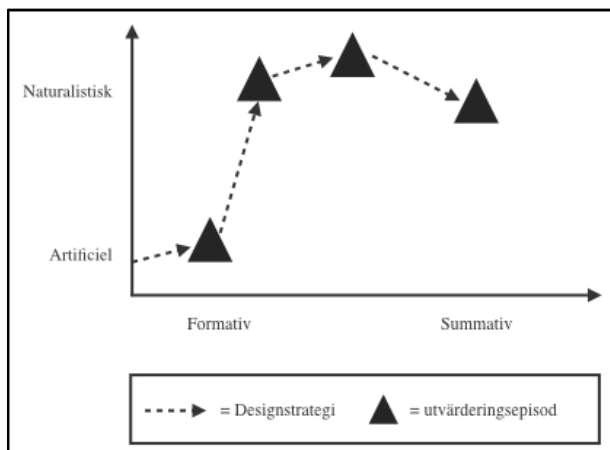


Figur 7: Utvärderingsepisod 3

UE3 var precis som UE2 av mer naturalistiskt karaktär då artefakten även denna gång testades i samarbete med en organisation. Frågorna som ställdes var fortfarande av formativ karaktär då även denna episod skedde i en tidig del av utvecklingsprocessen. Figur 7 visar att denna utvärderingsepisod är snarlik UE2 i sin utformning men då mer information har införskaffats blir utvärderingen mer summativ.

Den tredje utvärderingsepisoden genomfördes tillsammans med en respondent från organisation C. Då respondenten hade breda kunskaper inom DevOps både ur ett Dev- och ett Ops-perspektiv, samt erfarenheter i verksamheter ur ett ledningsperspektiv användes episoden för att granska kopplingar mellan de utformade DevOps-påståenden och dess kapacitetsnivåer i bedömningsmodellen. Respondentens breda kunskaper gjorde det särskilt lämpligt att granska huruvida modellens påståenden var väl utformade för att bedöma en verksamhet eller om förändringar behövde genomföras. Inledningsvis fick respondenten förklara sin tolkning av DevOps ur ett verksamhetsperspektiv. Följdfrågor till tolkningen ställdes där respondenten fick beskriva sin syn och erfarenheter på bedömningsmodeller och hur dessa bör användas i verksamheter. Respondenten introducerades sen för vår bedömningsmodell i det digitala verktyget Improve. Respondenten fick efter en kort genomgång granska påståenden och kapacitetsnivåer för att senare fritt kommentera dessa vilket ledde till många diskussioner som var relevanta och användbara för modellens utveckling.

3.2.2.4 Utvärderingsepisod 4



Figur 8: Utvärderingsepisod 4

Enligt FEDS ramverk bör utvärderingsepisoder av artefakter i slutet av utvecklingsprocessen vara summativt utformade i sin karaktär, det vill säga att den empiriskt samlade kunskapen används för att utvärdera huruvida artefakten kan uppfylla de mål som satts snarare än att identifiera förändringsmöjligheter för att forma artefaktens karaktär eller prestanda. Figur 8 visar var UE4 befinner sig enligt FEDS ramverk för utvärdering. Då episoden genomförts tillsammans med en respondent som har goda kunskaper i ämnet, både ur ett verksamhetsmen även ur ett forskningsperspektiv har episoden rört sig mot den artificiella axeln.

Den sista utvärderingsepisoden genomfördes tillsammans med en respondent med goda kunskaper inom IT-service management och DevOps. Respondenten har dessutom varit delaktig i utformandet av de processpåståenden som skapats av InnovationLab i deras projekt, datadriven innovation, samt varit delaktig vid utveckling av verktyget Improve som artefakten implementerats i. Även om respondenten har akademiska kunskaper inom ämnet och varit delaktig i InnovationLab, representerar ändå respondenten verksamheter vid utvärderingsepisoden, det vill säga att respondenten även har utgått från sina praktiska erfarenheter vid utvärdering av artefakten.

Respondenten fick en genomgång där vi beskrev hur vi använt oss av Improve för att implementera vår modell. De utformade påståendena förklarades då de förändrats kontinuerligt under utvecklingsprocessen och därmed skilde sig från de ursprungliga påståendena som respondenten själv varit med och utvecklat. Vidare beskrevs de kapacitetsnivåer som utformats samt hur vi anser att dessa matchar CMMIs modell. Respondenten fick därefter ge kritik och uttrycka sig fritt kring vår utformning av artefakten. Artefakten utvärderades även i denna episod mot de uppsatta målen som utformats.

3.2.3 Reflektion och lärande

Reflektions- och inlärningsfasen rör sig konceptuellt från att bygga en lösning för en specifik instans av ett problem till att tillämpa lärandet för en bredare klass av problem. Fasen sker kontinuerligt och parallellt med metodens tidigare steg och påvisar att forskningsprocessen involverar mer än att bara lösa ett problem. Reflektions- och inlärningsfasen säkerställer även att bidrag till kunskap anskaffas då de sker bevislig reflektion över de tidigare stegen: problemformulering och “bygga, intervention och utvärdering” (Sein et al 2011). Det är därmed av stor vikt att forskningsprocessen baserats på de reflektioner och den kunskap som införskaffats vid utveckling av artefakten

Vid genomförandet av samtliga utvärderingsepisoder har vi reflekterat över hur vårt problem samt lösning kan generaliseras. Vi har i våra reflektioner utgått från att skapa en klass av problem, snarare än en enskild instans av problem med syfte att bidra till vidare forskning inom området. De organisationer, människor samt den litteratur som har granskats i utvärderingarna har varit avgörande för utvecklingen av artefakten och bidragit till kontinuerliga förändringar av dess design. Vid de utvärderingsepisoder som genomfördes gavs förändringsförslag av organisationer i syfte att förbättra artefakten. De förändringar som påpekades av organisationerna ledde till reflektion där deras motivering utvärderats, reflektionen innebar även att vi försökte hitta stöd för deras påståenden i litteraturen. Utvärderingsepisoderna har bidragit till att artefakten har utvecklats snabbt och iterativt. Den iterativa processen har även gjort det möjligt för oss att identifiera och utveckla våra designprinciper under hela processen. Genom att kontinuerligt och noggrant reflektera vid varje utvärderingsepisod har vår frågeställning och våra designprinciper växt fram vilket går i linje med ADR.

3.2.4 Formalisering av lärande

Det fjärde och slutgiltiga steget i ADR är att formalisera den kunskap som anskaffats. För att genomföra denna formalisering behöver forskaren generalisera problem, lösning samt härleda forskningsresultatet i designprinciper. Ursprungsproblemet som initierade forskningen skall därmed ses som en enskild instans av en klass av problem. Lösningen på samma instans av problem behöver också generaliseras till en klass av lösningar. Slutligen måste även lärdomen som anskaffats vid forskningen för den specifika instansen omformuleras som generella designprinciper för en klass av lösningar (Sein et al 2011). Designprinciperna som identifieras och utvärderas under reflektion och lärande fasen tar därmed sin slutgiltiga form i denna metods sista fas.

Då vår artefakt har utvecklats under flera iterationer har även lärande och kunskap ökat iterativt längs med artefaktens utveckling. Endast när tillräcklig kunskap införskaffats ute i verksamheter och litteratur började designprinciper gestaltas för den enskilda instansen av problem: hur bedömningsmodeller för DevOps bör utformas. Den kunskap som anskaffats under utveckling av artefakten har slutligen kunnat generaliseras vilket har resulterat i tre designprinciper som presenteras vidare i kapitel fyra.

4 Resultatanalys

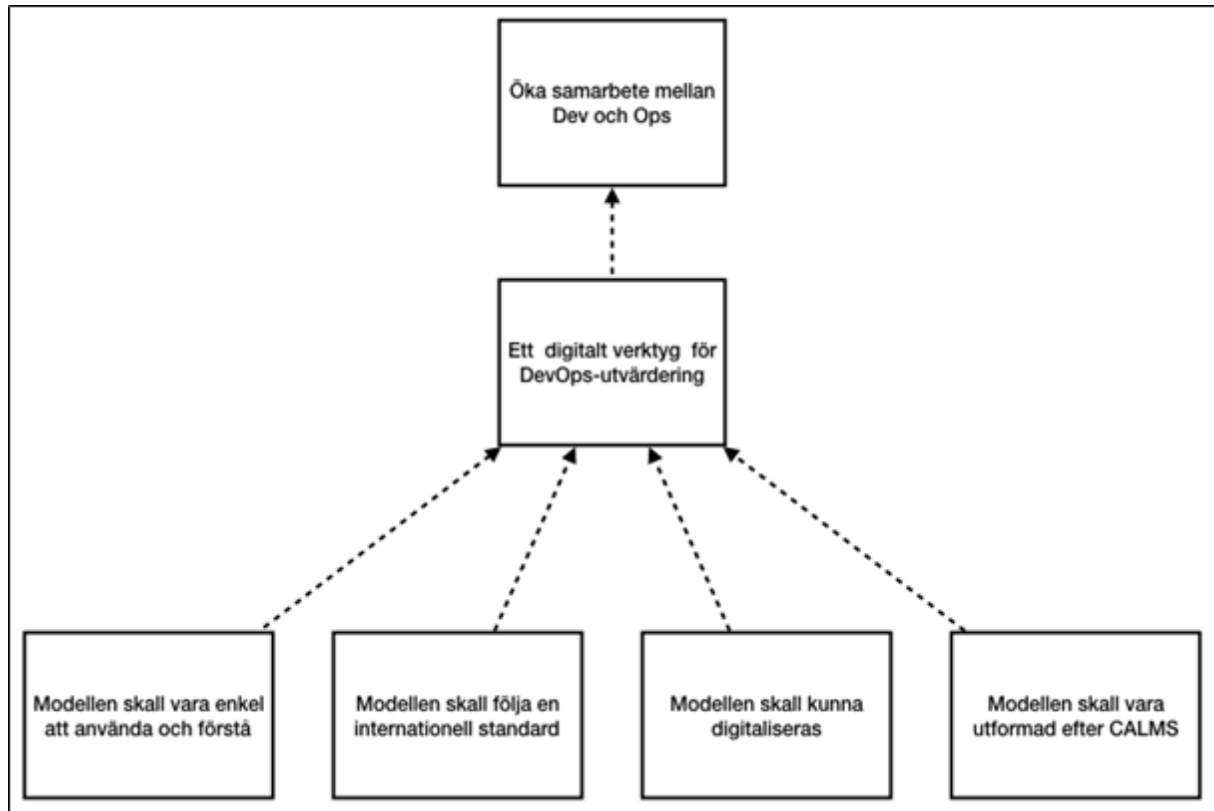
Resultatanalysen presenterar de mål som skapats för artefakten samt hur dessa mål påverkat utformning och utvärdering av IT-artefakten. Här presenteras också de designprinciper som identifierats genom införskaffade kunskaper. Vidare utvärderas och sammanfattas resultatet från den iterativa utvecklingsprocessen för IT-artefakten och designprinciperna. Kapitlet innefattar 4.1) Mål med IT-artefakt, 4.2) Beskrivning av IT-artefakt, 4.3) Beskrivning av designprinciper och 4.4) Utvärdering av IT-artefakt och designprinciper.

