



GÖTEBORGS UNIVERSITET

# **Beteendedriven utveckling i praktiken**

**En empirisk studie kring den praktiska tillämpningen av  
beteendedriven utveckling.**

## **Behavior Driven Development in practice**

**An empirical study on the practical application of behavior driven development.**

Robin Wilde  
Gustav Rydstedt

Kandidatuppsats i Informatik  
Rapport nr. 2016:015

## **Abstrakt**

Enligt traditionella principer skall mjukvaruutveckling ske i en linjär och välplanerad process där det först i de sista delarna av projektets livscykel genomförs tester på systemets funktionalitet. Agila utvecklingsmetoder blev mer populära under 2000-talets början där metoder med testfokuserad inriktning var inkluderade. Dan North publicerade under år 2006 en artikel där han presenterade sin metod Behavior Driven Development (BDD). Med en strukturerad utvecklingsprocess där ett gemensamt språk är kärnan för kommunikation ville Dan North öka förståelsen för mjukvaran för alla projektdeltagare i ett utvecklingsprojekt. Då metoden är relativt ny och den publicerade litteraturen är begränsad gällande användandet av BDD ställde vi oss frågan: Vilka faktorer behöver beaktas vid användandet av beteendedriven utveckling? Med hjälp av en kvalitativ studie med en induktiv ansats fick vi fram ett resultat som grundar sig i teori från publicerad litteratur och empiri från flera semistrukturerade intervjuer. Vårt resultat tydde på en skillnad i den praktiska tillämpningen av BDD än det vi fann i teorin. Anpassning av en metod kan ibland vara nödvändig men vi fann att vissa faktorer är relevanta att ta i beaktning vid användandet av BDD för att inte affärsnytta skulle gå förlorad.

**Nyckelord:** Beteendedriven utveckling, BDD, gemensamt språk, agil mjukvaruutveckling, affärsnytta

## **Abstract**

According to traditional software development principles, software development should be done in a linear and well planned process that, only in the last parts of the project life cycle, will conduct tests on the system's functionality. Agile development methods became more popular in the beginning of the 21st century where methods of testing-focused orientation was included. In 2006, Dan North published an article in which he presented a new method called Behavior Driven Development (BDD). With a structured development process which used a ubiquitous language as the core for communication, Dan North intended to increase the understanding of the software for all project participants in a development project. As this method is relatively new and the published literature is limited regarding the use of BDD we asked ourselves: What factors are of particular relevance for the use of behavior driven development? Using a qualitative study with an inductive approach, we received the results based on theory from published literature and empirical data from multiple semi structured interviews. Based on our results, the practical application of BDD differs from what we found in theory. Adapting of a method might sometimes be necessary, but we found that some factors are relevant to take into consideration when using BDD to prevent loss of business benefits.

**Keywords:** Behavior driven development, BDD, ubiquitous language, agile software development, business benefits

# Tack

Vi vill tacka “Bolaget” som gav oss möjligheten att genomföra vår fallstudie hos dem. Vi vill även rikta ett stort tack till vår handledare på “Bolaget” som ägnade mycket tid åt att planera intervjuer och att stötta oss i vårt arbete. Även ett stort tack till alla informanter som ville ställa upp och dela med sig av sina erfarenheter vilket låg till grund för vårt slutresultat.

Sist men inte minst vill vi rikta ett stort tack till vår handledare på Göteborgs Universitet, Aida Hadzic som har hjälpt oss från idéfas till en färdig kandidatuppsats.

# Innehållsförteckning

1. Inledning.....	1
1.1 Problemformulering.....	2
1.2 Syfte och frågeställning .....	3
1.3 Avgränsning .....	3
1.4 Disposition.....	3
2. Teori.....	4
2.1 Behavior Driven Development .....	4
2.2 Utvecklingsprocessen .....	5
2.3 Viktiga komponenter.....	8
2.3.1 Gemensamt språk .....	8
2.3.2 Automatiserade tester .....	8
2.3.3 Dokumentation.....	8
3. Metod .....	10
3.1 Metodval .....	10
3.2 Fallstudieobjekt - Projektet.....	10
3.3 Datainsamling .....	11
3.4 Urval .....	12
3.5 Presentation av urvalsgruppen .....	12
3.6 Dataanalys.....	13
3.7 Utvärdering av metod .....	13
3.8 Studiens upplägg.....	14
4. Resultat.....	15
4.1 Utbildning/upplärning.....	15
4.2 Utvecklingsprocessen .....	17
4.3 Gemensamt språk.....	18
4.5 Automatiserade tester.....	20
4.6 Dokumentation .....	21
5. Diskussion .....	23
6. Slutsats .....	28
6.1 Studiens relevans .....	28
6.2 Vidare forskning .....	28
Referenser.....	29

## Bilaga 1 - Intervjumall

## 1. Inledning

Mjukvaruutveckling är som disciplin en relativt ny företeelse när det kommer till mänskligt hantverk. Trots det har mjukvarusystem och informationsteknologi (IT) snabbt kommit att bli en självklar del av varje människas vardag och idag hanterar och påverkas vi av mjukvarusystem både i vardagen och i yrkeslivet. Den framtida vikten av IT uppmärksammades redan 1967 och svaret på det resulterade i organiseringen av den första mjukvaruutvecklingskonferensen år 1968 (Mens & Demeyer 2008). Konferensen organiserades av NATO för att sammanställa mjukvaruutvecklingsprinciper som skulle resultera i utveckling av mer tillförlitliga, effektiva och ekonomiskt lönsamma mjukvarusystem (ibid). En av insikterna, som senare kom att bli en av de centrala principerna inom mjukvaruutvecklingen, var att underhåll och rättningen av buggar skulle ske som sista del i en mjukvaruutvecklings livscykel (ibid). Ett resultat av denna åsikt lade grunden för den mycket populära mjukvaruutvecklingsmetoden, Vattenfallsmodellen, som år 1970 presenterades av Winston Royce i artikeln *Managing the development of Large Software Systems* (Larman & Basili 2003; Mens & Demeyer 2008). Vattenfallsmodellen är en linjär metod där varje del av ett projekt ska vara färdigt innan nästa får påbörjas. Testfas, underhåll samt rättningar av buggar sker först i slutet av projektets livscykel, i enlighet med riktlinjerna från Natokonferensen år 1968 (Stober & Hansmann 2010; Stoica, Mircea & Ghilic-Micu 2013).

Vattenfallsmodellens linjära process visade sig dock i många lägen vara problematisk. Modellens livscykel började med framtagandet av en fullständig kravspecifikation för all funktionalitet i mjukvaran och då krav på mjukvaran tenderade att förändras under projektets gång behövdes en anpassning (Stoica, Mircea & Ghilic-Micu 2013). Redan under mitten av 1900-talet fanns en förståelse för detta problem. Det finns exempel på IT projekt som jobbade med kortare iterationer för att kunna vara flexibel i sitt arbetssätt och svara på löpande förändringar (Nerur & Balijepally 2007; Greer & Hamon 2011). Det skulle dock dröja ytterligare en tid innan liknande arbetsmetoder på allvar började slå sig in på marknaden, metoder som inte fruktade förändringar utan snarare välkomnade dem. (Greer & Hamon 2011; Cao & Balasubramaniam 2007).

I samband med att dessa nya arbetsmetoder började bli allt vanligare tog 17 områdeexperter fram *The Agile Manifesto* år 2001 (Fowler & Highsmith 2001; Dingsøy, Nerur, Balijepally & Moe 2012). Syftet med manifestet var att lägga en gemensam grund i form av riktlinjer för att uppnå rörlighet i IT projekt med hjälp av de nya arbetsmetoderna. Med detta manifest som bakgrund började dessa rörliga mjukvaruutvecklingsmetoder kallas agila p.g.a. sin flexibilitet i form av att kunna anpassa sig till en konstant föränderlig miljö (Larman & Basili 2003). Målet med agila mjukvaruutvecklingsmetoder är att leverera värdefull och bättre mjukvara till kunden genom att acceptera och svara på uppkomna förändringar i ett mjukvaruutvecklingsprojekt (Fowler & Highsmith 2001).

En agil mjukvaruutvecklingsmetod som uppkommer är testdriven utveckling, på engelska Test Driven Development (TDD). Som namnet avslöjar är test en central del i arbetsmetodiken. TDD vänder på åsikten om att test och underhåll sker i slutet av ett mjukvaruutvecklingsprojekt som

presenterades som en princip på NATOS konferens 1968. Här skrivs alltid test innan skapandet av kod börjar (Janzen & Saiedian 2005). Testerna går senare att automatisera med olika verktyg för ständig felkontrollering av koden. Genom att ha automatiserade tester kan en mjukvaruutvecklare ändra i kod och genom testerna tidigt få reda på om någonting har kraschat (ibid). Beteendedriven utveckling, på engelska Behavior Driven Development (BDD), är en utveckling av TDD och presenterades i artikeln *Introducing BDD* av Dan North. North skapade metoden eftersom han kände att TDD ofta ledde till förvirring (North 2006). TDD använder sig av ett ostrukturerat språk i sina tester och testerna fokuserar på att verifiera ett tillstånd hos ett system snarare än att testa ett önskat beteende hos systemet (Solís & Wang 2011). North (2006) skriver att:

*“Programmers wanted to know where to start, what to test and what not to test, how much to test in one go, what to call their tests, and how to understand why a test fails.”*

I BDD används en tydlig struktur vid framtagandet av test och genom att benämna testet som ett beteende hos ett system menar North att det blev mycket tydligare vad som skulle testas (ibid). Metodens struktur med testbeskrivningar formulerade som beteenden gör det enklare att belysa ett affärsrelaterat problem som det angivna beteendet ska lösa. Strukturen använder sig av ett gemensamt språk för att skriva tester som gör det enklare för samtliga projektdeltagare, oavsett roll och teknisk erfarenhet, att referera till systemet på samma sätt (Soeken, Wille & Dreschler 2012; Diepenbeck, Soeken, Große & Dreschler 2012). Det är strukturen för testframtagning och det gemensamma språket som är kärnan i BDD och också det som bidrar till att projektdeltagare har enklare att kommunicera med varandra gällande ett system (Lazar, Motogna & Pârv 2010)

Det är med detta i fokus som allt fler har börjat titta mot BDD då metoden fokuserar på att utvecklarna, med intressenternas hjälp ska bygga rätt system istället för enbart bygga system rätt (Larsson 2008).

### **1.1 Problemformulering**

Det finns relativt mycket publicerad litteratur kring att använda sig av ett testinriktat fokus vid mjukvaruutveckling. Det finns dock ett par olika testinriktade metoder som i sin specifika tillämpning ämnar bidra till affärsnytta på olika sätt. BDD är en sådan metod och då den är relativt ny finner vi likt Solis och Wang (2011) att informationen är knapp kring den praktiska tillämpningen. Bristen av publicerad litteratur kan bidra med att egna tolkningar och anpassningar av metoden skapas så att de passar de förutsättningar och krav som verksamheten har. Verksamhetens egna tolkningar kan orsaka att arbetsprocesser skilda från, men snarlika, BDD skapas. Tolknings och anpassningar av en arbetsmetod är vanliga, och ibland nödvändiga, men de bör kritiskt granskas. Utan att kritiskt granska sitt användande av en arbetsmetod finns det risk att affärsnyttan drabbas negativt och metoden inte uppnår sin fulla potential. Det är med detta i beaktning som vi genomför vår studie.

