



LUNDS UNIVERSITET

Ekonomihögskolan

Institutionen för informatik

Hur kvalitetssäkring påverkas av agila arbetsmetoder

Kandidatuppsats, 15 högskolepoäng, SYSK02 i Informationssystem
Framlagd juni 2015

Författare: Jonathan Barr
Axel Odell
Christopher Widell

Handledare: Bo Andersson

Examinatorer: Markus Lahtinen
Paul Pierce

Hur kvalitetssäkring påverkas av agila arbetsmetoder

Författare: Jonathan Barr, Axel Odell och Christopher Widell

Utgivare: Inst. för informatik, Ekonomihögskolan, Lunds universitet

Dokumenttyp: Kandidatuppsats

Antal sidor: 96

Nyckelord: kvalitetssäkring, agila utvecklingsprojekt, agila arbetsmetoder, scrum, extreme programming, quality assurance.

Sammanfattning:

Uppsatsen hade syftet att presentera kvalitetssäkringsbrister som eventuellt uppstod vid användandet av agila arbetsmetoder i utvecklingsprojekt. För att identifiera vad som ansågs vitalt för kvalitetssäkring inom agila utvecklingsprojekt utfördes en litteraturstudie som således ledde till framtagandet av en rad centrala beståndsdelar. Utifrån de identifierade centrala beståndsdelarna arbetade vi fram ett ramverk. Detta ramverk användes som grund för framtagandet av en modell som kunde appliceras på en empirisk studie. Den empiriska kvalitativa studien utgick ifrån två fall som arbetade utefter agila arbetsmetoder och vårt ramverk med tillhörande modell applicerades på studien. Resultatet visade att en rad beståndsdelar inte uppfylldes och därmed ledde till brister av kvalitetssäkring i den slutgiltiga produkten. Sambanden mellan resultat och litteratur diskuterades för att analysera de kvalitetssäkringsbrister som uppstod. Vidare forskning av liknande karaktär kan använda sig av det ramverk och den modell vår uppsats tog fram.

Innehåll

1	Introduktion.....	1
1.1	Bakgrund	1
1.2	Problemformulering.....	2
1.3	Syfte.....	2
1.4	Forskningsfråga	2
1.5	Avgränsningar	3
1.6	Intressenter.....	3
1.7	Centrala begrepp.....	3
2	Litteraturgenomgång	4
2.1	Vad är agil utveckling?.....	4
2.1.1	Extreme Programming	5
2.1.2	Scrum	7
2.1.3	Fördelar och nackdelar med agil utveckling	8
2.2	Kvalitetssäkring	9
2.3	Teoretiskt ramverk.....	12
2.3.1	Flygplansmodellen	15
3	Metod	17
3.1	Metodval.....	17
3.2	Urval	17
3.3	Genomförande av intervjuer	19
3.3.1	Motivering av intervjuguide.....	19
3.4	Undersökningskvalitet	21
3.4.1	Etik	22
3.4.2	Studiens validitet och reliabilitet.....	22
3.5	Datainsamling och analyser.....	23
3.6	Trovärdighet	23
4	Resultat av empirisk data	24
4.1	Företaget Avensia	24
4.2	Generell upplevelse och attityd	24
4.3	Centrala beståndsdelar	24
4.3.1	Arbetskvalitet	25
4.3.2	Kodkvalitet	27

4.3.3	Planeringskvalitet	28
4.3.4	Tillägg	32
4.4	Sammanfattning	33
4.4.1	Sammanfattande tabell	33
4.4.2	Sammanfattande diagram	34
5	Analys & Diskussion	37
5.1	Centrala beståndsdelar	37
5.1.1	Arbetskvalitet	37
5.1.2	Kodkvalitet	40
5.1.3	Planeringskvalitet	41
5.2	Urvalsproblematik	46
5.3	Sammanfattning av analys och diskussion	46
6	Slutsats	48
6.1	Hur påverkas kvalitetssäkring av agila arbetsmetoder?	48
6.2	Förslag till vidare forskning	48
7	Bilagor.....	49
7.1	Intervjuguide.....	49
7.1.1	Kort introduktion av författarna	49
7.1.2	Kort introduktion av informant	49
7.1.3	Intervjufrågor	49
7.2	Intervjuprotokoll.....	50
7.2.1	Intervjuprotokoll 1.....	50
7.2.2	Intervjuprotokoll 2.....	56
7.2.3	Intervjuprotokoll 3.....	61
7.2.4	Intervjuprotokoll 4.....	65
7.2.5	Intervjuprotokoll 5.....	78
8	Referenser	87