4.1 Mål med IT-artefakt

För att möjliggöra utvärdering av artefakten har fyra artefakt-mål skapats. Målen baserades på de behov som identifierades vid granskning av de problem som följs med dagens befintliga bedömningsmodeller. Målen för artefakten var att (i) modellen skulle vara enkel att använda och förstå, (ii) följa en internationell standard, (iii) kunna digitaliseras samt (iv) vara utformad efter CALMS. För att ge en tydligare bild över samtliga mål har de presenterats i en målgraf, se figur 9.

Det första målet att modellen skulle vara enkel att använda och förstå utformades då befintliga modeller som analyserats upplevdes som otydliga och svårbegripliga. En viktig del med detta mål var att särskilja verktyget (Improve) och artefakten (modellen) eftersom uppsatsen har avgränsats från att genomföra en fullständig studie i huruvida modellen är användarvänlig. Enkelheten och tydligheten var därmed mer inriktad kring påståenden och kapacitetsnivåer i modellen snarare än gränssnittet av verktyget som den implementerats i. Målet gick även i linje med nästa mål som innebär att modellen skall följa en internationell standard. Då CMMI och ISO/IEC 15504 är välkända modeller för processutvärdering kan verksamheter som redan är bekanta med dessa modeller och dess struktur lättare använda och förstå modellen (Adalı, Özcan-Top, Demirörs 2016). Målet bidrar även till modellen har en transparens i sin utformning vilket saknas i många av dagens befintliga bedömningsmodeller. Målet att modellen skall gå att digitalisera utformades då tidigare forskning visar att digitala verktyg kan påskynda implementering eller förbättring av processer (Shrestha et al 2014). Shrestha et al (2014) påstår även att det idag finns en brist på verktyg för processbedömning, vilket kan bero på att många bedömningsmodeller inte är transparenta i sitt utformade, samtidigt som existerande modeller främst utformats för praktiker som arbetar med processbedömning snarare än för verksamheter som vill ha möjlighet till självbedömning. Ovanstående problem påvisade att det finns ett behov att modeller bör gå att digitalisera vilket därmed blev ett mål för modellen. Det finns idag många olika tolkningar av DevOps vilket kan vara problematiskt när modeller skall tolkas och användas. Vissa tolkningar är dock mer etablerade än andra. För att uppnå maximal förståelse och användningsmöjligheter för modellen valdes den definition eller tolkning av DevOps som ansågs som mest etablerad i litteraturen och av praktiker nämligen CALMS. För att kunna utvärdera att modellen förhåller sig till akronymen skapades det sista målet att modellen skall vara utformad efter CALMS.

Målen har använts vid samtliga utvärderingsepisoder, dels för att säkerställa att rätt frågor ställdes men även för att avgöra när artefakten kunde anses som klar. För att artefakten skulle anses som klar behövde samtliga ovan beskrivna mål vara uppfyllda. Förhoppningen var att artefakten därmed kunde anses som ett färdigt digitalt verktyg för DevOps-utvärdering med syfte att öka samarbetet mellan Dev och Ops.



Figur 9: Målgraf

4.2 Beskrivning av IT-artefakt

För att besvara forskningsfrågan har en digital bedömningsmodell för DevOps utvecklats. Modellen har utformats efter de mål som presenterats, krav från praktiker och litteratur. Modellen har implementerats i verktyget Improve som är ett digitalt verktyg utvecklat av forskningsgruppen InnovationLab vid Högskolan i Borås. Verktyget gör det möjligt för verksamheter att själva utföra bedömning av sina processer ur ett DevOps-kontext. Vid bedömning kan verksamheter uppnå fyra olika kapacitetsnivåer som har baserats på litteratur för processutvärdering (CMMI och ISO/IEC15504s modeller). Kapacitetsnivåerna har som syfte att hjälpa verksamheter genomföra nulägesanalyser ur ett DevOps-kontext och därmed identifiera förbättringsmöjligheter. Kapacitetsnivåerna i modellen uppfylls när samtliga processpåståenden för respektive nivå har uppnåtts enligt både Dev och Ops. De processpåståenden som används i modellen har initialt utvecklats av InnovationLab och specifikt forskningsprojektet datadriven innovation. Påståenden har senare vidareutvecklats i de genomförda utvärderingsepisoderna (se sektion 4.4.4).

Vid utvärderingen används verktyget Improve av representanter från Dev och Ops som separat besvarar ett antal påståenden som kategoriserats i olika påståendekområden: *samskapande av tjänst, resurser, automation, ständiga förbättringar* och *helhetsperspektiv*. Modellens påstående går i linje med CALMS som är en välkänd akronym som används för att beskriva hur verksamheter bör arbeta för att brygga samman avdelningarna Dev och Ops (Kavis 2014). Efter att påståendena har besvarats skall representanterna för de båda avdelningarna samlas för att tillsammans utvärdera resultatet. Vid den gemensamma utvärderingen presenteras de svar som getts från båda avdelningarna som åter får motivera och diskutera sina val. Utvärderingen leder till att verksamheten direkt kan identifiera brister om båda avdelningarna svarat att de inte uppnår vissa påståenden. Utvärderingen kan även identifiera eventuella gap mellan avdelningarna som kan ge olika svar på samma påståenden. För att uppnå en viss kapacitetsnivå på ett visst område måste både Dev och Ops svara ja på samtliga påståenden som är kopplade till den aktuella nivån. Påståenden som är kopplade till tidigare nivåer är av mer operativ karaktär för att på högre nivåer blir mer strategiska i sin utformning. Då artefakten inte följer CMMIs traditionella mognadsmodell kan verksamheter klättra flera kapacitetsnivåer på samma område oberoende av de andra processområdena i modellen.

Vid utvärderingen av artefakten har det även framkommit att en facilitator bör leda utvärderingen ute i verksamheter för att: styra den i rätt riktning, säkerställa att samtliga får sin röst hörd, ställa utmanande frågor samt för att anpassa och modifiera påståenden i modellen för att bättre matcha verksamheten vid behov. En sådan facilitator behöver nödvändigtvis inte ha goda insikter i den verksamhet som utreds utan behöver snarare vara införstådd i DevOps samt ledning av workshops. Modellen kan användas utan en sådan facilitator men det finns en risk att utvärderingen då inte uppnår samma effekt, speciellt om utvärdering inte genomförs av Dev och Ops tillsammans.

Processes Assessment - Operations processes

Helhetsperspektiv | Samskapande av tjänst | Resurser | Automation | Ständiga förbättring

A set of proposition for the selected process can be found below. Rate each proposition based on the current situation. Use the column comments to motivate and specify the rating.