## **1.2 Syfte och frågeställning**

Syftet med denna uppsats är att undersöka vilka faktorer som behöver beaktas vid användandet av mjukvaruutvecklingsmetoden BDD. Vi avser att studera den praktiska tillämpningen av metoden i en fallstudie och granska den publicerade litteratur som finns.

Med det som bakgrund formulerar vi vår forskningsfråga enligt följande:

*Vilka faktorer behöver beaktas vid användandet av beteendedriven utveckling?*

Målet med studien är att belysa vilka faktorer som behöver beaktas vid användandet av BDD. Studien kan således användas av både verksamheter som använder sig av BDD idag och verksamheter som ämnar tillämpa metoden i framtiden. Vår förhoppning är att de med ökad förståelse för dessa faktorer kan öka affärsnyttan i användningen av BDD och även minska avståndet mellan teori och praktisk tillämpning.

## **1.3 Avgränsning**

Studien kommer enbart fokusera på vad teorin säger om användandet av BDD och senare studera hur dessa används i den praktiska tillämpningen av metoden. Vi vill belysa de faktorer som teorin upprepat nämner som relevanta vid implementation och användandet av metoden för att uppnå den tänkta affärsnyttan. Följaktligen har ingen större hänsyn tagits till varför viss problematik finns vid användandet av metoden utan vi vill snarare hjälpa till med att belysa generella faktorer som behöver beaktas vid den praktiska tillämpningen av BDD. Vår förhoppning är att kunna bidra med ett resultat som är överförbart på andra verksamheter.

Vi har valt att avgränsa till två projekt inom samma bolag för att studera det ena mer djupgående och sen använda det andra projektet som referens. Båda projekt använder sig dock av BDD dagligen och projektdeltagarna som intervjuats har en god uppfattning om metoden och därför finner vi studien trovärdig.

## **1.4 Disposition**

Kapitel 2 presenterar den insamlade teorin som har legat till grund för studien. Vidare ges en övergripande förklaring av BDD som metod, en förklaring av vilken affärsnytta som metoden ämnar att skapa, metodens utvecklingsprocess samt viktiga komponenter för metoden. Kapitel 3 redogör för vilken forskningsmetod som har använts vid insamlandet av empiri, syftet med metodvalet och kapitlet avslutas med en metodutvärdering. I kapitel 4 behandlas det empiriska resultat som insamlats med hjälp av den valda forskningsmetoden. Resultatet presenteras genom en strukturering som följer teoridelens kapitel. Kapitel 5 innehåller en diskussion kring resultatet som ämnar att knyta ihop teori och det insamlade resultatet genom tolkningar över vad som har samlats in. Kapitel 6 innefattar en slutsats där vi avser besvara forskningsfrågan, presentera sammanfattade reflektioner av undersökningen, belysa studiens relevans och till sist följer förslag till vidare forskning.



## 2. Teori

Nedanstående teoridel börjar med en övergripande presentation av beteendedriven utveckling, på engelska behavior driven development (BDD), som metod och hur metoden ska användas. Därefter följer en mer detaljerad beskrivning av utvecklingsprocessen vid framtagandet av BDD. Vidare ger vi en djupare förklaring till de komponenter som är viktiga att ta i beaktning i användandet av BDD.

### 2.1 Behavior Driven Development

BDD är en agil mjukvaruutvecklingsmetod som presenterades av Dan North i artikeln *Introducing BDD* (2006). Metoden tillhör den gren av agila mjukvaruutvecklingsmetoder som har en testfokuserad inriktning. BDD börjar sin livscykel med att skriva tester innan själva programmerande av källkod tar vid (North 2006; Diepenbeck et al. 2012). Därefter refaktoreras koden, som innebär att omstrukturera kod för att förbättra eller korrigera den (Demeyer 2005), och kontinuerlig implementation av ny kod utförs (North 2006; Solís & Wang 2011). BDD är i sin tur en utveckling av metoden testdriven utveckling, på engelska test driven development (TDD) (North 2006; Solís & Wang 2011). BDD bygger på samma principer som TDD där syftet är att genom automatiserade acceptanstester underlätta arbetet för utvecklare då källkoden testas innan själva mjukvaruutvecklingen tar vid (Solís & Wang 2011; Soeken, Wille & Dreschler 2012). Automatiserade acceptanstester ger utvecklaren modet och tryggheten att göra ändringar i koden då testerna direkt påvisar eventuella felaktigheter (Janzen & Saiedian 2005). North ansåg att användandet av TDD ofta skapade förvirring. North (2006) skriver:

*“Programmers wanted to know where to start, what to test and what not to test, how much to test in one go, what to call their tests, and how to understand why a test fails.”*

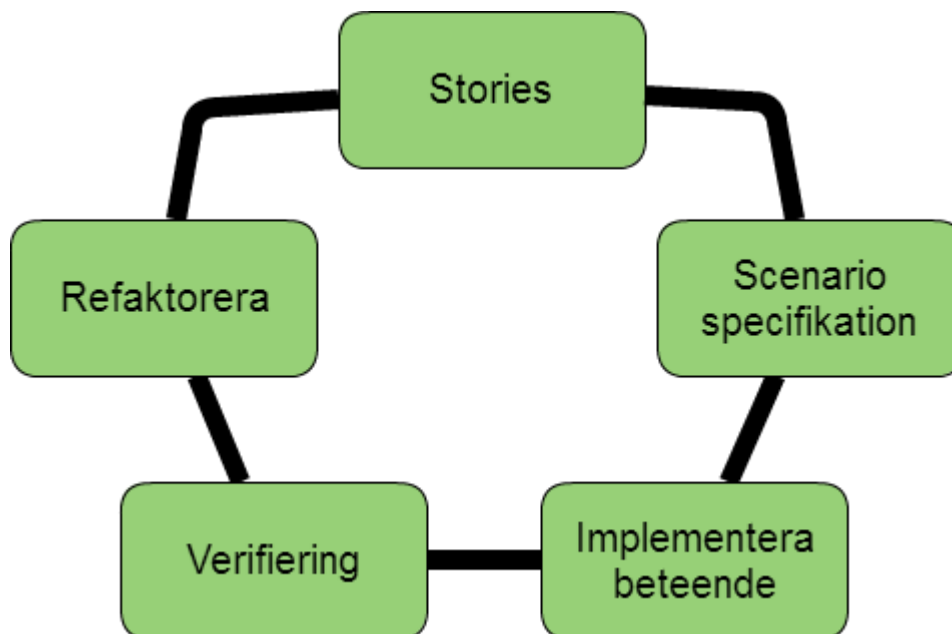
Problematiken hos TDD är att metoden testar ett tillstånd hos ett system snarare än ett förväntat beteende (Solís & Wang 2010). North (2006) fann att vid omskrivning av tester så de hänvisar till ett förväntat beteende hos systemet, snarare än ett tillstånd, var resultatet att testerna enklare att förstå för utomstående.

BDD använder sig av en tydlig struktur för testframtagning och med hjälp av ett gemensamt språk blir det enklare att koppla kravställning till “stories” som i sin tur översätts till ett “scenario”. Scenarion översätts till sist till ett körbart test för en viss funktionalitet i systemet. Det gemensamma språket är kärnan i BDD (North 2006; Lazar, Motogna & Pârv 2010). Det är det gemensamma språket som gör det möjligt att tydliggöra funktionalitet hos systemet för samtliga projektdeltagare och därmed minska avståndet mellan affärs- och teknikmänniskor. Alla projektdeltagare ska prata om och hänvisa till systemet på samma vis (Lazar, Motogna & Pârv 2010). Tanken är att projektgruppen redan vid analysfasen i mjukvaruutvecklingens livscykel ska bestämma ett gemensamt språk i form av en ordlista som ska vara ett komplement till det gemensamma språk som används i framtagningen av test. Dock kan ord tillkomma i ordlistan under projektets gång om det behövs (Solís & Wang 2011). Missförstånd mellan olika roller i projektet kan med hjälp av det gemensamma språket elimineras. Det gemensamma språket omfattas både i form av ordlistan och också strukturen som BDD innehar i form av

bestämda regler för att skriva stories och scenario som senare kopplas till de automatiserade testerna (North 2006).

Vid framtagandet av krav på funktionalitet för ett system ska BDD användas och dessa krav ska beskrivas med hjälp av stories som beskriver syftet med en viss funktionalitet (Solís & Wang 2011). Genom att det redan i framtagandet av krav används BDD struktur lyfts affärsvärdet för en specifik funktionalitet fram. Genom BDD:s struktur på stories lyfts även vem eller vilken roll som kommer tjäna på funktionaliteten i systemet. BDD ska hjälpa till att skapa en djupare förståelse för systemet hos samtliga roller (North 2006). Stories som representerar krav på funktionalitet i systemet kan senare översättas till scenarios av en analytiker eller utvecklare. Detta för att klargöra de specifika beteenden som kan uppstå vid användandet av den beskrivna funktionaliteten hos systemet (Lazar, Motogna & Pârv 2010). Både stories och scenarion ska skrivas med det bestämda gemensamma språket d.v.s. både i form av den mall som BDD använder och de termer som bestämts inom projektet (North 2006; Solís & Wang 2011). Det är det gemensamma språket som underlättar kommunikationen inom ett projekt och bidrar till ökad förståelse för systemet hos alla parter av projektet oavsett roll eller teknisk erfarenhet (Soeken, Wille & Dreschler 2012). Nästa steg i utvecklingsprocessens livscykel blir att scenarion ska översättas till automatiserade acceptanstester som stöttar mjukvaruutvecklare i sitt programmerande (North 2006). Metoden syftar till att alla projektdeltagare ska vara involverade i någon del av framtagandet av BDD-testerna. Eftersom samtliga roller i projektet har använt sig av en gemensam struktur och ett gemensamt språk kan BDD användas som dokumentation av systemet och dess funktionalitet då det beskriver vilka krav som finns på systemet (North 2006; Solís & Wang 2011).

## 2.2 Utvecklingsprocessen



*Figur 1: BDD:s utvecklingslivscykel Källa: Egen.*

## Stories

Användandet av BDD börjar med att krav över funktionalitet skrivs i berättelser. Dessa berättelser kallas stories och ska beskriva vad ett system ska göra. Stories skrivs med en bestämd mall och med det gemensamma språk som BDD bidrar med (Solís & Wang 2011). Stories ska beskriva vilken roll som är tänkt som användare samt vilken nytta den bestämda funktionaliteten ska hjälpa systemet med. Därmed blir det enklare i framtiden för en utvecklare att förstå vilken funktionalitet som ska utvecklas och med vem som eventuella funderingar kring funktionaliteten kan diskuteras då användaren är angiven (ibid). Genom att följa BDD:s struktur med stories redan vid kravanalysen underlättas arbetet senare i mjukvaruutvecklingsprocessen genom att fler personer i projektet är inkopplade i en djupare förståelse av systemet (North 2006).

Stories ska även innehålla en titel som beskriver det förväntade beteendet för en viss funktionalitet (North 2006). Med hjälp av stories behöver de som tar fram den fundera på affärsvärdet med funktionen och därigenom behöver de även överväga om funktionen är absolut nödvändig (Solís & Wang 2011). Lazar, Motogna och Pårv (2010) skriver att varje system ska ha ett verifierbart affärsvärde vilket BDD:s struktur hjälper till att tydliggöra.

I mallen för att ta fram stories finns termerna “As a”, “I want” och “so that”, på svenska “Som en”, “vill jag kunna” och “så att jag kan göra/få”, vilka är exempel på termer från det gemensamma språket BDD använder sig av i sin struktur.