Figurer

Figur 1: Flygplansmodellen	16
Figur 2: Statistik över informanternas svarsfrekvens.....	35
Figur 3: Statistik över uppfyllandet av centrala beståndsdelar	36

Tabeller

Tabell 1: Agila principer (Beck et al., 2001).....	4
Tabell 2: Centrala beståndsdelar för kvalitetssäkring i <i>XP</i> (Cao et al., 2009; Iivari & Iivari, 2011).....	10
Tabell 3: Centrala beståndsdelar för kvalitetssäkring i <i>Scrum</i> (Cho et al., 2011; Dingsøy et al., 2012; Hoda et al., 2011; Iivari & Iivari, 2011).....	11
Tabell 4: Teoretiskt ramverk	12
Tabell 5: Informanttabell.....	18
Tabell 6: Projektöverblick	18
Tabell 7: Motivering av intervjuguide.....	19
Tabell 8: Sammanfattande tabell över intervjusvar.....	33
Tabell 9: Sammanfattande tabell över centrala beståndsdelssuppfyllnader	46

1 Introduktion

Följande kapitel inleder vi med en generell bakgrund till agil utveckling. Vi formulerar därefter problemområdet som existerar i det givna ämnet samt syftet med vår studie. Vi presenterar vår forskningsfråga med tillhörande avgränsningar, vilka intressenter som vår forskning riktar sig mot, och slutligen de centrala begrepp som används i vår uppsats.

1.1 Bakgrund

Agil utveckling blir vanligare då organisationer i större utsträckning strävar efter ett flexibelt och tidseffektivt arbetssätt (Holmström et al., 2006). Planbaserade mjukvaruutvecklingsmetoder, som exempelvis den inflytelserika vattenfallsmetoden (Royce, 1970), har sedan sitt ursprung haft problem med att hantera och kvalitetssäkra förändringar under utvecklingsperioden (Schümmer et al., 2006). Dessa svagheter, samt en önskan att kunna arbeta mer flexibelt, ledde till att det ansågs vara nödvändigt att ta fram en annorlunda och så kallat agilt arbetssätt (Beck et al., 2001). Det agila arbetssättet grundades 2001 efter att innovatörer inom ämnet samlades och skrev *The Agile Manifesto for Software Development* (Beck et al., 2001). Efter att grunden för agil mjukvaruutveckling cementerats började diverse ramverk med syftet att optimera utvecklingsprocessen produceras, exempelvis *Scrum* och *Extreme Programming* (Cho, 2008), fortsättningsvis *XP*.

Scrum är, som ovan nämnt, ett agilt arbetssätt som används för att hantera och kontrollera komplexa produkter (Cho, 2008). Termen *Scrum* kommer från sporten rugby, arbetssättet började kallas *Scrum* efter att Takechi och Nonaka använde rugbystrategier för att beskriva utvecklingsprocesser i Japan (Cho, 2008). Tre välkända strategier inom rugby applicerades på agil utveckling och metodologin fick sitt namn. Själva processen *Scrum* utvecklades av Schwaber och Sutherland och introducerades för allmänheten för första gången vid OOPSLA konferensen 1996 (Cho, 2008).