Assessment Date: 2018-04-16 | Participants: Anna Forest - Operation assets

Statement	Assessment	Comment
Level 1		
1. Dev och Ops har synliga processer som följs av alla berörda medarbetare.	Ja	
2. Dev och Ops har förståelse för varandras verksamheter.	Ja	
3. Dev och Ops utvecklar och tillvaratar medarbetarnas kompetens.	Ja	
4. Dev och Ops har kunskap om hur deras delar i processen bidrar till värde för kund.	Nej	
5. Dev och Ops förstår hur interna och externa underleverantörens processer bidrar till kundvärde.	Nej	
6. Dev och Ops förstår hur deras egna processer påverkar interna och externa underleverantörernas processer.	Ja	
7. Dev och Ops förstår hur deras egna processer påverkar interna och externa underleverantörernas processer.	Select Nej Ja	
8. Det finns processägare som bevakar och styr våra processer.	Nej	
9. Dev och Ops tar tillvara sina resurser (människor, teknik och ekonomi) på ett optimalt sätt.	Nej	
10. Dev och Ops uppmuntrar beteenden som syftar till en DevOps-inriktad kulturförändring.	Ja	
11. Dev och Ops försöker tillsammans skapa en kultur som stödjer utveckling och förändring.	Ja	
Level 2		
12. Det finns en ömsesidig bevakning av att aktiviteter i området Helhetsperspektiv* genomförs.	Nej	
13. Personal utbildas vid behov.	Ja	
Level 3		
14. Det finns en gemensamt överenskommen och dokumenterad processbeskrivning för området Helhetsperspektiv.	Nej	
15. Det finns en gemensamt överenskommen strategi för området helhetsperspektiv.	Nej	
16. Erfarenheter analyseras gemensamt i syfte att förbättra arbetet med området helhetsperspektiv.	Nej	

Historik Save

Copyright © InnovationLab 2012 - 2018

Figur 10: Uppfyllande av mål i verktyget Improve

Figur 10 hänvisar till de fyra mål som identifierats för artefakten samt hur målen uppnås i verktyget Improve. Eftersom modellen har implementerats i verktyget kan vi konstatera att (i) *modellen går att digitalisera*. Målet visas som nummer fyra i figur 10. Fyran har placerats uppe till vänster i bilden men hade kunnat placeras var som helst då den enbart påpekar en lyckad implementering av modellen. Vid användning av modellen läser användaren ett antal påståenden som de senare behöver besvara med ja eller nej samt kommentera vid behov. Siffran ett i figur 10 visar denna enkla utformning där användare kan klicka på rullgardinen för att välja ja eller nej samt kommentera genom att skriva i fältet till höger om påståendet. Utformningen ansågs som enkel att använda och förstå av de praktiker som utvärderat modellen och påvisar därmed att målet att (ii) *modellen skall vara enkel att använda och förstå* har uppnåtts. Påståendena i modellen har kopplats till tre kapacitetsnivåer likt CMMI och ISO/IEC 15504s modeller för processutvärdering. Påståenden som kopplats till tidigare kapacitetsnivåer är mer operativa medan mer övergripande strategiska påståenden tillhör högre nivåer i linje med CMMI och ISO-standarderna. Därmed uppfylls målet att (iii) *modellen skall följa en internationell standard*. Nummer tre i figur 10 visar att påstående 1-11 tillhör nivå ett, 12-13 nivå två, 14-16 tillhör nivå tre. Det sista målet att (iv) *modellen skall vara utformad enligt CALMS* går att urskilja i figur 10 vid siffran två. Tvåan visar de påståendesområden som samtliga påståenden kategoriserats efter. Ett av områdena är automation som direkt är en av delarna i CALMS. Resterande delar av CALMS (Culture, Lean, Measurement och Sharing) berörs istället inom varje påståendebeskrivning. Vid granskning av samtliga processområden kan därmed samtliga delar av CALMS identifieras.

4.3 Beskrivning av designprinciper

Resultatet av designforskning är inte bara skapandet av innovativa artefakter, resultatet är även kunskap om hur det kan skapas andra instanser av artefakter som berör samma klass av problem, kunskapen kallas även för designprinciper. Våra designprinciper har identifierats och formats under det steg som ADR kallar för reflektion och lärande. Principerna har senare definitivt formulerats för att vara generaliserbara för en klass av problem, de går därmed att använda vid andra tillfällen då samma problemområde berörs. Problemområdet har generaliserats till: *Hur bör en IT-artefakt som möjliggör bedömning av DevOps utformas?* Designprinciperna har utformats enligt van Akens (2004) struktur för designprinciper och ser ut enligt följande:

Vid skapandet av en bedömningsmodell för DevOps i en IT-kontext bör det användas en betygsskala uppdelad på fyra kapacitetsnivåer.

I denna designprincip förespråkar vi att liknande modeller i samma kontext behöver utformas utifrån en betygsskala uppdelad på fyra kapacitetsnivåer. CMMIs kapacitetsmodell är väletablerad och används av många verksamheter med syfte att förbättra processer (Adalı, Özcan-Top, Demirörs 2016), därför förespråkar vi att framtida bedömningsmodeller i en IT-kontext bör utgå från samma betygsskala som denna kapacitetsmodell, inte bara för att modellen känns igen men även då det visats i vår undersökning att modellen följer ett logiskt steg som är lätt för verksamheter att förstå. Det framgår både från litteraturen och empirin i de genomförda utvärderingsepisoderna att en bedömningsmodell för processutvärdering bör vara utformad efter kapacitetsnivåer snarare än mognadsnivåer som använts flertalet av de befintliga modeller som analyserats i sektion 2.3. Det vill säga att verksamheter bör kunna bedöma och utveckla enskilda processområden oberoende av resterande processer i verksamheten, vilket modeller som baseras på mognadsnivåer inte tillåter då de ställer generella krav på processer ur ett helhetsperspektiv i verksamheten.

Vid skapandet av en bedömningsmodell för DevOps i en IT-kontext bör påståenden som används i modellen vara förändrings- och anpassningsbara då verksamheter är unika.

Designprincipen har utformats då vi förespråkar att liknande modeller i samma kontext behöver vara dynamiska i sin utformning för att gå i linje med verksamhetens processer. Det finns en risk när verksamheter vill förbättra sina processer att de följer modeller slaviskt vilket kan leda till motsatt effekt. Modeller bör därför vara utformade med initiala påståenden som är generella så de kan anpassas till flera typer av IT-verksamheter. Att påståendena är generella räcker dock inte, de behöver även kunna förändras och/eller anpassas till en mer detaljerad nivå för att göra de mer tolkningsbara av verksamheter. För att möjliggöra förändringsbarheten som vi förespråkar i denna designprincip, anser vi därför att modeller bör implementeras i ett digitalt verktyg.

Vid skapandet av en bedömningsmodell för DevOps i en IT-kontext bör modellen utvecklas så att Dev och Ops kan utföra utvärderingen tillsammans.

Den sista designprincipen som identifierats syftar framförallt till det grundläggande målet med DevOps, det vill säga att riva ner silos, brygga samman och minska gapen mellan avdelningarna Dev och Ops (Verona 2016; Smeds, Nybom, Porres 2015). Avdelningarna

måste därför träffas och utföra utvärderingarna tillsammans, först då kan de utveckla förståelse för varandras arbete, vilket krävs för att skapa en DevOps-kultur i verksamheten. Vi menar att alla bedömningsmodeller som skall användas för att bedöma DevOps bör följa denna designprincip då den säkerställer att avdelningarna börjar kommunicera med varandra.

The screenshot shows the 'improve' tool interface for 'Processes Assessment - statement deviations'. The table below represents the data shown in the interface:

Statement	Supplier	Client	Id	Comment	Prio
1. Dev och Ops har synliga processer som följs av alla berörda medarbetare.	+	+			
2. Dev och Ops har förståelse för varandras verksamheter.	+	+			
3. Dev och Ops utvecklar och tillvaratar medarbetarnas kompetens.	+	+			
4. Dev och Ops har kunskap om hur deras delar i processen bidrar till värde för kund.	+	+			
5. Dev och Ops förstår hur interna och externa underleverantörens processer bidrar till kundvärde.	-	-			
6. Dev och Ops förstår hur deras egna processer påverkar interna och externa underleverantörens processer.	-	-			
7. Dev och Ops förstår hur deras egna processer påverkar interna och externa underleverantörens processer.	+	+			
8. Det finns processägare som bevakar och styr sina processer.	+	+			
9. Dev och Ops tar tillvara sina resurser (människor, teknik och ekonomi) på ett optimalt sätt.	+	-			
10. Dev och Ops uppmuntrar beteenden som sätter till en DevOps-inriktad kulturändring.	-	+			
11. Dev och Ops försöker tillsammans skapa en kultur som stöder utveckling och förändring.	+	+			
12. Det finns en ömsesidig bevakning av alla aktiviteter i området "helhetsperspektiv" genomföra.	-	-			
13. Personal utbildas vid behov.	-	-			
14. Det finns en gemensamt överenskommen och dokumenterad processbeskrivning för området Helhetsperspektiv.	+	+			
15. Det finns en gemensamt överenskommen strategi för området helhetsperspektiv.	-	-			
16. Förbättra arbetet med området helhetsperspektiv.	-	-			

Figur 11: Uppfyllande av designprinciper i verktyget Improve

Figur 11 hänvisar till de designprinciper som identifierats för utförande av en bedömningsmodell för DevOps i en IT-kontext. Modellen bör använda sig av en betygsskala uppdelad på fyra kapacitetsnivåer (se nummer 1). Modellens påståenden bör vara förändrings- och anpassningsbara (se nummer 2) samt att den utformas så att Dev och Ops kan utföra utvärderingen tillsammans (se nummer 3).

4.4 Utvärdering av IT-artefakt och designprinciper

Då *reflektion och lärande* fasen skett kontinuerligt under hela utvecklingsprocessen och parallellt med ADRs två första steg: *problemformulering* och *bygga, intervention och utvärdera*, presenteras resultatet utifrån de olika utvärderingsepisoderna som genomfördes (se kapitel 3). Då syftet med utvärderingen inte bara har varit att lösa en specifik instans av ett problem utan snarare en klass av problem, har den kunskap anskaffats presenterats i form av tre designprinciper (se sektion 4.3). Hur dessa designprinciper uppenbarats genom olika iterationer beskrivs även i de fyra utvärderingsepisoderna nedan.

4.4.1 Utvärderingsepisod 1

Vid första utvärderingsepisoden fastslogs det att de initiala påståendena för processbedömning kan användas vid skapandet av en bedömningsmodell för DevOps. Vid utvärderingen framgick dock att det inte var lämpligt att gruppera påståenden på akronymen CALMS vilket gjorts i den första iterationen av artefakten. Då påståendena som utvecklats av InnovationLab inte ursprungligen kategoriserats på CALMS blev de osammanhängande när de omgrupperats. Vi fastslog därför att vi inför nästa iteration borde återgå till de ursprungliga påståendena som skapats av forskningsgruppen med motiveringen att vi fortfarande ansåg att påståendena gick i linje med CALMS oavsett hur de kategoriserats, men att de med den ursprungliga kategoriseringen var mer sammanhängande och lättare att tolka.