Nedan visas mallen för hur en story ska skrivas enligt BDD:s struktur:

[Story-titel] (En mening som beskriver historien)

**Som en** [Roll]

**Vill jag kunna** [Funktion]

**Så att jag kan göra/få** [Nytta]

Exempel på en story som beskriver en inloggningsprocess för ett system:

Titel: Användare loggar in i systemet

**Som en** användare

**Vill jag kunna** logga in med användarnamn och lösenord

**Så att jag kan** komma åt systemet.

## Scenario-specifikation

Nästa steg i metodens livscykel är att omvandla storyn till ett eller flera scenario-specifikationer. Ett scenario syftar till att beskriva hur ett system ska bete sig vid ett specifikt tillstånd och när en viss handling genomförs (Solís & Wang 2011). Antalet scenarion speglar antalet beteenden som systemet kan ha beroende på antalet utfall som kan uppstå vid olika handlingar hos en användare (North 2006). Scenarion skrivs enligt strukturen “Given”, “When” och “Then” som även de tillhör det gemensamma språket för BDD (North 2006). På svenska

översätts det till “Givet”, “När” och “Då”. Även här ska det finnas en titel som beskriver scenariot.

Nedanstående två scenario-exempel bygger på ovanstående story-exempel och beskriver två potentiella beteenden hos ett system:

Scenario 1: Användarnamn och lösenord är korrekt

**Givet:** Användarnamn och lösenord är korrekt

**När:** Användaren anger användarnamn och lösenord och klickar på knappen “logga in”

**Då:** Loggas användaren in

Scenario 2: Användarnamn och/eller lösenord är inkorrekt

**Givet:** Användarnamn och/eller lösenord är inkorrekt

**När:** Användaren anger användarnamn och lösenord och klickar på knappen “logga in”

**Då:** Felmeddelande visas

Som ovanstående exempel visar ska scenariot konkretisera olika beteenden hos ett system. Dessa scenarion baseras utifrån den story som ligger till grund för den funktion som ska utvecklas (Solís & Wang 2011). Antalet scenarion för en viss story varierar därför och antalet är beroende på hur många scenarion som krävs för att täcka in alla eventuella beteenden som systemet kan ha i en viss funktionalitet (North 2006). Därför gäller det att vara specifik vid framtagandet av scenario-specifikationer och följa den givna mallen med det bestämda gemensamma språket för att underlätta för utvecklingsarbetet (ibid).

### **Implementera beteende**

Efter att scenariot är skrivet skall BDD:erna implementeras i koden och detta görs på olika sätt beroende på vilka verktyg som används (Taveres, Rezende, Mota dos Santos, Manhaes & Atem de Carvalho 2010). Det är i denna fas testerna som går över från att vara kravspecifikationer av funktionalitet hos systemet till att bli automatiserade acceptanstester av beteendet hos systemet (North 2006). Genom att kunna implementera dessa scenarion till automatiserade tester skapas en trygghet och ett mod hos systemutvecklare att ändra i källkod. Detta då testerna ger kontinuerlig information genom att exempelvis ett felmeddelande som visas om något har ändrats inte går igenom testet (Janzen & Saiedian 2005).

### **Verifiering**

Testerna bidrar senare till en verifiering av beteenden hos systemet. Beroende på vilka verktyg som används så visas resultatet av ett test på olika sätt. Oavsett verktyg visas någon form av resultat som påvisar om beteendet lyckades eller misslyckades i systemet. Först när alla test av ett system är klara är systemet helt färdigt (Soeken, Wille & Drechsler 2012). Verifiering innebär alltså att de körbara testerna verifierar de olika beteenden hos ett system snarare än tillståndet av systemet som är fallet i TDD (Solís & Wang 2011).

## **Refaktorera**

Refaktorering innebär att koden korrigeras för att förbättras. I BDD kan det handla om att resultatet av ett test är misslyckat och således behöver testet rättas eller utvecklas så att testet av beteendet går igenom. Det kan även handla om att ett nyligen framtaget test går igenom men påverkar en annan del av systemet vilket resulterar att något annat behöver korrigeras. Beteendetestet kan alltså lyckas men att det i sin tur påverkar ett annat beteende negativt som kan medföra ett behov av refaktorering (Soeken, Wille & Drechsler 2012).

### **2.3 Viktiga komponenter**

Nedan beskrivs de komponenter som i litteraturen lyfts fram som viktiga vid användandet av BDD som mjukvaruutvecklingsmetod. Komponenterna i sig är inte unika utan det är flera metoder som använder sig av dem. Det som gör BDD unikt är hur de ska användas tillsammans för att bidra till ökad affärsnytta för projekt som väljer att använda sig av BDD som mjukvaruutvecklingsmetod.

#### **2.3.1 Gemensamt språk**

Gemensamt språk, på engelska “ubiquitous language”, har en central roll i BDD och är ett begrepp som används inom olika områden. James Shore och Shane Warden (2007, 125) skriver: “*We understand each other*”. De menar att med ett gemensamt språk i en projektgrupp skapas en övergripande förståelse, oavsett roll eller teknisk erfarenhet, för vad systemet ska göra och vilket värde det har för användaren.

Kärnan i BDD handlar om att genom ett gemensamt språk kunna tillhandahålla en struktur för kommunikation inom projektet (North 2006; Solís & Wang 2011). Kommunikation i detta fall handlar både om BDD:s struktur med bestämda mallar för att utforma stories och scenarion och att bestämma gemensamma termer för att referera till systemet. Genom att använda ett naturligt språk i stories och scenarion ökar förståelsen för utvecklingen av systemet oavsett vilken roll eller teknisk erfarenhet projektdeltagaren har (Soeken, Wille & Dreschler 2012). Därför är det med hjälp av det gemensamma språket i alla delar av mjukvaruutvecklingsprocessen som BDD ska hjälpa till att projektgruppen, oavsett roll eller teknisk erfarenhet, har enklare att förstå varandra och missförstånd ska elimineras (Lazar, Motogna & Pårv 2010; Soeken, Wille & Dreschler 2012).

#### **2.3.2 Automatiserade tester**

BDD är en av ett antal mjukvaruutvecklingsmetoder som använder sig av automatiserade acceptanstester (Solís & Wang 2011). Dessa tester hjälper till med en verifiering av det förväntade beteendet hos systemet samt att de är skrivna enligt den specifikation som har upprättats genom scenariona. Då testerna körs kontinuerligt bidrar det med en trygghet hos utvecklarna att genomföra ändringar i koden då det snabbt meddelas om något har blivit fel (Janzen & Saiedian 2005).

#### **2.3.3 Dokumentation**

I “The agile manifesto” (2001) som togs fram av 17 områdesexperter beskrivs det hur en agil arbetsmetodik ska värdesätta “Fungerande programvara framför omfattande dokumentation”.

BDD som agil arbetsmetodik understödjer detta då metoden med sin tydliga struktur och sitt gemensamma språk bidrar med dokumentation under hela utvecklingsprocessen. Detta då de bestämda mallar för utformandet av stories och scenarion ligger till grund för den funktionalitet som systemet och följaktligen kan ses som dokumentation av systemet (Atem de Carvalho, Luiz de Carvalho e Silva & Soares 2010). Alla delar i koden som klasser, metoder och objekt ska vara självförklarande i koden vilket ska följa projektets uppsatta regler kring det gemensamma språket (Solís & Wang 2011). Genom att kravanalysen och testningen går in i varandra med hjälp av BDD minskar dokumentationen och alla steg i processen blir lätta att följa och återkoppla till. Med hjälp av BDD blir dokumentationen av krav mindre missvisande och utvecklarna vet direkt vad som ska göras och varför och det sker då färre missförstånd mellan olika roller (North 2006). Minskat dokumentationsarbete ökar tiden till att faktiskt utveckla och leda utvecklingen av systemet framåt något som sker med användandet av BDD (Solís & Wang 2011).

### **3. Metod**

Syftet med vår studie är att med hjälp av en kvalitativ och deskriptiv ansats undersöka hur verksamheter använder sig av mjukvarututvecklingsmetoden BDD. För att skapa oss en bättre förståelse började vi med att göra en förstudie med hjälp av ett studiebesök där en presentation hölls. Denna förståelse bidrog till att vår datainsamling i form av semistrukturerade intervjuer underlättades. Genom en kvalitativ studie menar Patel och Davidson (2011) att en djupare förståelse bildas av problemområdet.

#### **3.1 Metodval**

Vår kvalitativa studie utförs i ett IT projekt bestående av fem till tio personer. IT projektet, senare i texten kallat "Projektet", genomförs på ett större IT bolag i Göteborg, senare i texten kallat "Bolaget". I Bolaget har vi även genomfört en intervju med en person i ytterligare ett projekt, senare i texten kallat "Referensprojektet". Därmed får vi ytterligare uppfattning om hur Bolaget använder sig av metoden och även ytterligare data om hur den praktiska tillämpningen av BDD ser ut.

Vi har valt en kvalitativ ansats då fokus ligger på att skapa sig en djup förståelse kring hur BDD används i Projektet. Målet är att skapa ett djup i studien snarare än en bredd (Klein & Myers 1999; Patel & Davidson 2011). För att genomföra studien har vi tagit fram en teoretisk bakgrund och använt oss av semistrukturerade intervjuer som spelades in. Intervjuerna ska hjälpa till med ny data till den befintliga teorin (Walsham 1995). Vi valde att ha hög grad av standardisering i vår intervjumall d.v.s. att vi hade ett antal teman som vi ansåg vara viktiga att beröra i intervjun för högsta möjliga kvalitet på vårt analysmaterial (Patel & Davidson 2011). Genom att sammanställa det insamlade materialet med transkribering och analyser kunde vårt syfte besvaras (Patel & Davidson 2011).

Vi valde att börja vår studie med en förstudie på plats på Bolaget för att skaffa oss en bredare förståelse för hur deras arbete ser ut. Det gav oss en inblick i vad det är för typ av mjukvara de utvecklar i Projektet och på vilket sätt som BDD används i Projektet. Därmed hade vi ett bättre underlag för att genomföra våra semistrukturerade intervjuer (Patel & Davidson 2011).

#### **3.2 Fallstudieobjekt - Projektet**

Vår fallstudie har genomförts i två projekt som är verksamma i Bolaget där fokus ligger på utvecklingsmetoden BDD. Fokus lades på Projektet då de hade använt metoden under en längre tid och flera intressanta informanter fanns att tillgå. Referensprojektet som är relativt nystartat fungerade som en referens till hur BDD används i Bolaget och bidrog till ett djup i fallstudien. Bolaget är en större offentlig organisation som är verksam i hela Sverige och deltagarna i Projektet sitter i Göteborg och består av fem till tio personer. Projektet startade 2011 och deras applikation driftsattes två år senare. Mjukvaran som de utvecklar är en applikation som hämtar data från andra interna system, d.v.s. samlar information från många system på ett ställe. Det ställer höga kvalitetskrav på applikationen.

### 3.3 Datainsamling

Vid datainsamlingen av vårt empiriska material använde vi oss av semistrukturerade intervjuer samt dokumentanalyser. Intervjuerna spelades in för att sedan transkriberas, så att en noggrann analys av materialet kunde genomföras. Nedan följer en detaljerad beskrivning hur vår datainsamling gick till.

För att öka och ytterligare fördjupa vår förståelse kring arbetssättet och den aktuella projektgruppens arbete gjordes en förstudie hos Bolaget. Projektmedlemmarna introducerade oss i deras projekt och förklarade övergripande hur deras arbetsprocess såg ut. De nämnde vilka verktyg som användes och vilka olika metoder som används för att genomföra sitt arbete. I samband med presentationen berättade de även om hur de använder sig av BDD och vilka utmaningar som de har stött på med metoden.