Således kan kvalitetssäkring av produkter som levereras inom IT-branschen utföras på olika vis beroende på vilket arbetssätt projektgruppen arbetar. Projektgrupper som arbetar enligt vattenfallsmodellen fokuserar på en fas av projektet i taget, och när den pågående fasen är avslutad går man vidare till nästa fas tills produkten är färdig för leverans. Projekt som arbetar enligt vattenfallsmodellen har således en fördefinierad fas för kvalitetssäkring, vilken ligger i slutet av projektet innan leverans till kund då all kod kontrolleras och testas (Royce, 1970; Sumrell, 2007). Agila projekt arbetar inte på samma sätt som vattenfallsprojekt vad gäller kvalitetssäkring. I agila projekt finns det ingen definierad fas för kvalitetssäkring, utan kvaliteten ska kontrolleras under utvecklingens gång (Lindstrom & Jeffries, 2004).

1.2 Problemformulering

Det agila arbetssättet har, till skillnad från vattenfallsprojekt, inte definierade milstolpar eller utstakad sträcka mellan start och mål, utan faserna i agila projekt itereras flera gånger under ett projekts livstid. Det är bland annat på grund av detta arbetssätt problematik med kvalitetssäkring kan uppstå (Lindstrom & Jeffries, 2004). I en projektgrupp som arbetar efter ett agilt arbetssätt med ett nära samarbete med kunden uppstår det ofta problem med kvalitetssäringen då tester ska genomföras kontinuerligt, trots att projektet inte är färdigt (Sumrell, 2007). Agila projektgrupper arbetar i sprintar och stämmer av med kunden kontinuerligt under projektets gång. Vid kundmöten kan förutsättningar ändras utefter vad kunden vill, detta innebär att planering och utförande ändras och projektets olika faser itereras tills kundens krav uppfylls. Testfasen itereras precis som de andra faserna, men koden är inte fullständig vid varje iteration, vilket gör det problematiskt att testa funktionaliteten.

Liknande problem uppstod när Sumrell (2007) skriver om ett företag som skulle övergå från att arbeta utifrån vattenfallsmodellen till att arbeta agilt. I Sumrells fall var det att ingen tog ansvar för testningen, och projektmedlemmarna överlät testning och kvalitetssäkring åt kvalitetssäkringsgruppen. Enligt Sumrell fungerade denna strategi dåligt i ett agilt projektarbete.

Även Cho (2008) fann svårigheter i sin studie gällande kravspecifikationer i agila projekt, eftersom kunden inte alltid har full vetskap i vad som efterfrågas. Samma studie visar även problemet med otillräcklig kommunikation mellan arbetsgrupper, där samma uppgift utförs på flera håll av separata projektmedlemmar.

1.3 Syfte

Syftet med vår uppsats är att presentera hur kvalitetssäkring påverkas vid utvecklandet av produkter i projekt som arbetar med agila arbetsmetoder, samt att komplettera tidigare forskning med ny empiri och nya infallsvinklar i området.

1.4 Forskningsfråga

Utifrån syftet att presentera hur kvalitetssäkring påverkas i projekt som arbetar med agila arbetsmetoder formades således följande forskningsfråga:

Hur påverkas kvalitetssäkring av agila arbetsmetoder?

1.5 Avgränsningar

Vi har valt att avgränsa oss från agila arbetsmetodiker som exempelvis *Lean*, *ASD (Adaptive Software Development)*, *AUP (Agile Unified Process)*. Vi har istället valt att begränsa oss till de agila arbetssätten *XP*, som är den mest använda agila arbetsmetodologin (Lindstrom & Jeffries, 2004), och *Scrum*, som är ett utbrett agilt arbetssätt inom mjukvaruutveckling (Lef-fingwell, 2007; Cockburn & Williams, 2003).

1.6 Intressenter

Intressenterna till vår studie är agila projektgrupper och projektgrupper som är i övergångsfasen till ett agilt arbetssätt, och önskar veta var fokus ska ligga för att uppnå god kvalitet med det nya arbetssättet. Uppsatsen ämnar belysa de kvalitetssäkringssvårigheter ett agilt arbetssätt medför för att uppmuntra läsaren att ha kvalitetssäkring i åtanke. Läsaren behöver ingen djup teknisk kunskap, uppsatsens fokus ligger istället på arbets- och tillvägagångssätt i projektarbeten, men läsaren förväntas känna till grundläggande termer inom IT och informatik.