Modellen hade skapats utifrån CMMIs kapacitetsnivåer då den har som mål att följa en internationell standard. Modellen behövde därför utvärderas i jämförelse med standardens ursprungsmodell i syfte att säkerställa att målet uppfylldes. Vid utvärderingen av modellen identifierades en del avvikelser mellan de två modellerna. Vi insåg att en del av de påståenden som vi initialt ansåg tillhöra nivå två eller tre i modellen inte motsvarade CMMIs mål för respektive nivå. Vi identifierade även att det saknades en enhetlighet i de krav som ställdes för att uppnå nivå två och tre mellan de olika CALMS-områdena i modellen. CMMIs modell förespråkar att processområdena följer en enhetlig standard vid nivå två och tre vilket vår modell inte uppfyllde. För att få vår modell att bättre överensstämma med CMMIs kapacitetsnivåer behövde många påståenden flyttas till tidigare nivåer. CMMI anser att nivå ett uppnås när processer används och kan utföra det arbete som krävs för att producera arbetsprodukter samt uppfyller de uppsatta målen för processområdet. Ur ett DevOps-kontext ansåg vi att många av de påståendena som tidigare var placerade på nivå två och tre var för grundläggande när det tolkades ur ett CMMI-perspektiv och flyttades därmed ner till nivå ett. CMMI kategoriserar processområden som hanterar kompetensutveckling, involvering av intressenter och monitorering av processer till nivå två. Den första versionen av modellen saknade påståenden som berörde dessa områden eller var för starkt kopplat till ett specifikt CALMS-område, vilket gjorde att vi behövde skapa nya påståenden som bättre mappade CMMIs modell. I tabell 4 presenterar vi ett av modellens påståendekområde *Delning av erfarenheter*, samtliga påståenden varierar i de olika områdena men har utformats på samma sätt. Vid utvärdering av kapacitetsnivå tre påvisades att påståendena inte gick i linje med CMMIs modell där processområdena bland annat beskrivs som standardiserade och med en mer rigorös beskrivning. Det fastslogs därmed att även påståendena på denna nivå behövde ändras för att överensstämma med CMMI.

Sammanfattningsvis fastslår den första utvärderingen av artefakten att den uppnådde ett av de fyra uppsatta målen. Då inga potentiella användare har deltagit i utvärderingen gick det inte bedöma huruvida vi uppnått artefaktens mål att (i) *modellen skall vara enkel att använda och förstå*. Det framgick även att artefakten misslyckas med målet att (ii) *modellen följer en*

internationell standard då den inte överensstämmer med CMMIs kapacitetsmodell. Innan UE1 genomfördes kunde vi fastslå att vi redan uppnått målet att (iii) *modellen skall kunna digitaliseras* då vi lyckats med att implementera artefakten i det digitala verktyget Improve. Målet var därmed redan uppfyllt och behövde inte utvärderas vidare i utvärderingsepisoden. Även om artefakten initialt uppfyllde målet att (iv) *modellen skall vara utformat efter CALMS* behöver detta utvärderas ytterligare då modellen kommer struktureras om för att istället kategoriseras utifrån sina ursprungliga påståendekområden. Det fastslogs därmed i UE1 att vi inte lyckats skapa en IT-artefakt som möjliggör bedömning av DevOps.

Tabell 4: Utvärderingsepisod 1 - Delning av erfarenheter

Nivå	Status	Delning av erfarenheter
1	Utförd	Dev och Ops har synliga processer som följs av alla berörda medarbetare
		Dev och Ops har en operationell modell som möjliggör delning av resurser
		Dev och Ops har gemensamma rutiner (tex möten) som stödjer att alla relevanta roller bidrar till tjänsteutveckling
2	Hanterad	Dev och Ops informerar om problem och risker, som kan påverka leverans av tjänsten
		Dev och Ops identifierar gemensamt vilka kompetenser roller kan bidra med för att tillhandahålla tjänster
		Dev och Ops utvecklar och tillvaratar medarbetarnas kompetens
3	Definierad	Dev och Ops delar med sig av tjänsterelevant information och kunskap till andra intressenter
		Resultat av mätvärden tillgängliggörs för relevanta intressenter
		Dev och Ops ger feedback om lösningar till alla intressenter

Vid initial utveckling av artefakten, alltså innan UE1 hade det framkommit att det var enkelt att förändra och utveckla artefakten då den implementerats i ett digitalt verktyg. Enkelheten ansågs som en stor fördel, dels ur ett utvecklingssyfte, men även att artefakten ansågs som flexibel och kunde därmed anpassas till olika verksamheter vilket stödjer designprincipen att (i) *bedömningsmodellen bör vara förändrings- och anpassningsbar*. Även om vi insåg att påståendena i modellen var kopplade till fel kapacitetsnivåer (då de skulle motsvara CMMI och ISO/IEC 15504 modeller) framkom en tydlighet i modellens utformning när det fanns en koppling mellan påstående och de fyra kapacitetsnivåerna. Denna tydlighet styrker designprincipen att (ii) *bedömningsmodellen bör använda en betygsskala uppdelad på fyra kapacitetsnivåer*. Sammanfattningsvis användes den samlade erfarenheten som införskaffats under denna episod vid samtliga av de framtida utvärderingsepisoder som genomfördes vilket bidrog till att designprinciperna stegvis uppenbarades.

4.4.2 Utvärderingsepisod 2

I utvärderingsepisod två framgick det att respondenterna från organisation A tyckte att frågorna var relevanta och bra med motiveringen att de väckte rätt frågor som behöver ställas i verksamheter för att reda ut gapen mellan Dev och Ops i projekt. Samtidigt riktades en viss kritik till att påståendena ibland var för övergripande men samtidigt förstod de varför de var utformade på det sättet då påståendena skall passa flera verksamheter. Det förklarades även av facilitatorn att det går att göra individuella anpassningar i verktyget vilket uppfattades som positivt. När modellens påståenden diskuterades vidare framgick det att respondenten tyckte att påståendena var relevanta ur ett DevOps kontext, dock utvärderades inte påståendenas koppling till CALMS då det upplevdes att respondenterna inte hade tillräckliga kunskaper för att bedöma denna koppling. Det framgick även i utvärderingsepisoden att respondenterna ställde sig positiva till att använda sig av kapacitetsnivåer i modellen, dels för att de lätt kan se vad som behöver förbättras men även för att arbeta med uppföljning och ständiga förbättringar. Respondenterna ställde sig även positiva till hur de olika kapacitetsnivåerna i modellen har utformats, där påståenden tillhörande nivå ett är mer operativa för att i nivå tre bli mer strategiskt utformade. Respondenterna menade dels att operativa påståenden är mer lättbegripliga än strategiska vilket gör att de bör tillhöra nivå ett. De tyckte även att det är positivt att utförlig dokumentation och standardisering infaller på nivå tre, då de anser att den typen av arbete är mer krävande och enbart nödvändig vid strävan efter en högre nivå. I tabell 5 presenterar vi ett av modellens påståendekområde *Helhetsperspektiv* som är den av de fem områden som modellen kategoriserats på. I tabellens syns vilka påståenden som kopplats till vilken kapacitetsnivå.

Sammanfattningsvis fastslår den andra utvärderingsepisoden att artefakten har uppnått två av de fyra uppsatta målen. Gällande målet att (i) *modellen skall vara enkel att använda och förstå* krävs utvärdering i fler organisationer då organisation A verkade tveksamma till att påståenden i modellen ibland var för övergripande, vilket gjorde det svårt att skapa en enhetlig syn på hur de skulle tolkas. Målet med att (ii) *modellen skall följa en internationell standard* anses uppnådd då respondenterna visat sig positiva för hur de olika kapacitetsnivåerna i modellen utformats. Utvärderingen visade inga tecken på det tidigare uppnådda målet att (iii) *modellen skall kunna digitaliseras* skall ha förändrats och anses därför fortfarande som uppnådd. Då inga frågor ställdes till respondenterna angående påståendenas koppling till CALMS kan inte utvärderingen fastställa om målet att (iv) *modellen skall vara utformad efter CALMS* har uppnåtts. Det fastslogs därmed i UE2 att vi inte lyckats skapa en IT-artefakt som möjliggör bedömning av DevOps.

Tabell 5: Utvärderingsepisod 2 - Helhetsperspektiv

Nivå	Status	Helhetsperspektiv
1	Utförd	Dev och Ops har synliga processer som följs av alla berörda medarbetare.
		Dev och Ops har förståelse för varandras verksamheter.
		Dev och Ops utvecklar och tillvaratar medarbetarnas kompetens.
		Dev och Ops har kunskap om hur deras delar i processen bidrar till värde för kund.
		Dev och Ops förstår hur interna och externa underleverantörers processer bidrar till kundvärde.
		Dev och Ops förstår hur deras egna processer påverkar interna och externa underleverantörernas processer.
		Dev och Ops tar ett helhetsansvar för tjänstens livscykel.
		Det finns processägare som bevakar och styr våra processer.
		Dev och Ops tar tillvara sina resurser (människor, teknik och ekonomi) på ett optimalt sätt.
2	Hanterad	Det finns en ömsesidig bevakning av att aktiviteter i området "Helhetsperspektiv" genomförs.
		Personal utbildas vid behov
3	Definierad	Det finns en gemensamt överenskommen och dokumenterad processbeskrivning för området Helhetsperspektiv.
		Det finns en gemensamt överenskommen strategi för området helhetsperspektiv.
		Erfarenheter analyseras gemensamt i syfte att förbättra arbetet med området helhetsperspektiv.