Efter förstudien studerades tidigare forskning och teori kring BDD samt viss annan litteratur som behandlade andra former av mjukvaruutveckling. Fokus låg på att identifiera teoretiska faktorer som behöver beaktas gällande användandet av mjukvaruutvecklingsmetoden BDD. Detta för att sedan kunna studera hur eller om dessa används i praktiken på Bolaget.

När bättre förståelse i ämnet var införskaffat började ett arbete med att ta fram en intervjumall med relevanta och utvecklande frågor. Målet var att intervjua fem till tio personer och resultatet blev att vi intervjuade fem personer. Med en semistrukturerad intervjuform tog vi hjälp av en förutbestämd mall med teman. Dessa innehöll ett antal öppna frågor som sedan utvecklades under samtalet (Patel & Davidson 2011). Därmed hade intervjun en hermeneutisk inriktning där det är upp till oss som forskare att hitta följd- och fördjupande frågor som blir intressanta för den helhetsbild vi försöker bilda oss (Patel & Davidson 2011). Målet med denna typ av datainsamling är att skaffa sig en så bra och djup bild som möjligt av hur medarbetarna i Projektet och Referensprojektet känner för metoden BDD och användningen av den.

Vi var vid samtliga intervjuer två personer som höll i intervjun och alla fem intervjuer utfördes på Bolaget och varade i ungefär 60 minuter. På så vis kunde en av oss fokusera på att föra samtalet framåt i intervjun och den andre föra enklare anteckningar och även lägga till fördjupande frågor under intervjuns gång. Intervjuerna spelades in med hjälp av inspelningsprogram på en mobiltelefon samt på en dator för att sedan transkriberas. Samtliga intervjupersoner gav oss sitt medgivande innan inspelningen började att det var okej att spela in. Inspelning användes för att senare kunna analysera informanternas exakta svar som i sin tur bidrar till ett enklare analysarbete av den insamlade datan (Patel & Davidson 2011). Genom inspelning av intervjun blev samtalet mer flytande och fokus kunde läggas på att ställa intressanta följdfrågor istället för att anteckna (Davidson & Patel 2011; Ranerup 2016). Med öppna frågor är chansen större att informanten skulle vara ärlig och öppen i sina svar samt ha möjlighet till egna tankar och funderingar. Inför varje intervju blev informanterna informerade om att studien lämpar sig för att förbättra och utveckla arbetet med BDD och detta påverkade förhoppningsvis att personalen blev mer tillmötesgående och villiga att dela med sig av information. Med hjälp av transkriberingen blev det enklare för oss att i efterhand analysera intervjuerna och kunde därmed hitta viktiga fokuspunkter för studien.



### 3.4 Urval

Det finns flera anledningar till att Projektet ansågs vara intressant för vår studie. Deltagarna i Projektet ansåg själva att de hade en del problem med utförandet av metoden och de ville, med vår objektiva bild av Projektet, ha möjlighet att undersöka vilka eventuella förbättringsåtgärder som fanns. Kompetens som anskaffades vid uppstart av Projektet har delvis gått förlorad då flera medarbetare har slutat under projektets gång samt att de ansåg sig själva vara lite låsta i sitt genomförande. Samtidigt har Bolaget en tro på BDD och dess fördelar och en vilja och ambition att utveckla metoden för att i framtiden tilldela BDD till fler projekt. I Bolaget verkar Referensprojektet som med sin kunskap om BDD kan understödja med ytterligare information av hur metoden används i praktiken. Detta resulterade i en god matchning mellan vårt syfte med uppsatsen och Bolagets ambitioner.

Vår förhoppning var att få möjlighet att intervjua:

- 1) personer som var med vid uppstart av Projektet, helst personen eller gruppen som initierade användandet av metoden.
- 2) Personer som idag jobbar i Projektet med BDD och då gärna personer i flera olika ansvarsområden eller roller i projektgruppen. Exempelvis utvecklare, kravställare, testare och projektledare.
- 3) Personer som tillhör andra projektgrupper där metoden är bestämd att vara BDD. På så vis får vi ett ytterligare djup i den praktiska tillämpningen av BDD.

Målet är att få med intervjupersoner med olika erfarenheter av BDD och som innehar olika arbetsroller i projektgrupper för att sålunda få med så många aspekter som möjligt. Till vår hjälp att få tag på rätt personer att intervjua hade vi en handledare på Bolaget som vi hade kontakt med under hela studien. Resultatet av våra intervjuer blev i det närmaste som vi önskat. Sett till antalet informanter och att informanterna hade olika erfarenheter av BDD blev det som vi önskade. Dock fick vi inte med så många olika roller vi hade förhoppningar på.

### 3.5 Presentation av urvalsgruppen

Nedan följer en kort presentation av de deltagande informanterna, då studien är anonymiserad kommer dessa endast presenteras väldigt enkelt.

#### **Informant 1:**

Utvecklare som var med vid uppstart av Projektet och införandet av BDD. Arbetar inte kvar i projektet sedan 1,5 år tillbaka. Stor erfarenhet av andra IT projekt.

#### **Informant 2:**

Konsult som utvecklare, har jobbat med Projektet i 1 år. Stor erfarenhet från IT andra projekt.

#### **Informant 3:**

Utvecklare som varit med i Projektet i 2,5 år. Viss erfarenhet av andra IT projekt.

#### **Informant 4:**

Utvecklare som varit med i Projektet i 6 månader. Viss erfarenhet av andra IT projekt.

### **Informant 5:**

Testledare och kravanalytiker i Referensprojektet som också använder sig av BDD. Stor erfarenhet av andra IT projekt.

Vi anser att med dessa informanter kunna skapa oss en bra bild över hur den praktiska tillämpningen av BDD ser ut samt att vi får en bra blandning av erfarna och mindre erfarna mjukvaruutvecklare. Vi ser också nytta i att ha med en informant som jobbar i samma Bolag med samma mjukvaruutvecklingsmetod, BDD, men som inte jobbar i Projektet. Detta bidrar till en bredare bild då det går att använda data i form av referens.

### **3.6 Dataanalys**

Vi har använt oss av kvalitativa metoder vid analysarbetet av data som samlats in under intervjuerna. Kvalitativa metoder har inte några färdiga mallar för hur analysarbetet ska utföras utan forskaren tillämpar själv sin egen metod utifrån vad som framkommer av analysen (Patel & Davidson 2011). Målet var dock att finna mönster i informanternas svar och således skapa sig en så god bild som möjligt av deras användande av BDD.

Intervjuerna transkriberades direkt efter intervjutillfällena och sammanställdes i ett dokument där vi sedan analyserade svaren på frågorna. De svaren vi ansåg vara intressanta för vår studie fördes sedan in i ett nytt dokument där vi samlade intressanta citat. När vi samlat alla citat i ett dokument skrev vi ut dem och klippte ut varje citat. Sedan grupperade vi citaten i relevanta högar med teman. Löpande sållades även vissa citat bort som inte var relevanta för vårt vidare analysarbete. Genom att analysera texten fysiskt skaffade vi oss en bra och god överblick över vårt insamlade material (Patel & Davidson 2011). Under vårt analysarbete fokuserade vi på de komponenter vi funnit i teorin om BDD. Vi sammanställde ett antal faktorer som vi tolkade som viktiga när det gäller att lyckas med BDD. Faktorer valdes ut med stöd av den teori vi läst gällande BDD samt av att vi fann mönster i svaren vi fick från informanterna.

### **3.7 Utvärdering av metod**

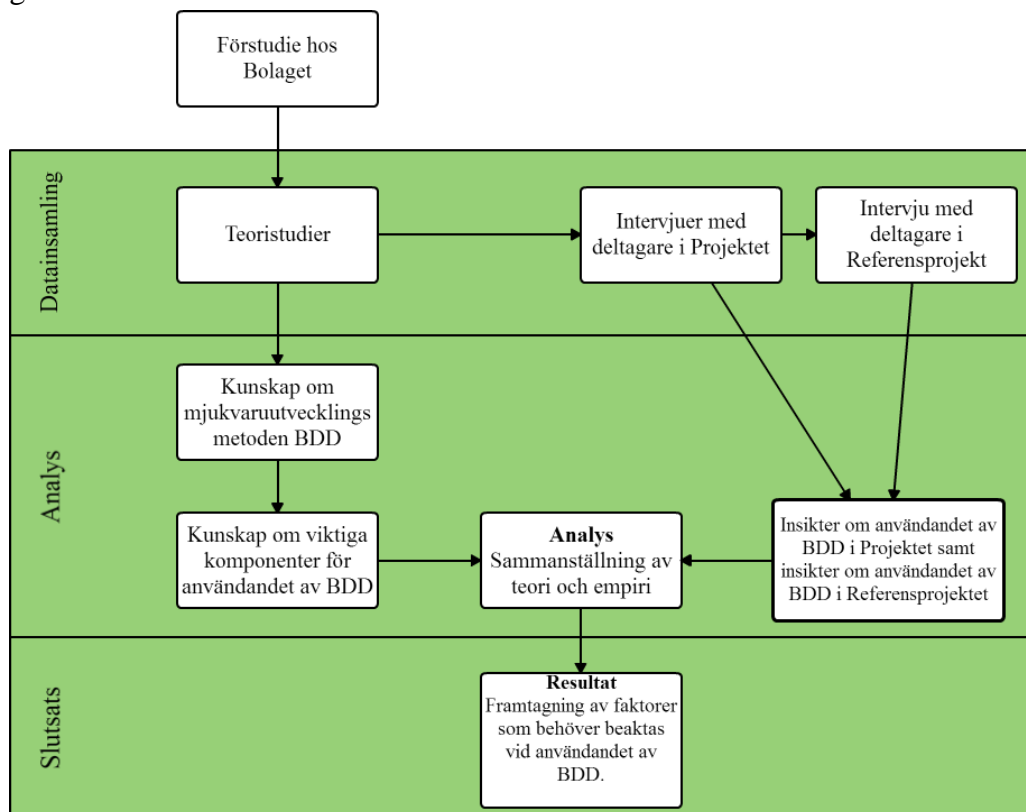
Det finns inga rätt eller fel i hur en kvalitativ studie skall genomföras utan istället bör metoden anpassas efter situationen runt dess förutsättningar (Patel & Davidson 2011). Nackdelarna med att använda sig av en kvalitativ studie, där analys och tolkning är viktiga delar, kan ligga i att oerfarna forskare inte har en utvecklad hermeneutisk förmåga och med det kan viktig information gå förlorad vid analysarbetet. Oerfarenhet vid intervjutillfället kan bidra till att forskarna har problem i att hålla sig neutrala och därigenom färga slutresultatet. Risken är att kvalitén på studien påverkas negativt (Walsham 1995; Davidson & Patel 2011). En annan risk är att informanterna blir påverkade av att intervjun spelas in och blir oärliga och stressade i sina svar (Walsham 2006). Detta är något vi försökte undvika med att informera informanterna att vårt mål med studien är att utveckla och förbättra deras arbete med BDD. Möjligheten att inte få tag på tillräckligt många informanter kunde också varit ett potentiellt hot mot studiens genomförande (Davidson & Patel 2011).