1.7 Centrala begrepp

Nedan följer ett antal centrala begrepp som återkommer i uppsatsen, och är därför viktiga för läsaren att tolka enligt vår intention. Följande lista beskriver begreppen så som vi använt dem:

- Kvalitetssäkring - Med kvalitetssäkring menar vi kvaliteten på informationssystemet som levereras till kund, och uppfyller följande fem aspekter (Braa & Øgrim, 1995): teknisk kvalitet, användbarhet, estetisk kvalitet, symbolisk kvalitet samt organisatorisk kvalitet.
- Utvecklingsmiljö - Med utvecklingsmiljö menar vi den mjukvara produkter utvecklas i.
- Utvecklingsprojekt - Med utvecklingsprojekt menar vi ett projekt som utvecklar en produkt åt en kund.

2 Litteraturgenomgång

Kommande avsnitt inleder vi med att kort förklara vad agil utveckling innebär samt går djupare in på de två vanligaste arbetsätten (Lindstrom & Jeffries, 2004; Leffingwell, 2007; Cockburn & Williams, 2003). Detta görs för att ge läsaren en helhetsförståelse att använda som grund då vi senare går in på mer specifika arbetsmetoder. Vidare presenterar vi vad vår undersökning säger om kvalitetssäkring och vi väver samman detta med arbetsmetoderna. Utifrån teorin framgår det en rad centrala beståndsdelar som vi använder som grund för att skapa ett eget teoretiskt ramverk samt en tillhörande ramverksmodell. Slutligen motiveras val av teori samt utformandet av det teoretiska ramverket under ett trovärdighetsavsnitt.

2.1 Vad är agil utveckling?

Agil utveckling är samlingsnamnet för en grupp mjukvaruutvecklingsmetoder där krav och lösningar utvecklas genom ett kontinuerligt samarbete kunden och utvecklaren emellan. Agila arbetsmetoder är framtagna för att utvecklingsföretag snabbt ska kunna anpassa sig och sin utveckling utefter marknadens krav och ständiga rörelse (Highsmith & Cockburn, 2001). Bland annat karaktäriseras agil utveckling genom små och frekventa lösningsframtagande, vars kod noggrant och kontinuerligt testas (Cao et al., 2009). En del organisationer väljer att använda sig av agila arbetsmetoder för att få konkurrensfördel, medan andra är mer skeptiska till dem eventuella fördelar som agil utveckling kan resultera i (Barlow et al., 2011).

Agila arbetsmetoder anses vara bäst lämpade åt små till medelstora projekt (Cao et al., 2009; Highsmith & Cockburn, 2001). Trots detta började även större utvecklingsföretag inse behovet av ett agilt förhållningssätt i stora projekt (Lyytinen & Rose, 2006; Mathiassen & Pries-Heje, 2006) på grund av tidspress, oklara användarkrav, samt föränderlig utvecklingsmiljö (Cao et al., 2009). Kravspecifikationer från kund förändras kontinuerligt under ett projekts livscykel, då förväntningar från beställaren har förändrats (Austin & Devin, 2009).

2001 samlades en grupp utvecklare för att tillsammans etablera grundprinciperna för agila arbetsmetoder (Beck et al., 2001) eftersom tidigare hade forskare problem att enas om vad konceptet flexibla arbetsmetoder innebar (Sarker et al., 2009). Dessa grundprinciper summeras i tabell 1:

Tabell 1: Agila principer (Beck et al., 2001)

Agil princip	Beskrivning
Kundsamarbete istället för kontraktsförhandling	Minskar formaliteter för att starta och avsluta snabbare, med ett stort fokus på kunden genom utvecklingsprocessen

Individer och interaktioner istället för processer och verktyg	Förbättra kommunikationen inom grupper och bryta barriärer
Fungerande mjukvara istället för omfattande dokumentation	Utvecklare spenderar mer tid med kodning och testning istället för att skriva utförlig dokumentation
Reagera på förändringar istället för att följa en plan	Ger grupper friheten att göra förändringar och justeringar utefter projektets behov