Respondenterna i UE2 ställde sig positiva till kapacitetsnivåernas utformning samt hur de uppnås. Alltså stöds designprincipen att (i) *en bedömningsmodell bör använda en betygsskala uppdelad på fyra kapacitetsnivåer*. Det framgick att organisation A upplevde en osäkerhet kring de påståenden som var beskrivna på en övergripande nivå. Om verksamheten själva hade kunnat förändra eller skapa påståenden hade de tillsammans kunnat utforma dessa vilket stödjer designprincipen att (ii) *påståenden som används i modellen bör vara förändrings- och anpassningsbara då verksamheter är unika*. När påståendena var för övergripande upplevde de en svårighet i att skapa en gemensam bild av påståendet. Problemet förespråkar designprincipen att (i) *modellen bör utvecklas så att Dev och Ops kan utföra utvärderingen tillsammans*. Om avdelningarna utvärderar tillsammans minskar risken för missförstånd samtidigt som det säkerställer att samtliga tolkat påståenden i modellen på samma sätt.

4.4.3 Utvärderingsepisod 3

Inledningsvis kritiserade respondenten från organisation B om DevOps egentligen kan göras mätbart då verksamheters processer till stor omfattning varierar och menar att ingen verksamhet är den andra lika. Respondenten menar att tidigare existerande bedömningsmodeller lätt kan tolkas som ett facit på hur verksamheters och dess processer skall drivas, därför blir det ofta fel när verksamheter ska anpassa sig efter dessa modeller. Ingen verksamhet är den andra lik och därför bör påståendena kunna förändras beroende på verksamheten. Respondenten menade att nutida ramverk inte har denna förmåga och blir därför snabbt förgångna, att vårt verktyg är digitalt och förändringsbart sågs därmed som ett stort plus av respondenten. Det förklarades att verksamheters kunskaper om DevOps växer längs med ett förändringsarbete, därför måste även processer, ramverk och modeller som dessa utgå från att kunna anpassas längs med förändringsarbetet. Om ramverk och modeller används för fyrkantigt av organisationer kommer de motverka sitt syfte och istället försvåra förändringsarbetet. Respondenten menar därmed att inte att det är fel att använda sig av ramverk eller modeller, men verksamheter måste förstå hur dessa skall användas med utgångspunkt från sin egen verksamhet. För att säkerställa att vår modell används på ett korrekt sätt vid utvärdering, menar respondenten att det krävs en duktig och kunnig facilitator som kan vägleda de som använder sig av modellen. Med syfte att bland annat säkerställa att alla har förstått alla begrepp och påståenden men även för att stimulera dialoger och styra deltagarna i rätt riktning.

Vidare förklarades det att det är viktigt att arbeta med beteendeförändringar när verksamheter vill lyckas med sin implementering av DevOps. Problematiken med att gruppera påståenden i en modell diskuterades då många verksamheter ser och tolkar DevOps olika. Gällande vår modell ansåg respondenten att det kan vara en bra ide att utgå från CALMS då akronymen säkerställer att verksamheter försöker skapa en kultur som bottnar i beteendeförändring. Beteendeförändring anser respondenten vara så pass viktiga och grundläggande att det diskuterades att få in påståenden som möjliggör dessa redan på kapacitetsnivå ett på samtliga områden i modellen.

Sammanfattningsvis behöver modellen anpassas för att på ett tydligare sätt förespråka beteendeförändringar i verksamheter, detta kan uppnås genom att införa nya påståenden på nivå ett i respektive CALMS-område. Det framgick även att modellen behöver granskas så att påståenden inte är för detaljerade utan kan anpassas till alla typer av verksamheter. Det finns en risk när påståenden är för väldetaljerade att de hämmar verksamheters förändringsarbete, speciellt om verksamheter använder modellen på ett felaktigt sätt och följer den slaviskt utan eftertanke. För att eliminera dessa problem bör modellen vara utformad med övergripande påståenden som går att anpassa oavsett verksamhet. Det framgick även att en facilitator bör leda utvärderingen när modellen används i verksamheter för att säkerställa att den används på ett korrekt sätt. Att modellen var digital ansågs som mycket positivt då det underlättar genomförandet av förändringar för att anpassa den till olika verksamheter.

Respondenten kritiserar även CMMIs mognadsnivåer då de kan hämma verksamheters utveckling. Mognadsnivåer är utformade på ett sätt som innebär att verksamheter ibland påtvingas att förbättra processområden enbart för att uppnå en högre mognadsnivå, trots att affärsverksamheten inte nödvändigtvis påverkas. Han menar att alla verksamheter är unika och därmed även deras behov, vilket gör att mognadsnivåer är förkastliga. Därmed styrker respondenten att det är bättre att utgå ifrån CMMIs kapacitetsnivåer då nivåerna tillåter verksamheter att fokusera på de processer eller områden som de anser är viktigast för sin verksamhet.

Sammanfattningsvis fastslår UE3 att modellen uppnått två av de fyra uppsatta målen. Gällande målet att (i) *modellen skall vara enkel att använda och förstå* upplevde respondenten att detta uppfyllts. Vi upplevde dock att nästa utvärderingsperiod behövde genomföras för att utvärdera målet ytterligare innan det kunde styrkas att målet var uppfyllt. Då artefakten inte genomgått några förändringar gällande kapacitetsnivåer sen tidigare utvärderingsepisod samt att artefakten sen tidigare har implementerats i det digitala verktyget Improve anser vi att den fortfarande uppfyller målen att (ii) *modellen skall följa en internationell standard* samt att (iii) *modellen skall kunna digitaliseras*. Artefakten anses inte uppfylla målet att (iv) *modellen skall vara utformad efter CALMS* då våra påståenden inte förtydligande tar upp beteendeförändring och kultur vilket är en betydande faktor i CALMS. Det fastslogs därmed i UE3 att vi inte lyckats skapa en IT-artefakt som möjliggör bedömning av DevOps.

Vid UE3 stärktes samtliga tre designprinciper som identifierats. Respondenten styrker att (i) *en bedömningsmodell bör använda en betygsskala uppdelad på fyra kapacitetsnivåer* då han anser att CMMIs kapacitetsnivåer är en bättre utgångspunkt än mognadsnivåer vid skapandet av en bedömningsmodell då alla verksamheter är unika. Att verksamheter är unika var även en viktig grund för den andra designprincipen att (ii) *påståenden som används i modellen bör vara förändrings- och anpassningsbara då verksamheter är unika*. Respondenten påpekade vikten av att ramverk och modeller måste vara förändringsbara då det annars finns en risk att de används på fel sätt, vilket han även påstår är ett vanligt problem historiskt sett inom verksamheter som vill implementera DevOps. Respondenten påpekade även att en facilitator bör leda utvärderingen när modeller likt vår används med syftet att motverka ovanstående problem, respondenten gav även flera andra motiv till varför det är viktigt att använda sig av en facilitator. Ett av motiven var att DevOps till stor del handlar om att skapa en beteendeförändring som gör att Dev och Ops börjar kommunicera. Om utvärderingen inte leds av en facilitator finns det en risk att denna beteendeförändring inte framgår vilket gör att det slutgiltiga målet att skapa en ny DevOps-kultur inte uppnås. Det gick därmed att fastslå att vår designprincip att (iii) *bedömningsmodellen bör utvecklas så att Dev och Ops kan utföra utvärderingen tillsammans* hålls med av respondenten.

4.4.4 Utvärderingsepisod 4

Det framgick i UE3 att påståenden i modellen måste vara anpassningsbara för de verksamheter som använder sig av modellen då de annars riskerar att tolka påståendena som ett facit på hur verksamheten och dess processer skall drivas. Inför UE4 hade vi därmed ifrågasatt några av de påståenden som skapats av InnovationLab som respondenten deltagit i. Vi ansåg att påståenden som: *Dev och Ops förstår hur interna och externa underleverantörers processer bidrar till kundvärde* samt *Dev och Ops förstår hur interna och externa underleverantörers processer bidrar till kundvärde* eventuellt var för specifikt formulerade för IT-verksamheter då de kan sakna underleverantörer vilket behövde diskuteras med respondenten. Respondenten ansåg dock att underleverantörer kan ses på olika sätt i verksamheter och styrker därmed att dessa är relevanta påstående att ha med i modellen. Respondenten menar också att påståendena är framtagna i direkt samarbete med verksamheter och har därmed accepterats av de praktiker som deltagit i forskningsprojektet. Även om det finns undantag där verksamheter inte har någon form av underleverantör kan de själva välja att ta bort påståendet i modellen om så önskas. Episoden styrker därmed tidigare anskaffad kunskap om att artefakten bör vara förändringsbar till den verksamhet som använder modellen.