För analysarbetets skull finner vi att det möjligtvis hade varit bättre att genomföra fler intervjuer och ha mer insamlad data. Dock var det både av tids- och tillgänglighetsmässiga skäl som det stannade på fem personer. Genomförandet av intervjuer är en tidskrävande process där planering av intervju, intervjutillfälle, transkribering och sedan analys av data ingår. Vi önskade också att intervju en kravanalytiker i Projektet men det var inte möjligt. Det skulle kunna bidra till en ensidig bild av användandet av BDD i Projektet vilket kan ha påverkat resultatet negativt. Vi lyckades dock få en intervju med en kravanalytiker i Referensprojektet vilken bidrog med sin erfarenhet av användandet av BDD. I efterhand ser vi att det hade kunnat vara bra att planera in ytterligare en intervju med en person i Referensprojektet för att skapa oss en bredare förståelse i hur metoden används i specifikt deras projekt. Med bara en intervju finns det risk att uppfattningen blir färgad av endast en persons åsikter (Davidson & Patel 2011).

Projektet bestod bara av fyra heltidsdeltagare när vår studie genomfördes och de kunde med sin erfarenhet delge oss en bra bild av användandet av BDD i Projektet. Vi anser även att intervjun med personen ur Referensprojektet var värdefull då personen innehar mycket erfarenhet av andra projekt samt har en central roll i Referensprojektet. Därmed kunde personen bidra med en helhetsbild av användandet av BDD. Därför anser vi att intervjuer med fyra informanter som antingen har verkat eller är verksamma i Projektet samt en intervju med en person ur Referensprojektet gav oss en bra bild av hur BDD används i Bolaget.

### 3.8 Studiens upplägg

Nedan visas en modell över studiens upplägg. Detta för att ytterligare förtydliga hur studien har gått till.



**Figur 2:** Modell över studiens upplägg *Källa: Egen.*

## 4. Resultat

Vår undersökning har resulterat i flera reflektioner om användandet av BDD och hur metoden används i Projektet samt i Referensprojektet. De mest centrala delarna av vårt empiriska material som tagits fram under de semistrukturerade intervjuerna kommer att presenteras i form av citat och reflektioner. Inledningsvis presenteras hur metoden har implementerats samt hur upplärningen av nya deltagare i Projektet och Referensprojektet ser ut. Efter det kommer mjukvaruutvecklingsprocessen med BDD presenteras och därefter följer en redogörelse för användandet av de komponenter som nämnts i teoridelen. Dispositionen är densamma som i teoridelen, d.v.s: gemensamt språk, automatiserade tester och dokumentation. I varje del presenteras resultatet från Projektet först (Informant 1 till 4) efter det presenteras resultatet från Referensprojektet (Informant 5).

### 4.1 Utbildning/upplärning

Utbildning- eller upplärningsprocessen är en viktig del i mjukvaruutveckling då det gäller att få hela gruppen att förstå innebörden av metoden och vilket syfte som ett projekt använder sig av BDD. Projektet startade 2011 och det var en deltagare i Projektet som yrkade på att använda sig av metoden då han ansåg sig se en nytta med BDD i Projektet.

*“Det var egentligen en kille som hade varit iväg på en konferens där det hålls föreläsningar och de hade varit på en föreläsning om BDD och han tyckte att det verkade bra och ville testa det och så hade han nog läst på lite. Så han sålde in det till oss andra och frågade om vi inte skulle kunna använda BDD. Så vi läste väl på lite allmänt om det här klassiska, han Dan North som liksom har hittat på det hela, så vi läste liksom hans artikel där han sålde in det hela om skummade lite på nätet men artikeln var ju väldigt bra”*

Informant 1

Informant 2 till 4 var inte med vid uppstart av Projektet, i deras fall handlade det om att lära sig hur Projektet använder sig av BDD. För alla informanter var detta deras första projekt med BDD och ingen har tidigare använt metoden ordentligt. Två av informanterna hade hört talas om BDD innan de började i Projektet, men inte använt det fullt ut eller i samma skala som Projektet använder metoden.

*“Jag visste inte ens var det var”*

Informant 3

*“Jaa, men inte jobbat på djupet med det som vi försöker göra här”*

Informant 2

För att lära sig mer om metoden och utbilda nya medlemmar i projektet finns det inga specifika riktlinjer. När utvecklarna hittat ett arbetssätt som fungerade valde de att hålla fast vid sitt sätt att arbeta och fann ingen nytta med att söka mer teori eller att utveckla användandet av metoden. Detta i sin tur har medfört att de nya deltagarna som kommer in i Projektet behöver lära sig av andra utvecklare genom praktisk användning och learn-by-doing.

*“Jag kan väl säga att jag lärt mig genom andra utvecklare. Bara praktisk användning med bra stöd från de gamla utvecklarna som var väl pålästa”*

Informant 3

När Projektet väl startat var deltagarna inte säkra på hur metoden skulle svara upp till deras förväntningar och mål. Det är viktigt att känna sig trygg den valda metoden när det uppstår problem. Informant 1, som var med vid uppstarten av projektet, kände sig otrygg i början av användandet av BDD. De deltagare som kommit in i Projektet i efterhand upplever att de har tillräckligt med kunskap och trygghet för att använda metoden.

*“Jaa det gjorde jag, rätt enkel metod spontant, sen trodde jag det skulle vara enklare”*

Informant 2

*“Jo, jag tycker att jag har tillräckligt med kunskap för att arbeta med BDD men jag hade gärna testat på att arbeta med det i ett annat projekt också”*

Informant 3

Vår informant från Referensprojektet, som är ett nystartat projekt, har en annorlunda syn på upplärningen och menade att den är mycket viktig för att få projektet att flyta på samt att ha en tydlig struktur så att inte kunskapen lämnar projektet med tiden. I detta projekt spenderade deltagarna ordentligt med tid innan uppstart att fundera på hur de skulle använda metoden och vilka verktyg de skulle behöva för att få ut nyttan av BDD. I början lades mycket tid och pengar på att få ihop gruppen och skapa ett arbetssätt som skulle vara tydligt att följa. Genom en ordentlig struktur i upplärningsprocessen kan vem som helst i gruppen hantera BDD:erna och metoden kommer till större nytta.

*“Började med att skriva BDD:er tillsammans så att alla skriver på samma sätt. Det fick kosta lite, går 6 gånger så många timmar än att en person hade gjort det. Det är en puckel man får ta [...] Det var mycket gemensamt arbete i början, tar vi in nya resurser så går de inte själva utan får gå parallellt en stund. Vi ser det som extra viktigt för att vi vill ha med BDD-tänket när man utvecklar.”*

Informant 5

## 4.2 Utvecklingsprocessen

I detta avsnitt presenteras hur Projektet arbetar med BDD. Projektet arbetar agilt genom att använda sig av planeringsmetoden Scrum och sen använder de BDD som hjälp vid mjukvaruutvecklingen. De använder Scrum genom att Projektet arbetar i sprintar om tre veckor. Innan sprinten börjar planerar de upp vilka funktioner som ska utvecklas under följande tre veckor. BDD:erna kommer in som en återkommande standarduppgift under utvecklingen och är en stor del av arbetssättet.

*“I projektet kommer BDD:erna in som en standard task och blir en gul lapp i en sprint. En blir att skriva BDD:er, en blir att implementera dom och en blir att stämna av dom. Vi har 4-5 standard uppgifter som är om BDD:er [...] Det genomsyrar egentligen hela vårt arbete. Jag tycker att det är vårt viktigaste arbetssätt för att nå våra mål, jag kan inte säga att det är det jag jobbar med men det är en viktig del av det”*

Informant 3

Projektgruppen består av en kravanalytiker som diskuterar med beställare och en referensgrupp om vilka krav och uppgifter systemet måste ha. Efter överlämningen till utvecklarna börjar arbetet med att skriva BDD:er och är något oklart pratar utvecklarna direkt med referensgruppen som består av verksamhetspersonal med specialkunskap om vissa delar i Bolaget. Under dessa möten är det vissa som förstår BDD:er men till största delen är det utvecklarna som äger BDD:erna.

*“När vi utvecklare fått paketet med krav försöker vi samlas och bena ut vad vi ska göra och då kommer vi in i vår del av det agila arbetet där vi sätter upp en eller flera user-stories”*

Informant 2

Informanterna menar att det bara är utvecklarna som skriver BDD-testerna. De skriver direkt scenarion som översätts till körbara tester och börjar inte med stories redan vid kravanalysen.

*“Teoretiskt ska det väl börja med att vi ska börja skissa på BDD:er eller att man analyserar kraven och skriver ner det någonstans. Så att de hänger med hela vägen till mål. Men i vårt projekt lämnas kraven över och sedan gör vi BDD:er av det. Kravarna är inte bekväma med BDD så det är vi utvecklare som gör det så vi kanske skapar BDD:erna lite för sent”*

Informant 4

Utvecklarna gör alltså om kraven till fungerade BDD:er som implementeras i koden och testar specifika beteenden i systemet. Flera menar på att det gärna sett att BDD:erna började skrivas tidigare och att hela gruppen skulle vinna på att fler var involverade i metoden. Utvecklarna menar att de använder BDD mer som att bara testa kod och som deras omskrivning av krav på funktionalitet. Tankarna kring hur tekniska BDD:erna är går isär något.

*“Det är att BDD:erna har blivit lite för tekniska än vad de borde vara. Vi har ett tänk att BDD:erna ska bli test. Vi bygger det mycket nu för att det ska vara kvalitetssäkring”*

Informant 4

*“Inte alls tekniska men där emot är dom väldigt specifika”*

Informant 2

Utvecklingsprocessen i Referensprojektet ser lite annorlunda ut än i Projektet. Det som är likt är att BDD används i samma låga utsträckning med verksamhetspersonalen i deras referensgrupp.”

*“De stöttar oss med kraven och det är väl meningen att de ska signa dom så vi kan gå vidare men de får inte se BDD:erna utan de får testademiljön i tre veckor och är dom nöjda med det så blir det som att dom signar att BDD:erna stämmer för denna version. “*

Informant 5

Dock har de ett annat användande av BDD i Referensprojektet. Där använder sig fler projektdeltagare av metoden vilket visar sig i frågan om vilka som skriver BDD i projektet.

*“Det är i princip vem som helst. [...] Vi har en utvecklingsaktivitet för en viss feature som kan vara utveckla BDD för kanske då 'Skapa anteckning'. Då tar man den lappen och är den inte skriven så skriver man den, är den redan skriven så behöver man bara utveckla den.”*

Informant 5

### **4.3 Gemensamt språk**

Gemensamt språk är en viktig del i arbetet med BDD. Samtliga utvecklare i Projektet ansåg att den interna kommunikationen mellan utvecklare fungerade bra gällande BDD:er.

*“Men jag tycker vi ändå pratar ett bra språk med varandra och vi förstår varandra”*

Informant 4

*“Projektet var faktiskt ett av de projekt där kommunikationen funkade som bäst”*

Informant 1

Dock rådde det delade meningar gällande vilken nytta BDD:er har för det gemensamma språket mellan olika roller samt mellan Projektet och sina intressenter.

*“Det som vi skiljer oss från teori är att vi utvecklare äger BDD:erna i för stor utsträckning snarare än att det är samtal för alla”*

Informant 3

*“Om vi tänker på våra referensgrupper som varit med länge har de ju fått en viss förståelse för BDD:er men en helt ny som får se en BDD fattar inget, vad ska du ha det till.”*

Informant 4

Samtidigt menar utvecklarna att BDD:erna med sin struktur kan skapa en ökad förståelse hos intressenter. Även fast ett bestämt gemensamt språk inte existerar bidrar BDD:er till att ett växer fram.