2.1.1 Extreme Programming

XP är den mest använda agila arbetsmetodologin (Lindstrom & Jeffries, 2004). Då andra populära arbetsmetoder försöker besvara frågan ”Vilket är det bästa arbetssättet för ett utvecklingsprojekt?”, försöker *XP* istället endast besvara ”Vilket är det enklaste arbetssättet för ett utvecklingsprojekt, och vad krävs för att realisera det?”. *XP* är ett antal fristående tillvägagångssätt, som tillsammans blir ett kraftfullt verktyg när det används i projekt med hög föränderlighet. Precis som med traditionella arbetsmetoder råder det skilda meningar kring vilken projekttyp *XP* bör användas i (Lindstrom & Jeffries, 2004; Mangalaraj et al., 2009).

XP fungerar genom att samla samtliga ur utvecklingslaget och under enkla förhållande tillsammans ge feedback på utförd kod, som en form av kodgranskning i grupp. Varje deltagare i ett *XP*-projekt ses som en medlem i det gemensamma laget, och laget ansvarar för och hantear hela projektet. Kunden står i fokus för *XP*-projekt och en eller flera företagsrepresentanter sitter aktivt och arbetar tillsammans med laget varje dag (Lindstrom & Jeffries, 2004; Mangalaraj et al., 2009).

I *XP*-projekt ligger stor fokus på företagsvärde och laget producerar en serie små, fullt integrerade produktversioner som uppfyller samtliga test som kunden specificerar vilket görs för att säkra kvaliteten på produkten. Kärnmomenten inom *XP* är lagsammanhållning, planeringsspel, små produktsläpp och acceptanstester, och för varje kärnmoment finns det specifika rekommenderade riktlinjer. Deltagare i *XP*-projekt arbetar tillsammans i par eller som en mindre grupp där enkel design och kod testas regelbundet. *XP*-lag håller systemen integrerade och körbara konstant, och kodar utifrån specifika namnkonventioner så att samtliga i laget kan förstå och förbättra koden om det behövs (Lindstrom & Jeffries, 2004; Mangalaraj et al., 2009).

Parprogrammering

Parprogrammering innebär att programmerare delar upp sig i par, där en antar rollen som "föraren" och den andra "navigatören" (Hannay et al., 2009). Rollerna samt paren byts frekvent ut för att minska kunskapsklyftor samt öka utbytbarheten inom laget. Föraren skriver samtlig kod under tiden som navigatören överser och dirigerar arbetet. Detta tillåter föraren att lägga

full fokus på att skriva kod av hög kvalitet, då navigatören agerar säkerhetsnät och guide (Cao et al., 2009).

Planning game

Planning game är ett planeringsmoment inom XP. Momentet tar formen av ett möte och utförs en gång per iteration, vanligtvis en gång i veckan (Melnik & Maurer, 2004). Momentets mål är att vägleda en produkt från idé till leverans utan konkreta leveransdatum. Planeringsprocessen är indelad i två delar:

- Releaseplanering, som fokuserar på vilka krav som tillhör vilken release, samt när de bör levereras. Både kund och utvecklare är involverade i denna del, som består av tre faser:
 - *Exploration phase*: Kunden bidrar med en kort lista över krav med hög prioritet. Dessa skrivs ned på användarfallskort.
 - *Commitment phase*: Företaget och utvecklare har full fokus på funktionaliteten som ska utvecklas samt det preliminära datumet till nästa release.
 - *Steering phase*: Här kan planen förändras, det vill säga krav kan omformuleras, läggas till eller tas bort.
- Iterationsplanering, som fokuserar på aktiviteter och uppgifter för utvecklare. I denna del är kunden inte involverad, och även denna består av tre faser:
 - *Exploration phase*: Krav översätts till olika uppgifter, som skrivs ned på uppgiftskort.
 - *Commitment phase*: Uppgifterna fördelas till programmerare och en tidsestimering etableras.
 - *Steering phase*: Uppgifterna utförs och slutresultatet jämförs med det användarfall som först skrevs.