Inför UE4 hade två påståenden lagts till i modellen för att komplettera de befintliga som skapats av forskningsgruppen InnovationLab, syftet med de nytillkomna påståendena var att stärka en beteendeförändring redan på kapacitetsnivå ett, se tabell 6. Det första påståendet såg ut enligt följande:

Dev och Ops försöker tillsammans skapa en kultur som stödjer utveckling och förändring

Påståendet har skapats för att säkerställa att Dev och Ops arbetar mot att skapa en kultur som stödjer utveckling och förändring. Om avdelningarna inte stödjer utveckling och förändring går det inte att uppnå en DevOps-kultur. Påståendet kan därmed ses som en grundpelare som måste uppnås av verksamheter på samtliga påståendeeområden i artefakten. Det räcker inte med att både Dev och Ops är förändringsbara, betoningen i påståendet ligger på ordet *tillsammans*, det vill säga att Dev och Ops försöker uppnå denna kulturförändring tillsammans. Ytterligare ett påstående har skapats som kompletterar det första påståendet men inriktar sig mer mot beteende. Påståendet ser ut enligt följande:

Dev och Ops uppmuntrar beteenden som syftar till en DevOps-inriktad kulturförändring

För att verksamheter skall uppnå en kultur där Dev och Ops inte längre arbetar i silos och istället kommunicerar med varandra, krävs det att verksamheter uppmuntrar sådant beteende. Respondenten från organisation B påpekade vid ett flertal gånger vikten av att skapa beteendeförändringar som i sin tur leder till att det skapas en ny kultur. Det vill säga att kulturen i verksamheter är ett resultat eller en effekt av rådande beteenden. Om verksamheter uppmuntrar beteenden som bygger samman Dev och Ops är förhoppningen att verksamheten snabbare kan röra sig mot en DevOps-kultur. Detta påstående syftar därmed till att säkerställa att verksamheter uppmuntrar beteendeförändringar för att i slutändan uppnå en DevOps-kultur.

Respondenten ställde sig positiv till de påståenden som skapats och ansåg att borde vara med i modellen då de redan i ett tidigt stadie säkerställer en beteendeförändring för att uppnå en DevOps-kultur i verksamheten. Respondenten ansåg även att det var bra att dessa påståenden implementerats på samtliga påståendeområden i modellen. Respondenten ser även en fördel med att kultur uttrycks bokstavligen i modellen då detta är en del av CALMS. Det förklarades av respondenten att han ansåg att samtliga påståendeområden i modellen gick i linje med CALMS. Respondenten ansåg även att modellen var enkel och tydlig i sitt utformande då påståenden tydligt kunde kopplas till de utformade kapacitetsnivåerna, vilket respondenten anser syns tydligt i verktyget Improve.

Sammanfattningsvis fastslår UE4 att modellen har uppnått samtliga av de fyra uppsatta målen för artefakten. Utvärderingsepisoden påvisade att artefakten uppfyller målet att (i) *modellen skall vara enkel att använda och förstå* då dess påståenden tydligt kan kopplas till kapacitetsnivåerna. Sedan tidigare utvärderingsepisoder uppnåddes målen att (ii) *modellen skall följa en internationell standard* samt att (iii) *modellen skall kunna digitaliseras*. Då inga förändringar har skett gällande dessa mål kopplat till artefakten ansågs de fortfarande som uppnådda. Artefakten har nu påståenden som berör beteendeförändring och kultur vilket ansågs från UE3 som ett krav för artefakten för att uppfylla målet att (iv) *modellen skall vara utformad efter CALMS*. Respondenten från UE4 ställde sig även positiv till dessa påståenden samt de påståendeområdena som artefakten innehåller då de går i linje med CALMS. Därför ansågs detta mål som uppnått. Modellen uppfyller därmed de fyra målen och kan därmed fastslå att vi lyckats skapa en IT-artefakt som möjliggör bedömning av DevOps.

Tabell 6: Utvärderingsepisod 4 - Helhetsperspektiv

Nivå	Status	Helhetsperspektiv
1	Utförd	Dev och Ops har synliga processer som följs av alla berörda medarbetare.
		Dev och Ops har förståelse för varandras verksamheter.
		Dev och Ops utvecklar och tillvaratar medarbetarnas kompetens.
		Dev och Ops har kunskap om hur deras delar i processen bidrar till värde för kund.
		Dev och Ops förstår hur interna och externa underleverantörers processer bidrar till kundvärde.
		Dev och Ops förstår hur deras egna processer påverkar interna och externa underleverantörernas processer.
		Dev och Ops tar ett helhetsansvar för tjänstens livscykel.
		Det finns processägare som bevakar och styr våra processer.
		Dev och Ops tar tillvara sina resurser (människor, teknik och ekonomi) på ett optimalt sätt.
		Dev och Ops uppmuntrar beteenden som syftar till en DevOps-inriktad kulturförändring
		Dev och Ops försöker tillsammans skapa en kultur som stödjer utveckling och förändring

2	Hanterad	Det finns en ömsesidig bevakning av att aktiviteter i området "Helhetsperspektiv" genomförs.
		Personal utbildas vid behov
3	Definierad	Det finns en gemensamt överenskommen och dokumenterad processbeskrivning för området Helhetsperspektiv.
		Det finns en gemensamt överenskommen strategi för området helhetsperspektiv.
		Erfarenheter analyseras gemensamt i syfte att förbättra arbetet med området helhetsperspektiv.

Den sista utvärderingsepisoden bidrog till vidare kunskap som stärkte våra designprinciper. Respondenten ansåg att CMMIs kapacitetsnivåer med en betygsskala uppdelad på fyra kapacitetsnivåer är en bra grund för vår modell och motiverade detta med att våra påståenden och nivåer går i linje med hur CMMI strukturerar sina mål och kapacitetsnivåer. Samtidigt förkastar respondenten CMMIs mognadsnivåer med motiveringen att de "är uråldriga och uppdateras inte ens längre". Därmed stärktes vår första designprincip att (i) *bedömningsmodellen bör använda en betygsskala uppdelad på fyra kapacitetsnivåer*. Respondenten håller även med tidigare organisationer som yttrat sig att det är viktigt att påståenden i modellen skall vara dynamiska för att kunna anpassas till flera verksamheter, därmed stärktes även designprincipen att (ii) *påståenden som används i modellen bör vara förändrings- och anpassningsbara då verksamheter är unika*. Respondenten var även enig med respondenten från organisation B gällande att modellen bör utvecklas så att Dev och Ops kan utföra utvärderingen tillsammans för att den skall vara verksam och styrker därmed designprincipen att (iii) *bedömningsmodellen bör utvecklas så att Dev och Ops kan utföra utvärderingen tillsammans*.

4.4.5 Sammanfattning av utvärdering

Sammanfattningsvis påvisar våra utvärderingsepisoder att det har varit framgångsrikt att basera artefakten på relaterad litteratur vilket resulterat i att den enbart fått mindre påpekningar och förändringsförslag vid de olika episoderna. Analys av både generella och specifika modeller har varit till nytta vid utformning av artefakten och dess kapacitetsnivåer. Att bara utvärdera modellen på en artificiell nivå hade däremot varit omöjligt, vilket även anses av ADR som är den metod vi använt oss av. Att artefakten måste testas i verksamheter visades i samtliga utvärderingsepisoder då konsekvensen av episoderna var att artefakten behövde utvecklas vidare. Att organisation A och B varierade i storlek ansågs även vara väldigt positivt i vår utvärdering då den samlade kunskapen som anskaffades därmed kunde ses som mer generaliserbar. Målen har varit en bra utgångspunkt för utvärdering, dels för att säkerställa att rätt frågor ställdes, men även för att identifiera när modellen kunde anses som klar.

Den samlade erfarenheten från samtliga utvärderingsepisoder är att artefakten upplevs som enkel att använda och förstå. Det kan vara svårt för användare att hålla isär artefakten (bedömningsmodellen) och det digitala verktyget (Improve) som den används i, vilket leder till att artefakten saknar fullständig evidens som styrker huruvida den är enkel att använda och förstå. Skapandet av sådan omfattande evidens ligger dock utanför ramarna för denna uppsats då syftet inte har varit att utvärdera Improves användbarhet. Vårt empiriska material visar att praktiker som testat artefakten upplever att den är enkel att använda och förstå, vi anser därmed att målet att (i) *modellen skall vara enkel att använda och förstå* har uppnåtts även om mer omfattande forskning kan användas för att styrka måluppfyllnaden. Utvärderingsepisoderna påvisar att målet att (ii) *modellen skall följa en internationell standard* har uppnåtts då respondenterna visat sig positiva till användningen av CMMIs kapacitetsnivåer i bedömningsmodellen. De menar att artefakten genom kapacitetsnivåer möjliggör arbetet med uppföljning och ständiga förbättringar för Dev och Ops. Redan innan utvärderingsepisoderna kunde vi fastslå målet att (iii) *modellen skall kunna digitaliseras* var uppnått då en lyckad implementering av modellen genomfördes i det digitala verktyget Improve. Att modellen går att digitalisera har varit till stor fördel ur ett utvärderings- och utvecklings syfte. Utvärderingarna har visat att artefakten har lyckats med målet att (iv) *modellen skall vara utformad efter CALMS*. Då påståenden tillkommit vilka behandlar beteendeförändring samt att modellen består av påståenden som behandlar CALMS menar utvärderingen att modellen uppfyller detta mål. Artefakten anses därmed vara färdig då den har uppnått samtliga av de uppsatta målen och kan användas med syfte att tillfredsställa de behov och lösa de problem som identifierats i kapitel 2.

För att identifiera designprinciperna behövde vårt problem ses som en klass av ett problem snarare än en enskild instans. Vi behövde även utgå från samma förhållningssätt när vi skapat vår lösning där vi istället försökte skapa en klass av lösningar. Med hjälp av detta förhållningssätt uppenbarades tre designprinciper som blivit allt tydligare i de senare utvärderingsepisoderna. Genom att testa och utvärdera artefakten i flera utvärderingsepisoder så utvärderades även designprinciperna. Designprincipen att (i) *bedömningsmodellen bör använda en betygsskala uppdelad på fyra kapacitetsnivåer* har stärkts i samtliga UE. Respondenterna påpekade vikten av att verksamheter själva får prioritera vilket eller vilka processområden de behöver fokusera på, samtidigt som strukturen på kapacitetsmodellen känns igen och är begripligt av verksamheter. Att (ii) *påståenden som används i modellen bör vara förändrings- och anpassningsbara då verksamheter är unika* ansågs som viktigt framförallt i de UE som genomfördes hos organisation A och B. Respondenterna från organisation A påpekade vikten av att modellen skulle vara dynamisk för att vara användbar.