*“Det var det som var grejen att de blev ganska svåra när vi utvecklare hade skrivit dom men när vi förklarade för honom så förstod han”*

Informant 1

Alla informanter menar dock att det finns flera anledningar till att ett gemensamt språk inte existerar. Detta p.g.a. att enbart utvecklarna använder sig av att skriva BDD:er och det sker då missförstånd när de pratar om mjukvaran som ska utvecklas.

*“Mycket missförstånd mellan krav, utveckling och projektledning”*

Informant 2

*“Då ska vi liksom utifrån en text förstå kravaren, på något magiskt sätt kunna förstå vad hen har diskuterat fram”*

Informant 4

Inom Projektet finns det dock en svag tro på BDD som lösningen till ett gemensamt språk mellan intressenter och Projektet.

*“Men om man frågar ekonomer så undrar ju dom varför vi använder BDD:er, det är ju ingen sagostund liksom, som BDD kan uppfattas som ibland. “*

Informant 4

Informanten från Referensprojektet menar på att de har en mycket bra kommunikation i hela projektgruppen och även med utomstående intressenter. Dock har de beslutat att inte diskutera krav med hjälp av BDD:er med verksamhetspersonal då de anser att det är irrelevanta och tidskrävande. De menar att det är deras metod och att det är deras ansvar att BDD:erna blir rätt skriva utifrån vad verksamhetspersonalen säger.

*“Det är en mycket bra kommunikation, Vi har bytt sakkunniga för att få det så bra som möjligt. De måste kunna uttrycka saker i modeller o förstå processer men även kunna verksamheten. Där har vi rullat lite tills vi hitta rätt personer.[...] Vi känner att vi inte behöver det, blir lite tidskrävande och att det inte riktigt är relevant, vi pratar om det mer på en konceptnivå. Här ska vi kunna söka och förklara det enkelt för dom”*

Informant 5

Även i Referensprojektet finns en svag tro på att BDD ska vara lösningen till att skapa ett gemensamt språk med strukturerade mallar för att skriva krav.



*“Man får acceptans för sin BDD via prototyp [...] Kunden ska inte behöva slösa sin tid på att skriva något på en viss form. Det är lite vårt ansvar fånga det och säkra upp att vi har fångat det.”*

Informant 5

Informant 5 från Referensprojektet menar också på att de har haft andra förutsättningar för att lyckas än Projektet. De har lärt sig av Projektet och har kunnat vara mer noggranna i uppstarten samt att de har lagt mycket vikt på att få en gemensam bild av vad BDD ska hjälpa dem med.

*“Vi har haft andra förutsättningar, de var ju först. Vi har haft en helt annan möjligt och kunnat ta ett helhetsbegrepp. De har haft andra krav på sig för att göra på ett annat sätt, medans vi har googlat och tagit redan på vilka förutsättningar vi behöver. Vi visste alla riskerna med det innan vi drog igång.[...] Har man en gemensam bild så funkar vilken metod som helst egentligen ”*

Informant 5

#### **4.5 Automatiserade tester**

I Projektet skrivs BDD:erna och implementeras i ett verktyg och med hjälp av det kan automatiska tester av beteendet på systemet köras. Utvecklarna i projektet menar att de automatiska testerna hjälper till att skapa en trygghet för projektdeltagarna vilket gör att de vågar testa nya lösningar och ändra på befintlig kod. Flera informanter menar att med automatiserade tester i form av BDD:er upptäcks ett eventuellt fel omgående och det skapar trygghet.

*“Det är en himla trygghet som utvecklare att veta att man har tester och då vågar man gå in och ändra på kod annars kan man ju vara rädd att förstöra något”*

Informant 1

Deltagarna i Projektet menar dock att det verktyg de använder sig av till de automatiserade testerna är opålitligt. Det tar lång tid att testa och ibland visar körda test felaktigt resultat. Ett test kan visas vara felaktigt när det i själva verket är korrekt vilket skapar irritation. Problemen har ökat med att testmassan har blivit större ju längre projektet pågått. Det skapar problem att hålla testerna rullande då det blir en tidskrävande process.

*“Jag trodde vi hade mer problem med att testmassan växte ju längre projektet gick. Desto längre projektet gick desto mer problem fick vi med att hålla testerna rullande och flytande.”*

Informant 1

Även i Referensprojektet finner dem att de automatiserade testerna är av nytta och bidrar med en trygghet till utvecklarna. De menar att skriva testerna kan ta tid men att det är värt det vid själva testningen.

*“BDD i sig är inte tidskrävande. Jag tycker BDD är väldigt bra att beskriva krav sen om du ska använda BDD för att testa då är det tidskrävande men det behöver du inte göra. Du kan*

*använda det bara till kravfångst också, men fördelarna kommer ju med det när du testar. Om du ändrar något så vet du var det inte fungerar längre.”*

Informant 5

*“Man har en leveransklar produkt hela tiden, så vidare du inte tar sönder något men då visar det sig. Vi siktar på att ha 70-80% krav automatiserade, då vet man att 80% funkar bra i systemet. Man behöver hålla upp den automationsnivån annars blir det en kravmassa som inte underhållits. Den byggs in i utvecklarnas vardag”*

Informant 5

Referensprojektet menar att stor anledning till vinsten av de automatiserade testerna hänger på rätt verktyg.

*“Det hade inte funkat om vi inte hade haft rätt verktyg.”*

Informant 5

#### **4.6 Dokumentation**

Dokumentationen av krav utförs i ett dokument som sedan bryts ner till flera BDD:er av utvecklarna i Projektet. Då BDD:erna sparas och är kravspecifikationen av systemet kan den användas som dokumentation av systemet.

*“Vi utvecklare menar att BDD:erna är facit och det dokumenterar ju det implementerade beteendet”*

Informant 2

*“Istället för att ha massvis med dokumentation med krav så kan man ha BDD istället”*

Informant 4

Dock finner deltagarna i Projektet att eftersom kravframställan inte är enligt BDD-modell skapas en del dubbeldokumentation.

*“De kanske hade varit enklare som alla varit på banan och känt ägarskap över BDD:erna. Det hjälper ju till att dokumentera kraven och om alla inte är med på det så blir det dubbeldokumentation.”*

Informant 3

Förutom BDD:erna och kravdokumentet finns egentligen ingen mer dokumentation i Projektet mer än några standarddokument som måste finnas i Bolaget. BDD:erna beskriver vilka beteenden systemet ska ha och ska vara lättförståeligt.

En stor skillnad mellan Projektet och Referensprojektet är att ägandeskapet av BDD:er finns bland alla deltagare. Därmed är det endast BDD som ligger till grund för deras dokumentation av systemet och att det då inte ska ske någon dubbeldokumentation. Informanten vill också påpeka att verktygen de använder skapar bra förutsättningar för att dokumentation blir mindre.

*“BDD som går igenom är ju både krav och testfall för oss. Det blir inget onödigt arbete och med rätt verktyg slipper vi skapa testrapporter [...] Man kan inte ta sig an detta halvhjärtat då blir det en risk. Då har du krav i pappersform, BDD:er, implementation och tester. [...] Vi vill inte ha någon dubbeldokumentation [...] Vi har bättre verktygsstöd än Projektet. Det behövs för att få struktur och inte få dubbelarbete av krav.“*

Informant 5

## 5. Diskussion

Uppsatsen ämnade besvara vilka faktorer som behöver beaktas vid användandet av beteendedriven utveckling. Det har gjorts genom att studera publicerad litteratur om BDD och hur metoden ska användas samt vilken affärsnytta metoden ska bidra med. För att komplettera teorin genomfördes även en fallstudie i två projekt på ett företag som använder sig av mjukvaruutvecklingsmetoden BDD. I detta kapitel diskuteras vårt resultat från fallstudien mot bakgrund av vår insamlade teori. Kapitlet kommer vara strukturerat på samma sätt som vår resultatdel med följande underrubriker: utbildning/upplärning, utvecklingsprocessen, gemensamt språk, automatiserade tester och dokumentation.

### Utbildning/upplärning

Utbildningen är viktigt oavsett vilken metod som ska implementeras och senare användas. Genom utbildning av ett specifikt arbetssätt, i detta fall en mjukvaruutvecklingsmetod, skapas en förståelse hos verksamheten om varför ett visst arbetssätt används. Att få samtliga inblandade parter att förstå affärsnyttan med den mjukvaruutvecklingsmetod som ska implementeras kan visa sig vara avgörande för huruvida en implementation kommer lyckas eller misslyckas. I teorin ska BDD som mjukvaruutvecklingsmetod bidra till att koppla samman projektdeltagare oavsett roll eller teknisk bakgrund genom att använda sig av en gemensam struktur med ett gemensamt språk att jobba efter (Solís & Wang 2011; Soeken, Wille & Dreschler 2012). Om alla projektdeltagare inte har fått den utbildning eller upplärning som behövs kan det vara svårt att förstå syftet med metoden och då kan den förväntade affärsnyttan påverkas negativt. Det gäller att som organisation kunna anpassa tiden som läggs vid utbildning vid en implementation av en utvecklingsmetod efter vilken kunskap som finns i gruppen och stötta dem som behöver det. Det kan tyckas vara tidskrävande i början men det kan ge positiva effekter i form av ett ökat affärsvärde senare i projektet.

Enligt en av våra informanter som var med vid implementationen av metoden fann vi att viljan att byta arbetssätt saknades från vissa deltagare i projektgruppen. Det kan ha berott på att samtliga deltagare inte förstod affärsnyttan med metoden och således skapades en anpassning av metoden utifrån deltagarna i Projektet. Då BDD syftar till att samtliga projektdeltagare ska använda sig av metoden kan anpassning ha bidragit till att metoden inte kunde leva ut sin fulla potential. Det skulle kunna ha hjälpt att utbilda alla personer i Projektet gällande metoden för att skapa en förståelse för metoden och vilken affärsnytta den skulle kunna ha. I Referensprojektet finns en skillnad i synsättet och användandet av BDD. Vi tror att det beror på att de hade möjlighet att studera Projektets implementation och därigenom förstå nyttan med att spendera mer tid för att diskutera syftet med BDD och hur maximal affärsnytta uppnås med metoden.

När en organisation väljer att använda sig av BDD behöver den förstå riskerna med metoden innan den implementeras. En stor risk är att hela gruppen inte förstår affärsnyttan med metoden men det skulle kunna förhindras med en bättre utbildning av samtliga deltagare. En annan risk kan vara att den verksamhet som redan implementerat metoden fortsätter i ett invariant arbetssätt och lär upp ny personal att använda metoden som de alltid har gjort. I båda fall riskerar metoden att tappa sitt förväntade affärsvärde och eftersom BDD bygger på att samtliga gruppmedlemmar

känner sig delaktiga är detta enligt oss en viktig del att ta hänsyn till i implementationsfasen av metoden. Det kan givetvis vara problematiskt att utbilda personal i en metod där det finns relativt lite publicerad litteratur. Tillgången till relevant litteratur och användningsfall var troligtvis mindre för Projektet när de startade år 2011. Referensprojektet hade både mer litteratur och ett användningsfall inom samma verksamhet att tillgå. Vi anser dock att utbildning kan vara relevant för en metod som BDD som ämnar aktivera samtliga projektmedlemmar från flera olika roller som normalt kanske inte jobbar på samma sätt.