Kollektivt ägande

Kollektivt ägande innebär att samtliga utvecklare i projektet besitter kunskap inom alla projektdelar (Fitzgerald et al., 2006). Projektdelar består av både programmerings- samt företagskunskap. Detta utförs i syfte att projektdeltagarna ska känna ansvar för projektet i sin helhet istället för enskilda delar.

Intern kund

Intern kund är en metod som används då projektet saknar direkt kontakt med kunden. En utvald projektdeltagare åtar sig rollen som "kund" och agerar därefter (Holmström et al., 2006). Rollspelets syfte är att bistå med infallsvinklar på kundkraven som annars kan falla bort vid avsaknad av direkt kundkontakt.

Små produktsläpp tillsammans med testdriven utveckling

Idén bakom små produktsläpp är att en fungerande produkt skall regelbundet uppdateras för ge en bild åt kunden av hur arbetet fortskrider (Lindstrom & Jeffries, 2004). Små produktsläpp tillåter även kontinuerlig testning då det blir färre funktioner att testa vid varje etapp.

2.1.2 Scrum

Utvecklingsmetoden *Scrum* är en agil arbetsmetod som kan användas för att hantera och kontrollera komplex mjukvara (Cho, 2008). *Scrum*-metoden utvecklades av Schwaber & Sutherland, och Schwaber formaliserade metoden för användning till systemutveckling under början av 1990-talet (Schwaber & Sutherland, 2002). Sedan slutet av 1990-talet har *Scrum* blivit ett utbrett agilt arbetssätt inom mjukvaruindustrin (Leffingwell, 2007; Cockburn & Williams, 2003), och som är kraftfullt och lätt att ta sig ann (Willson, 2009). *Scrum* använder sig av iterativ och inkrementell praxis, där slutprodukten är objekt-orienterad mjukvara. Utvecklingsmetoden fokuserar på ett flexibelt och iterativt arbetssätt med stora inslag av transparens, synlighet och samarbete både inom arbetsgruppen samt med kunden (Cho, 2008; Kautz et al., 2014), för att enklare bidra till en hög kvalitetsgrad. Inom arbetsgruppen väljs en så kallad *Scrum master*, som har ansvaret att bistå arbetsgruppen med ledarskap, motivation samt bidra med kunskap till vilken prioritet och hur utvecklingsmetoderna inom *Scrum* bör gå till väga.

Utvecklingsmetoderna inom *Scrum* struktureras utefter en lista, även kallad *product backlog*, med dokumenterade krav på tekniska funktioner och användarfunktioner som sorteras efter prioritering. Denna lista förändras ständigt i samband med att kunden lyfter fram nya krav eller önskemål om förändring. Varje krav har även en notering om vilken *release* som kravet tillhör, en notering som benämns som *release backlog* (Cho, 2008; Kautz et al., 2014).

Scrum-projekt delas upp i mindre delar som kallas *sprints*, som alla består av utsatta utvecklings- och/eller processmål. De mål som ska uppfyllas för varje *sprint* samlas i en lista, även kallad *sprint backlog*. Målen definieras i ett planeringsmöte mellan arbetsgruppen, *Scrum master*, samt utomstående intressenter. Varje *sprint* har fördefinierade tidsramar som vanligtvis varar mellan 5 och 30 kalenderdagar (Schwaber & Beedle, 2002). Efter varje *sprint* utförs en så kallad *sprint review*, där arbetsgruppen, *Scrum master*, produktägaren, ledningsmedlemmar, samt representanter från kund tillsammans diskuterar vad som har utförts, resultatet, samt lärdomar från eventuella misslyckanden.

Dagliga avstämningsmöten, så kallade *Scrum*-möten, utförs för att säkerställa att arbetsgruppen är synkade mot projektet samt gentemot andra i arbetsgruppen. När produkten har levererats till kund hålls ett slutgiltigt, retrospektivt möte mellan arbetsgruppen, *Scrum master*, och eventuellt produktägaren. Det retrospektiva mötet har samma agenda och struktur som *sprint review* (Cho, 2008; Cho et al., 2011; Kautz et al., 2014).