Det framgick tydligt att det inte går att utveckla en detaljerad modell som passar alla verksamheter, därför måste påståenden formuleras på en mer övergripande nivå för att sedan anpassas till den verksamhet som använder sig av modellen. Respondenten från organisation B påpekade risken med att modeller används på fel sätt om de inte är anpassningsbara, vilket innebär att organisationer kan arbeta mot mål som inte är relevanta ur ett affärsperspektiv.

För att riva ner silos, brygga samman och minska gapen mellan avdelningarna Dev och Ops, alltså själva syftet med DevOps, krävs det att Dev och Ops kommunicerar med varandra och talar samma språk. Designprincipen att (iii) *bedömningsmodellen bör utvecklas så att Dev och Ops kan utföra utvärderingen tillsammans* är en del i att skapa denna kulturförändring då en utvärdering som genomförs tillsammans säkerställer att de olika avdelningarna kommunicerar och därmed utvecklas en förståelse för varandra vilket kan ses som det första steget mot en DevOps-kultur. Vid UE2-4 som genomfördes i naturalistiska miljöer kunde vi konstatera att samtliga respondenter ställer sig positivt till designprinciper fungera. Vi anser därmed att identifierade designprinciperna kan användas vid framtida utveckling av modeller som skall hantera samma klass av problem.

5 Slutsats

I detta kapitel kommer studiens slutstatser presenteras där motivering ges till att forskningsfrågan besvarats vilket har resulterat i två bidrag, i form av en IT-artefakt samt i form av designprinciper. Avslutningsvis diskuteras studiens genomförande och tillförlitlighet innan förslag till fortsatt forskning ges. Kapitlet innefattar 5.1) Bidrag i form av IT-artefakt, 5.2) Bidrag i form av designprinciper och 5.3) Reflektion och fortsatt forskning.

Det finns ett brett samförstånd att forskning som berör informationssystem måste hantera två saker: dels att lösa det aktuella problemet som adresserats men även att skapa teoretiska bidrag i form av designkunskap (Sein et al 2011; Hevner et al 2004). Det som initierade vår forskning var då det identifierats en avsaknad av designkunskap för utformning av bedömningsmodeller för DevOps. Forskningsfrågan utformades därför enligt följande: *Hur bör en IT-artefakt som möjliggör bedömning av DevOps utformas?* För att besvara frågeställningen har vi använt oss av metoden ADR. Metoden anses lämplig vid forskning som berör informationssystem eftersom den inte bara hjälper forskare att lösa det aktuella problemet men även ser till att anskaffad kunskap bevaras och delas i så kallade designprinciper.

5.1 Bidrag i form IT-artefakt

Forskningen har resulterat i en IT-artefakt i form av en modell som kan användas av verksamheter i syfte att bedöma verksamhetens processer ut ett DevOps-kontext. Målen för artefakten har skapats utifrån ett behov som identifierats när befintliga modeller för bedömning av DevOps analyserats. Artefakten utvecklades i enlighet med metoden i ett flertal iterationer. För att säkerställa att målen har uppnåtts har artefakten utvärderats i autentiska miljöer där praktiker har fått utvärdera modellen. Resultatet av dessa utvärderingsepisoder visar att artefakten går att implementera i ett digitalt verktyg, följer en internationell standard, har baserats på CALMS samt är enkel att använda och förstå. Vi anser därför att artefakten tillgodoser de behov som identifierats och kan därför vara ett bättre alternativ vid bedömning av verksamhetens DevOps, i jämförelse med befintliga bedömningsmodeller. Vi anser att artefakten är generaliserbar och kan därmed användas vid andra instanser av samma problemområde, det vill säga bedömning av verksamhetens DevOps.

5.2 Bidrag i form av designprinciper

Resultatet av designforskning är inte bara skapandet av innovativa artefakter, resultatet är även kunskap om hur det kan skapas andra instanser av artefakter som berör samma klass av problem, kunskapen kallas även för designprinciper (Sein et al 2011). Vi förespråkar att de designprinciper som vi har identifierat bör användas vid utveckling av modeller som skall hantera samma klass av problem som vi adresserat. Vi anser att designprinciperna är generella, empiriskt grundade, har stöd i litteraturen och är lätta att applicera vid utformning av liknande modeller. De tre designprinciperna ser ut enligt följande:

1. *Vid skapandet av en bedömningsmodell för DevOps i en IT-kontext bör det användas en betygsskala uppdelad på fyra kapacitetsnivåer.*
2. *Vid skapandet av en bedömningsmodell för DevOps i en IT-kontext bör påståenden som används i modellen vara förändrings- och anpassningsbara då verksamheter är unika.*

- 3. Vid skapandet av en bedömningsmodell för DevOps i en IT-kontext bör modellen utvecklas så att Dev och Ops kan utföra utvärderingen tillsammans.*

Forskningsfrågan har därmed besvarats, dels som en enskild instans av ett problem men även som en mer generell klass av problem, vilket även var vårt förväntade resultat.

5.3 Reflektion och fortsatt forskning

Vi tror inte att det hade gått att uppnå samma resultat om en annan metod använts för att besvara frågeställningen. Vi motiverar påståendet att ADR har varit en särskilt lämplig metod då den säkerställer att det forskas i den riktiga världen där människor och praktiker involveras. Praktikerna ses inte heller som passiva undersökningsobjekt i metoden, snarare förespråkar ADR att praktiker skall vara delaktiga i forskningen där deras kunskaper anses som extremt viktiga (Johannesson, Perjons 2014). Den kunskap som anskaffats och format artefakten har inskaffats ute hos verksamheter där praktiker har fått ge input och vara delaktiga i utformningen. Vi anser därmed att metoden har varit särskilt lämplig för vårt syfte. En stor del av vårt forskningsresultat har baserats på den kunskap som införskaffats vid utvärderingsepisoderna som genomfördes i samspel med praktiker vilket innebär att andra metoder inte hade gett samma resultat. Det går dock att kritisera att vi inte följt ADR fullt ut då artefakten inte använts och testats av verksamheter under en längre period, vilket innebär att eventuella konsekvenser som uppstår vid användning av artefakten inte har utvärderats, en sådan kritik kan dock alltid ställas vid designforskning. Även om forskningen utförts under en begränsad tidsperiod anser vi att vi till stor del följt metoden. Vi motiverar detta med att vi följt de steg som ingår i ADR vilket har lett till ett resultat både i form av en artefakt samt ett antal designprinciper.

Vid vidare reflektion av den utförda forskningen anser vi att artefakten och designprinciperna bör utvärderas i fler utvärderingsepisoder med syfte att säkerställa att forskningsresultatet är generaliserbart. Då forskningen som genomförts varit delvis begränsad har artefakten enbart utvärderats i samspel med ett fåtal IT-organisationer under en kortare tid. Eftersom alla verksamheter är unika och ser olika ut finns det en risk att vårt resultat (artefakt och designprinciper) inte går att använda i alla IT-verksamheter vilket innebär att kunskapen som införskaffats inte är generaliserbar. Vi uppmuntrar därmed till kritisk granskning av artefakten och designprinciperna i ytterligare forskning genom att testa dem i fler organisationer och under en längre tidsperiod.

Källförteckning

Adali, O. E., Özcan-Top, Ö., & Demirörs, O. (2016, June). *Evaluation of Agility Assessment Tools: A Multiple Case Study*. In *International Conference on Software Process Improvement and Capability Determination* (pp. 135-149). Springer, Cham.

Albliwi, S. A., Antony, J., & Arshed, N. (2014, December). Critical literature review on maturity models for business process excellence. In *Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on*(pp. 79-83). IEEE.

Bass, L., Weber, I., & Zhu, L. (n.d.). *DevOps: A software architect's perspective*.

Beck, K et al. (2001) Manifest för Agil Systemutveckling
<http://agilemanifesto.org/iso/sv/manifesto.html> [2018-05-07]

Cronholm, S. (2018). *Datadriven innovation: algoritmer, plattformar och ekosystem*.
<http://www.hb.se/Forskning/Forskningsportal/Projekt/Datadriven-innovation-algoritmer-plattformar-och-ekosystem/>[2018-04-04]

DASA (2016). *Dasa DevOps Glossary*.
<https://drive.google.com/uc?export=download&id=0B5kVnjup6hPpY09oUld5dldKSjg>
[2018-04-22]

DevOps institute (2017). *DOI scrum org partnership*.
<https://devopsinstitute.com/2017/08/25/doi-scrum-org-partnership/> [2018-04-22]

Fritzsche, M., & Keil, P. (2007). Agile methods and CMMI: compatibility or conflict?. *e-Informatica Software Engineering Journal*, 1(1).

Gregor, S., & Hevner, A. R. (2013). *Positioning and presenting design science research for maximum impact*. *MIS quarterly*, 37(2).

Hevner, Alan R., March, Salvatore T., Park, Jinsoo, & Ram, Sudha. (2004). *Design science in Information Systems Research (1)*.(Research Essay). *MIS Quarterly*, 28(1), 75-105

Hüttermann, M. (2012). *DevOps for developers*. Berkeley, CA: Apress.

Johannesson, P., & Perjons, E. (2014). *An Introduction to Design Science* (Vol. 9783319106328). Cham: Springer International Publishing.