### **Utvecklingsprocessen**

North (2006) menar att han vid framtagandet av metoden fann att metodens struktur med bestämda mallar och med bestämda termer i olika delar av utvecklingsprocessen bidrog till att förståelsen för systemet ökade hos samtliga deltagare. Genom att använda sig av stories med bestämd BDD-mall för att beskriva vilka krav som finns för systemet tvingar det en kravanalytiker att fundera vilken affärsnytta en viss funktion ska ha för en viss roll. Eftersom storyn ska skrivas med ett gemensamt språk blir det enklare för en projektgrupp att diskutera kring en viss funktionalitet oavsett vilken roll personen har. Dessa stories skall senare ligga till grund för framtagandet av scenario-specifikationer som skall medverka till att konkretisera olika beteenden för en viss funktionalitet som är beskriven i en given story. Scenario-specifikationer har även de en given mall för hur de ska skrivas. Med denna struktur skall utvecklingsprocessen hjälpa till med att fler roller inom projektet skall känna sig delaktiga och en gemensam förståelse för systemet skapas (Lazar, Motogna & Pârv 2010).

I denna del fann vi att det fanns en del skillnader i hur Projektet använde sig av BDD och hur teorin beskriver processen. De får in krav från sin kravanalytiker och skriver sen user stories som ska beskriva vilken funktionalitet som den del av mjukvaran ska ha som de utvecklar. De använder inte den BDD mall som finns för att skriva stories utan menar att det kommer in krav på systemet och sen utvecklas user stories utifrån kraven. Enligt informanterna utvecklas BDD:erna först vid scenario-specifikationen. Där bidrar de till att vara en form av facit för kravspecifikation över vilka beteenden som systemet ska kunna ha. Några av informanterna bekräftade att de är medvetna om att det i teorin egentligen ska utvecklas BDD:er tidigare än de själva gör. I Projektet menar några informanter att samtliga deltagare inte är bekväma med att använda BDD som metod och använder det inte fullt ut. Det hjälper till att det bara är en viss grupp inom Projektet som använder sig av metoden, i detta fall bara utvecklarna, och därmed har en projektspecifik anpassning av BDD skapats. I Referensprojektet existerar en annan mentalitet vilket är ett möjligt utfall av att de har haft möjlighet att lära av Projektets implementation. I deras projektgrupp är det samtligas uppgift att utforma BDD:er och detta skapar en gemensam förståelse av hur de skriver BDD:er samt över systemet och dess funktionalitet. Vi finner i Referensprojektet att gruppen har en klar bild av vad de vill få ut av BDD och därför är det för dem viktigt att samtliga använder sig av metoden. Likheten mellan Projektet och Referensprojektet var att de inte kopplade in sin kund i utvecklingen av BDD:er utan istället bara använde BDD intern i sin projektgrupp.

Solís och Wang (2011) påpekar att nyttan med att koppla in stories med BDD-struktur redan vid kravanalysen då det blir tydligare för hela gruppen vad som ska utvecklas och vilken

affärsnytta funktionaliteten ska bidra med. Om missförstånd i vad som ska utvecklas uppstår skall det även bidra till att underlätta kommunikationen eftersom det i stornyn existerar en tänkt användare som går att prata med om funktionaliteten. I Projektets fall kan det som nämnts ovan bero på att hela projektgruppen inte hade fått utbildning i vilken affärsnytta metoden skall ha och därmed existerar inte den förståelse som krävs för att själv ändra arbetssätt inom vissa roller. Vi förstår att det är en komplex situation då det inte passar alla verksamheter att koppla in sina kunder eller samtliga projektdeltagare i utvecklingsprojekt. Men då BDD bygger på att samtliga deltagare känner ägandeskap av BDD:erna och således ökar sin förståelse för systemet finns det risk att metoden tappar av sitt förväntade affärsvärde.

### **Gemensamt språk**

Det är den gemensamma strukturen i alla delar av BDD tillsammans med det gemensamma språket som ska bidra till att projektgruppen, oavsett roll eller teknisk erfarenhet, har enklare att förstå varandra (Lazar, Motogna & Pârv 2010). Enligt Solís och Wang (2011) ska inte det gemensamma språket som existerar i utvecklingsfasens mallar vara allt som används. I sin artikel menar de att det även ska finnas en form av gemensam ordlista som sätts redan i början av projektet. Detta för att alla projektdeltagare ska referera till systemet på samma sätt i all form av kommunikation, inte bara i den skriftliga. Det är det gemensamma språket som North (2006) lyfter fram som en av de viktigaste aspekterna med BDD. Genom att använda sig av det gemensamma språket i både utvecklingsprocessen och i den kommunikation som råder mellan projektdeltagare ska en ökad förståelse för systemet och dess funktionalitet att skapas.

I Projektet fann vi att, trots att de inte använder sig av utvecklingsprocessens alla delar, att det ändå har skapats goda förutsättningar för att kommunicera genom BDD. De menar att de kunder de har haft kontakt med länge ändå har kunnat bilda sig en bättre uppfattning om systemets funktionalitet och det är till viss del på grund av BDD:s struktur. Dock menar informanterna att det samtidigt ibland sker missförstånd inom olika roller i projektgruppen på grund av en brist i ett gemensamt sätt att kommunicera. Då varken alla projektdeltagare eller BDD strukturen används fullt ut är det svårt att hitta att de använder sig av ett gemensamt språk i form av det är tänkt med metoden. I Referensprojektet är det gemensamma språket för utvecklandet av BDD bra och det gäller även mellan gruppens olika roller. Anledningen kan vara att de har tagit ett gemensamt ägandeskap för utvecklandet av BDD:er och genom en god implementationsfas har de skapat sig förutsättningar för att använda sig av ett gemensamt språk. Dock använder sig båda grupperna inte av BDD:er i kommunikationen med sin kund. Ingen av dem använder sig heller av BDD struktur i både sina stories och sina scenario-specifikationer vilket är en viktig del i användandet av BDD. Båda projektgrupper anser sig ha liten förhoppning att BDD skulle bidra till stor nytta när det kommer till kommunikationen mellan kund och projektgrupp.

Att ändra rutiner när det kommer till exempelvis framtagandet och formulerandet av krav förstår vi är en komplex del för både organisationen som helhet och för personerna som verkar i dem. Det kan finnas flera anledningar till att en organisation väljer bort en del i en utvecklingsmetod speciellt när det kommer till en del som primärt kan kännas tidskrävande. Dock finner vi att det finns en vilja i Projektet att dela på ansvaret när det kommer till skrivandet av BDD:er men att det ändå inte används av alla projektdeltagare. Genom att dela på ett sådant

ansvar hade de kunnat öka nyttan när det kommer till att utnyttja det gemensamma språket som ska finnas vid användandet av BDD. Gruppen hade eventuellt gynnats av att känna ett delat ägandeskap gällande utvecklandet av BDD:er och det hade i vissa fall eventuellt bidragit till mindre missförstånd. Huruvida en verksamhet ska koppla in även sin kund kan även det vara komplext. Det är en tidskrävande process att sätta in samtliga deltagare i ett projekt kring användandet av en viss struktur i en metod. Det faktum gör att vi förstår att vissa delar väljs bort i brist på övertygelse kring att öka affärsnyttan. I fallet med både Projektet och Referensprojektet kan vi förstå att kunden lämnas ute från användandet av BDD då det hade varit en tidskrävande process och eventuellt inte hade renderat i något affärsvärde. Samtidigt tror vi att det är viktigt att förstå nyttan kring att ha ett gemensamt språk att referera till i ett system. Genom ett gemensamt språk finns det vinster i att fler i projektet känner sig inkluderade i utvecklingsprocessen och genom att förstå varandra bildas det mindre utrymme för tidskrävande missförstånd. Att anpassa metoden efter sina förutsättningar är inte fel men inom den närmaste gruppen borde en samsyn existera.

### **Automatiserade tester**

Att använda sig av automatiserade tester ska enligt Janzen och Saiedian (2005) hjälper till att skapa en trygghet hos mjukvaruutvecklaren att ändra i koden utan att behöva vara orolig för att något blir allvarligt fel då testerna bidrar med snabb feedback av systemets tillstånd. Solís och Wang (2011) menar att BDD har ärvt tidigare testfokuserade metoders vinst genom att kunna överföra scenarion till körbara test.

Projektgruppens informanter bekräftar den teori som är behandlad kring automatiserade tester. De finner en stor trygghet i att koden testas kontinuerligt och att det påvisas relativt snabbt om något har gått fel. Den tryggheten visade sig vara stor hos informanterna när de var nya inom projektet för att de då vågade ändra mer i källkoden. Med automatiserade tester som hjälp hämmades oron över att göra fel i koden. Dock menar samtliga informanter i Projektet att brister i verktyget de använde för att köra testerna kan ligga till grund för irritation och tidsspill. Tester körs och fel visas trots att testerna borde ha godkänts. Vid omkörning visar det sig trots tidigare fel att testerna borde blivit godkända. Då detta var något som skedde relativt ofta hade en misstro till verktyget vuxit fram. I Referensprojektet bekräftas idén om att en stor fördel med metoden är att kunna köra automatiserade tester. Informanten bekräftar även att rätt verktyg är väsentligt för att kunna vinna fördelarna med automatiserade tester.

Både teori och vår fallstudie pekar på att det finns en stor vinst med att använda sig av BDD:s automatiserade tester. Att då kunna nyttja BDD:s struktur med scenarion och göra dem till körbara tester är väldigt bra. Dock påvisar vår fallstudie att mycket bygger på att använda sig av rätt verktyg för att köra testerna. Om en misstro växer fram för verktyget kan viktig affärsnytta i att kunna lita på sina tester gå förlorad.

### **Dokumentation**

Genom att använda sig av BDD:s olika delar kan det bidra till att all nödvändig dokumentation av ett system och dess funktionalitet redan existerar (North 2006). Minskat

dokumentationsarbete ökar tiden till att faktiskt utveckla och leda utvecklingen av systemet framåt (Solis & Wang 2011). Att lägga mindre tid på att dokumentera och mer på att faktisk utveckla ligger även i linje med de riktlinjer som finns för agila utvecklingsmetoder vilket BDD tillhör (Fowler & Highsmith 2001).

I Projektet menar informanterna att BDD hjälper till med dokumentation av systemet då BDD:erna anses vara facit av systemets funktionalitet. De menar att med hjälp av BDD går det att spara tid som senare annars skulle läggas på dokumentation. Dock menar de att en ökad känsla av ägandeskap för utvecklandet av BDD:er inom alla roller i projektet skulle kunna innebära att dubbeldokumentation undviks. Inom Referensprojektet har en stor känsla för ägandeskap för skrivandet av BDD och en gemensam överenskommelse för att det endast ska vara BDD som ligger till grund för dokumentationen bidragit till att ingen dubbeldokumentation sker. De anser att det är en av de stora vinsterna med att använda sig av metoden.

I olika verksamheter ser kravet på dokumentation olika ut. Dock är dokumentation av ett system alltid ett krav. Om BDD används på det sätt som är beskrivet i teorin ska dubbeldokumentation inte vara nödvändig utan det ska bidra till att utvecklingsprocessens olika delar ligger till grund för den dokumentation som krävs. I koden implementeras de körbara testerna vilket med sina tydliga strukturer och tydliga beteendeförklaringar skapar en dokumentation för källkoden och vilken funktionalitet systemet har. Vår fallstudie påvisar att om en verksamhet inte använder sig av BDD fullt ut finns det risk för dubbeldokumentation och att det då spills tid som annars kunde nyttjas åt annat. I Referensprojektet påvisas dock den vinst som användandet av BDD ska hjälpa till med när det kommer till dokumentation. Vid korrekt användande ska det i stort sett räcka med den dokumentation som BDD bidrar med.