Formell kodgenomgång

När en funktion anses vara färdigutvecklad kontrolleras koden av en annan utvecklare. Detta för att belysa fel eller misstag som programmeraren kan ha gjort. Funktionen ska även testas på båda utvecklarens arbetsstationer för att säkerställa att den slutgiltiga koden fungerar på ett kvalitetssäkert sätt (Cho et al., 2011).

Planeringspoker

Planeringspoker är en arbetsmetod där medlemmar i arbetsgruppen kan lyfta fram åsikter och frågor för diskussion anonymt, där diskussionen ska mynna ut i utvärdering av projektets arbetsgång. Deltagarna skriver ned diskussionsämnen relaterade till projektet på kort, som sedan läses upp högt och diskuteras i arbetsgruppen (Molokken-Ostvold & Haugen, 2007).

2.1.3 Fördelar och nackdelar med agil utveckling

Barlow et al. (2011) understryker styrkorna med att använda agila arbetsmetoder. En av de huvudsakliga fördelarna är den flexibilitet och anpassningsbarhet som agila metoder bidrar med (Austin & Devin, 2009). Eftersom designfasen inte är formell, och programmerarna arbetar i korta intervaller med mindre milstolpar, kan stora krav på projektet förändras under utvecklingens gång utan en större produktivitetsförlust (Barlow et al., 2011). Agila arbetsmetoder framhäver även ett fokus på kundens behov och krav, och på grund av anpassningsbarheten kan en kunds nya önskemål ofta realiseras i projektets nästa utvecklingsiteration. Dock har kunden inte full vetskap i vad som efterfrågas, vilket i sin tur leder till luddiga kravspecifikationer (Cho, 2008). Användare noterar att agila arbetsmetoder hjälper dem att behålla fokus på kundens behov och förändrande viljor genom projektets livscykel (Cockburn & Highsmith, 2001). Agila arbetsmetoder kan även leda till snabbare utveckling beroende på projektets storlek. Utan en detaljerad designfas eller dokumentation att utgå ifrån genom projektet kan gruppen fokusera på själva utvecklingen, det vill säga skriva kod och produkttestning (Austin & Devin, 2009). I små och mellanstora projekt, där gruppen utvecklar mjukvara med få iterationer, leder agil utveckling till en kortare utvecklingscykel än om projektgruppen använt planbaserade metoder (Barlow et al., 2011).

Även om agila arbetsmetoder har blivit populära att arbeta efter i små och medelstora projekt har svagheter identifierats i framförallt större projekt (Rising & Janoff, 2000), det vill säga projekt som innefattar mer än 20 personer (Cao et al., 2009). Med agila arbetsmetoder påbörjas utvecklingen innan en väldefinierad kravspecifikation är framtagen, och i större, mer komplexa projekt kan vitala delar glömmas bort eller missförstås, vilket leder till mer arbete senare i projektets livscykel (Barlow et al., 2011). Avsaknaden av en detaljerad projektplan kan även leda till problem när tid och resurser ska estimeras för projektet, eftersom oförutsägbara kravförändringar kan uppstå och därmed rubba estimeringar. I små och medelstora projekt är detta en acceptabel risk, men i större, mer komplexa projekt är risken mer ödesdiger (Cao et al., 2009). I säkerhetskritiska projekt, exempelvis inom sjukvård eller bankvärlden, kan agila arbetsmetoder anses vara osäkra och är inte tillräckligt testade (Cho et al., 2011).

Enligt studien av Cho et al. (2011) fungerade *Scrum* bra i det större projektet. Arbetsmodellerna främjade kommunikation, teamwork, och hjälpte att överskåda arbetsfördelning samt projektets utveckling. Arbetsmedlemmarna i det större projektet kände sig mer motiverade att prestera och mer intresserade av att arbeta i projektet. I samma projekt framfördes dock klagomål över att de dagliga *Scrum*-möten höll dålig fokus och varade för lång tid. I det mindre

projektet från samma studie föredrog de *Scrum* över deras tidigare användande av vattenfallsmetoder. Precis som i det större projektet kände projektmedlemmarna sig mer involverade och hade bättre överblick av utvecklingen, och även här framfördes det liknande kritik mot *Scrum*-mötena.