Jokela, T., Siponen, M., Hirasawa, N., & Earthy, J. (2006). A survey of usability capability maturity models: Implications for practice and research. *Behaviour & Information Technology*, 25(3), 263-282.

Kavis, M. (2014). *Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)* (Wiley CIO series.). John Wiley & Sons.

Kniberg, H. (2015). *Scrum checklist*. <https://www.crisp.se/wp-content/uploads/2012/05/Scrum-checklist.pdf> [2018-05-07]

Lasrado, L., Vatrapu, R., & Andersen, K. N. (2016). *A Set Theoretical Approach to Maturity Models: Guidelines and Demonstration*.

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2016). *An Exploratory Study of DevOps Extending the Dimensions of DevOps with Practices*. ICSEA 2016, 104.

Lwakatare, L., Karvonen, T., Sauvola, T., Kuvaja, P., Olsson, H., Bosch, J., & Oivo, M. (2016). *Towards DevOps in the Embedded Systems Domain: Why is It So Hard?* *System Sciences (HICSS)*, 2016 49th Hawaii International Conference on, 5437-5446.

Mohamed, S. I. (2015). *DevOps shifting software engineering strategy-value based perspective*. *International Journal of Computer Engineering*, 17(2), 51-57.

Poelwijk, S. (2016) *A maturity measurement model for devops teams*.
<https://github.com/LeaseWeb/devops-maturity-model> [2018-05-07]

Parks, J. (2016). *The Solinea DevOps Maturity Model Solinea* [blogg], 4 Oktober.
<https://solinea.com/blog/solinea-devops-maturity-model>

Rehn, A., Palmberg, T, Boström, P. (2013). *The Continuous Delivery Maturity Model InfoQ* [blogg], 6 februari. <https://www.infoq.com/articles/Continuous-Delivery-Maturity-Model#> [2018-04-12]

Riley, C. (2014) *How to Keep CALMS and Release More!* Logentrics [blog], 24 oktober.
<https://blog.rapid7.com/2014/10/24/how-to-keep-calms-and-release-more/> [2018-04-05]

Riley, C. (2015). *Metrics in DevOps* DevOps.com [blogg], 26 januari.
<https://devops.com/metrics-devops/> [2018-04-03]

Rong, G., Zhang, H., & Shao, D. (2016). *CMMI guided process improvement for DevOps projects: An exploratory case study*. *Proceedings - International Conference on Software and System Process, ICSSP 2016*, 76-85.

Sein, Maung K, Henfridsson, Ola, Purao, Sandeep, Rossi, Matti, & Lindgren, Rikard. (2011). *Action design research*. *Management Information Systems : Mis Quarterly*, 35(1), 37-56.

Shrestha, A., Cater-Steel, A., Toleman, M., & Tan, W. G. (2014, May). *Building a software tool for transparent and efficient process assessments in IT service management*. In *International Conference on Design Science Research in Information Systems* (pp. 241-256). Springer, Cham

Smeds, J., Nybom, K., & Porres, I. (2015). *DevOps: A definition and Perceived Adoption Impediments*. *Lecture Notes in Business Information Processing*, 212, 166-177.

CMMI Team, C. P. (2010). *CMMI® for Development, Version 1.3, Improving processes for developing better products and services*. no. *CMU/SEI-2010-TR-033*. *Software Engineering Institute*.

van Aken, J. 2004. "Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-tested and Grounded Technological Rules," *Journal of Management Studies*, (41) 2, pp. 219-246.

Venable, J., Pries-heje, J., & Baskerville, R. (2016). FEDS: A Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, 25(1), 77-89.

Venable, J. R., Pries-Heje, J., & Baskerville, R. (2017). Choosing a Design Science Research Methodology. In *ACIS2017 Conference Proceeding Hobart: University of Tasmania*.

Verona, J. (2016). *Practical DevOps*. Birmingham: Packt Publishing.

Wettinger, J., Breitenbucher, U., & Leymann, F. (2014). *Standards-Based DevOps Automation and Integration Using TOSCA. Utility and Cloud Computing (UCC)*, 2014 IEEE/ACM 7th International Conference on, 59-68.

Bilaga 1. InnovationLabs påståenden

Helhetsperspektiv
Dev och Ops har förståelse för varandras verksamheter.
Dev och Ops utvecklar och tillvaratar medarbetarnas kompetens.
Dev och Ops har kunskap om hur deras delar i processen bidrar till värde för kund.
Dev och Ops förstår hur interna och externa underleverantörers processer bidrar till kundvärde.
Dev och Ops förstår hur deras egna processer påverkar interna och externa underleverantörernas processer.
Dev och Ops tar ett helhetsansvar för tjänstens livscykel.
Det finns processägare som bevakar och styr våra processer. (diskutera vidare)
Dev och Ops tar tillvara sina resurser (människor, teknik och ekonomi) på ett optimalt sätt.
Samskapande av tjänst
Dev, Ops och kund identifierar gemensamt mål och värden som ska möjliggöras av tjänsten
Dev och Ops har gemensamt identifierat relevanta roller som krävs för att utveckla tjänster som möjliggör värde
Dev och Ops utvecklar tjänster gemensamt med övriga identifierade roller
Dev och Ops har gemensamma rutiner (tex möten) som stödjer att alla relevanta roller bidrar till tjänsteutveckling
Dev och Ops har nödvändiga mandat för tjänsteutvecklingen
Berörda roller granskar tjänster gemensamt och regelbundet
Dev och Ops identifierar gemensamt relevanta värdemått för tjänsten (t.ex. kritiska framgångsfaktorer eller KPIer)
Ops har en tillräcklig förståelse för Devs verksamhet
Devs har en tillräcklig förståelse för Ops verksamhet
Resurser
Dev och Ops identifierar gemensamt vilka kompetenser rollerna kan bidra med för att tillhandahålla tjänsten
Dev och Ops delar med sig av tjänsterelevant information och kunskap till andra intressenter
Dev och Ops informerar om problem och risker, som kan påverka leverans av tjänsten
Dev och Ops teammedlemmar är tillgängliga för varandra och andra intressenter
Dev och Ops utför omvärldsbevakning för att identifiera ny teknologi
Dev och Ops har ett gemensamt språk
Dev och Ops har en operationell modell som möjliggör delning av resurser
Ledningen har säkerställt att tillräckligt med resurser (kunskap, tid och verktyg) finns för att nå uppsatta mål
Automation
Dev och Ops använder digitala verktyg för att reducera fel vid releaser
Dev och Ops testar programkoden innan och efter releasen
Dev och Ops bygger in monitorering/övervakning av IT-tjänsten
Dev och Ops har en automatiserad uppsättning av infrastrukturkomponenter tillgängliga för projektteamen
Dev och Ops har digitala verktyg som möjliggör tjänsteinnovation
Det finns automatiserade mätningar för KPIer
Ständiga förbättringar
Dev och Ops tar gemensamt fram mätvärden

Framtagna mätvärden relaterar till tjänstens kritiska framgångsfaktorer
Framtagna mätvärden har en tydlig beskrivning, mål, mätprocess, och källa
Dev och Ops mäter regelbundet och rutinmässigt
Resultat av mätvärden tillgängliggörs för relevanta intressenter
Dev och Ops arbetar med ständiga förbättringar
Dev och Ops analyserar mätresultat tillsammans
Dev och Ops identifierar problem tillsammans
Dev och Ops identifierar lösningar tillsammans
Dev och Ops ger feedback om lösningar till alla intressenter
Dev och Ops har rutiner för kompetensutveckling

Högskolan i Borås är en modern högskola mitt i city. Vi bedriver utbildningar inom ekonomi och informatik, biblioteks- och informationsvetenskap, mode och textil, beteendevetenskap och lärarutbildning, teknik samt vårdvetenskap.

På **sektionen för informationsteknologi** har vi tagit fasta på studenternas framtida behov. Därför har vi skapat utbildningar där anställningsbarhet är ett nyckelord. Ämnesintegration, helhet och sammanhang är andra viktiga begrepp. På sektionen råder en närhet, såväl mellan studenter och lärare som mellan företag och utbildning.

Våra **utbildningar** med huvudområdet informatik är centrerade kring grundläggande begrepp som systemutveckling och verksamhetsutveckling. Inom vårt breda spektrum av inriktningar finns allt ifrån att programmera avancerade system, analysera behov och krav på verksamheter, till att bedriva integrerad IT- och affärsutveckling, dock med gemensamt syfte att verka för god IT-användning i företag och organisationer.

Vid sektionen bedrivs IT-relaterad **forskning** inom högskolans forskningsområde Handel & IT. Forskningsverksamheten är huvudsakligen ämnesmässigt inom **datavetenskap** respektive **systemvetenskap**. Speciella fokusområden är **data science** respektive **information systems science**. Forskningen är både vetenskapligt och professions-orienterad, vilket bland annat tar sig uttryck i att forskningen i många fall bedrivs med grund i domänspecifika verksamhetsbehov, med företag och offentliga organisationer på lokal, nationell och internationell arena. Forskningens professionsinriktning manifesteras också ofta genom vår delaktighet i Swedish Institute for Innovative Retailing (SIIR), som är en centrumbildning vid Högskolan med syfte att bidra till handelsföretag och det omgivande samhället med utveckling av innovativ och hållbar handel.



HÖGSKOLAN
I BORÅS

BESÖKSADRESS: JÄRNVÄGSGATAN 5 · POSTADRESS: ALLÉGATAN 1, 501 90 BORÅS
TFN: 033-435 40 00 · E-POST: INST.HIT@HB.SE · WEBB: WWW.HB.SE/HIT