## 6. Slutsats

Det huvudsakliga syftet med vår studie var att besvara frågan: *Vilka faktorer behöver beaktas vid användandet av beteendedriven utveckling?* Nedan följer vår slutsats:

Vår studie har visat att implementation och användandet av BDD kan vara problematisk. Vårt resultat visar att BDD, av olika skäl, anpassas för att fungera i den miljö den ska verka i p.g.a. att olika projekt har olika förutsättningar både i form av individer och i form av tillgång till rätt verktyg. Verksamhetsspecifika anpassningar i sig behöver inte vara fel men risken är att affärsnytta kan gå förlorad. I vår studie framgår att följande faktorer behöver beaktas vid användandet av beteendedriven utveckling:

- Skapa en förståelse för **affärsnyttan** med BDD i hela projektgruppen i implementationsfasen
- Få hela projektgruppen att känna **ägaransvar** över BDD:erna
- Bestäm ett **gemensamt språk** och följ BDD strukturen
- Förse sig med **rätt verktyg** för att nyttja fördelarna av automatiserade tester
- **Tydliga BDD:er** ska resultera i systemdokumentation

### 6.1 Studiens relevans

Vårt fokus har varit att studera vilka faktorer som behöver beaktas vid användningen av beteendedriven utveckling. Beteende driven utvecklingen är en relativ ny metod och som vi tidigare skrivit uppkom 2006 och därför finns det inte mycket forskning om detta. Den forskningen som finns om BDD är framtagen för att belysa positiva aspekter med metoden samt att påvisa hur BDD bör användas. Därför anser vi att vår studie kommer bidra med ytterligare ett perspektiv genom att påvisa hur det faktiskt används i praktiken. Genom att ha med båda perspektiven i vår studie, hur det bör användas och hur det faktiskt används hoppas vi att flera kommer kunna ha nytta av vår studie i antingen förbättringssyfte av metoden eller vid nya implementationer av BDD. Studien är skriven utifrån ett projekt med ytterligare ett projekt som referens men vi anser dock att resultatet och slutsatsen är generell och kan hjälpa flera projekt som använder sig av BDD eller som har som plan att börja använda BDD i framtiden. Vidare kan det diskuteras i att fler behöver forska i hur metoder och BDD används i praktiken. Detta är något vi anser vara viktigt för att kunna lyckas med att välja rätt metod för sitt projekt.

### 6.2 Vidare forskning

Med denna studie har vi undersökt hur beteende driven utveckling används i praktiken. Vårt fallstudieobjekt har varit ett IT-projekt som pågått en längre tid med många som slutat och börjat. Vid vidare forskning skulle det varit intressant att studera flera olika projekt i olika organisationer och undersöka vilka orsaker och aspekter som är viktiga vid användandet av BDD. Det skulle också varit intressant att studera en pågående implementation av BDD om det är möjligt. Till sist vore det intressant att undersöka varför en viss metod väljs och hur stor betydelse utbildningsprocessen har på metodens utfall.

## Referenser

- Atem de Carvalho, R. Luiz de Carvalho e Silva, F. Manhaes, R. (2010) *Mapping business process modeling constructs to Behavior Driven Development Ubiquitous Language*. Brazilian Public Software Portal 2010
- Cao, L. Balasubramanian R. (2007) *Agile Software Development: Ad Hoc Practices or Sound Principles?* IEEE Computer Society, March – April 2007
- Diepenbeck, M. Soeken, M. Große, D. Dreschler, R. (2012) *Behavior Driven Development for Circuit Design and Verification*. IEEE Computer Society 2012.
- Dingsøyr, T. Nerur, S. Balijepally, V. Moe, N.B. (2012) *A decade of agile methodologies: Towards explaining agile software development*. Journal of Systems and Software, Vol 85 June 2012, pp 1231-1221
- Fowler, M. Highsmith, J. (2001) *The Agile Manifesto*. Software Development, August 2001.
- Greer, D. Hamon, Y. (2011) *Agile Software Development*. Software: Practice and Experience, Vol 41 August 2011, 943-944.
- Janzen, D. Saiedian, H. (2005) *Test-Driven Development: Concepts, Taxonomy and Future Direction*. IEEE Computer Society, September 2008
- Klein, K., & Myers, D. (1999). *A set of principles for conducting and evaluating interpretive field studies in information systems*. MIS quarterly, 67-93.
- Larman, C. Basili, V. (2003) *Iterative and Incremental Development: A brief history*. IEEE Computer society, June 2003.
- Larsson, P. (2008) *Nu ska det gå att förstå program*. <http://computersweden.idg.se/2.2683/1.201407/nu-ska-det-ga-att-forsta-program/> [2016-05-12]
- Lazar, I. Motogna, S. Pârv, B. (2010) *Behavior Driven Development*. Electronic Notes in Theoretical Computer science 264 (2010) 91-105. ScienceDirect.
- Mens, T. Demeyer, S. (2008) *Software Evolution*. Berlin: Springer Publishing Company, 1-3.
- Nerur, S. Balijepally, V. (2007) *Theoretical Reflections on Agile Development Methodologies*. Communications of the ACM, March 2007/ Vol. 50 No. 3
- North, D. (2006). *Introducing BDD*. <http://dannorth.net/introducing-bdd/> [2016-05-12]

Patel, R. Davidson, B. (2011) "*Forskningsmetodikens grunder.*" Fjärde Upplagan. Lund: Studentlitteratur

Ranerup, A., jan 2016. Metod 4: Teorier, intervjuer och analys. Föreläsning på kursen TIG165 vid Institutionen för Tillämpad IT på Göteborgs Universitet, Göteborg.

Shore, J. Warden, S. (2007) *The Art of Agile Development*. Beijing: O'Reilly Media, 125.

Soeken, M. Wille, R. Dreschler, R. (2012) *Assisted Behavior Driven Development Using Natural Language Processing*. C.A. Furia and S. Nanz: TOOLS Europe 2012, LNCS 7304, pp 269-287, 2012. Springer-Verlag Berlin Heidelberg 2012.

Solís, C. Wang, X. (2011) *A Study of the Characteristics of Behavior Driven Development*. IEEE Computer society 2011

Stober, T. Hansmann, U. (2010) *Agile Software Development. Best Practices for Large Software Development Projects*. Berlin, Heidelberg: Springer, 16-21.

Stoica, M. Mircae, M. Ghilic-Micu, B. (2013) *Software Development: Agile vs. Traditional*. Informatica Economica vol. 17 no. 4/2013

Tavares, H. Rezende, G. Mota dos Santos, V. Manhaes, S. Atem de Carvalho, R. (2010) *A tool stack for implementing Behaviour-Driven Development in Python Language*. Brazilian Public Software Portal 2010.

Walsham, G. (1995). *Interpretive case studies in IS research: nature and method*. European Journal of information systems, 4(2), 74-81.

Walsham, G. (2006) *Doing interpretive research*. European Journal of Information Systems, 15(3): 320-330

# Bilagor

## Bilaga 1 Kvalitativ intervjumall

### Introduktion

- Förklaring av vilka vi är
- Vad vår uppgift är
- Varför vi intervjuar
- Förklara roller. En intervjuare och en som antecknar och eventuellt ställer fördjupande frågor.
- Förklara att vi spelar in
- Förklara att deras namn kommer styrkas i arbetet
- Förklara att deras roll och erfarenhet kommer vara med

### Presentation

1. Har du tidigare erfarenheter av projektarbete inom IT-branschen?

Vilka?

Metoder?

Jobbat med BDD innan?

2. Vad har du för koppling till projektet?

**Fokuspunkter:**

Vilken är din roll?

Hur länge har du jobbat med BDD?

Om ej med längre, varför är du inte med längre?

### Användning av BDD:

(Fokus ligger på att få en förståelse på hur de använder sig av BDD i de olika rollerna de har. Om de har varit med vid implementation vill vi veta mer om det)

**Till person som använder BDD idag:**

3. Vilka roller finns det i ert projekt och kan du beskriva dem?

**Fokuspunkter**

Vem gör vad?

Beslut?

4. Vid inkommande av något som ska tas fram. Kan du beskriva den processen från start till mål?

**Fokuspunkter**

Vem gör vad?

Kommunikation?

Använder alla BDD då?

## FRÅGA 5

5a. Hur fungerar processen när ni jobbar med BDD?

### **Fokuspunkter:**

Vilka är involverade i processen? Från start till mål?

Hur mycket är alla olika roller involverade?

Vem får du ny information ifrån?

Vem förmedlar du den informationen till?

Vem är kund?

Dokumentation av kod?

Förklara personens del i arbetsprocessen?

### **Till person som använt BDD tidigare:**

5b. Hur fungerade processen när ni jobbade med BDD?

### **Fokuspunkter:**

Vilka är involverade i processen? Från start till mål?

Hur mycket är alla olika roller involverade?

Vem är kund?

Dokumentation av kod?

Förklara personens del i arbetsprocessen?

Förklara personens del i implementationsdelen?

Varför BDD?

6. Förklara hur du ser på BDD som metod?

### **Fokuspunkter:**

Deras svar kommer generera följdfrågor vid positiva/negativa svar. 3 punkter?

Positivt?

Negativt?

Jämför med andra metoder?

Anser du att BDD är en bra metod för det ni gör?

## **Utbildning om BDD:**

(Vi vill i denna del få reda på hur det ser ut när Bolaget utbildar personer som ska använda sig av BDD.)

7. Förklara hur din utbildning av BDD såg ut?

### **Fokuspunkter:**

Om de var med i början. Hur såg deras utbildning ut?

Teoretisk eller praktiskt utbildning?

Hur stor del teoretisk/praktiskt? (Uppskattning)

8. Förklara hur det ser ut vid utbildning av en ny person som kommer in?

**Fokuspunkter:**

Dokumentation av gammal kunskap?  
Kompetensöverföring från gammal kollega?  
Spelar personens roll någon roll vid utbildning?

9. Anser du dig ha tillräckligt med kunskaper för ditt arbete med BDD?

**Fokus:**

Har du någon gång inte förstått din uppgift?  
Förstår du syftet med din uppgift?

10. Enligt dig och den uppfattning du har, ser du någon skillnad på BDD i teorin och hur ni använder BDD?

**Fokuspunkter:**

Om inga avvikelser. Utveckla svaret. Varför är det så?  
Om avvikelser. Utveckla svaret. Varför är det så? (offentlig verksamhet? kompetens?)

## **Kommunikation i projektet**

(Vi vill undersöka kommunikationsflödet i projektet)

11. Anser du att det finns alla förutsättningar för att dokumentera i ert projekt?

**Fokuspunkter**

På vilka sätt dokumenteras det  
Hur kommuniceras det  
Använder ni er av ett gemensamt språk?  
Uppstår det missförstånd eller har ni lätt att förstå varandra i hela projektet?

12. Kommunikationen överlag i er projektgrupp, hur ser den ut gällande exempelvis nedanstående fokuspunkter?

**Fokuspunkter:**

Förändringar  
Deadlines  
Planering  
Mail  
Nära kontakt  
Vid felaktigheter, vem gör vad?

## **Avslutning**

13. Är det något ni anser att vi har missat som du anses borde beröras när det kommer till användandet av BDD?

14. Om du skulle få önska hur BDD 2.0 såg ut. Vad skulle det vara?

15. Avslutningsvis. Hade du valt BDD i ert projekt om du hade fått välja idag?

**Ja/Nej**

Ge oss tre anledningar till varför/varför inte.