Agila arbetsmetoder uppmuntrar spontana kommunikationslinjer, vilket underlättar kommunikation mellan projektmedlemmar. I större projekt kan detta leda till problem då det kan finnas externa intressenter som inte har nödvändiga kommunikationsmedel (Pikkarainen et al., 2008), eller om separata arbetsgrupper har otillräcklig kommunikation som kan leda till upprepat arbete (Cho, 2008). Större projekt har även ett större behov av detaljerad dokumentation, något som agila arbetsmetoder vanligtvis inte prioriterar (Austin & Devin, 2009; Cho, 2008).

Nedan följer en sammanfattad och koncis lista över för- och nackdelar med agil utveckling (Barlow et al., 2011; Cho et al., 2011).

- Fördelar:
 - Fokus på kundbehov
 - Anpassningsbar till föränderliga krav
 - Snabb utvecklingstid
- Nackdelar:
 - Förespråkar inte formell kommunikation
 - Tid och resurser kan initialt vara okända
 - Krav är inte väldefinierade
 - Brist på dokumentation
 - Svårt att bibehålla hög säkerhetsnivå

2.2 Kvalitetssäkring

Kvalitetssäkring anses vara en term som är synonym med testning (Nabulsi, 2014). Enligt arbetsmetoden testdriven utveckling (kapitel 2.1.1) sker testning kontinuerligt under utvecklingens gång (Lindstrom & Jeffries, 2004). Enligt Cho et al. (2011) anses formell kodgenomgång (kapitel 2.1.2) i *Scrum* vara en vital och kritisk del av projekt för att skapa mjukvara av hög kvalitet. De företag som nämns i undersökningen som inte använde formell kodgenomgång ångrar det beslutet eftersom det tydligt ledde till ett resultat av sämre kvalitet. Kvalitetssäkringsaktiviteter borde vara en del av de standarder som företag förhåller sig till, som delvis kan bestå av mjukvarutestning (Nabulsi, 2014). Enligt Lindstrom & Jeffries (2004) räcker det inte att endast bygga ett test, utan testet måste även köras för att uppnå kvalitet.

Enligt ISO-standard 9000:2005 3.2.11 (ISO, 2005) är kvalitetssäkring en del av kvalitetshantering som fokuserar på att säkerställa att kvalitetskraven blir uppfyllda. Med *XP* förekommer tekniken *planning game* (kapitel 2.1.1) som en del av att säkerställa kvalitetskraven från kund (Melnik & Maurer, 2004). Enligt Cho et al. (2009) fungerade *planning game* väl i ett av

studiens projekt, men det poängterades dock att framgången var beroende av god kommunikation med kund genom projektets gång. I ett annat av studiens projekt fungerade det betydligt sämre eftersom metoden saknade stöd från ledningen då de krävde mer exakta datum och estimeringar inom projektet. *XP* använder sig även av tekniken intern kund (kapitel 2.1.1), där kundkraven förmedlas till utvecklarna på ett naturligt vis tillsammans med nya infallsvinklar (Holmström et al., 2006). Cao et al. (2009) ger exempel på projekt av varierande storlekar där användandet av intern kund fungerat väl med positiva resultat. Med *product backlog* dokumenteras krav från kund till *Scrum*-arbetsgruppen, och i *release backlog* noteras vilken *release* som kravet tillhör, för att säkerställa att kundens krav uppfylls (Cho, 2008; Krautz et al., 2014).

Parprogrammering (kapitel 2.1.1) används i *XP* med syftet att sprida kunskap mellan utvecklare samt hålla kvaliteten högre genom att motverka tunnelseende (Hannay et al., 2009). Cao et al. (2009) fann i sin studie att parprogrammering hade varierande resultat. I ett av projekten ansågs det vara en stor källa till kunskapsspridning, men i två andra projekt i studien provades parprogrammering på i en kortare period innan det övergavs då det ansågs vara ineffektivt. Kollektivt ägande (kapitel 2.1.1) inom *XP* bidrar till att arbetsgruppens medlemmar känner ansvar för att samtliga delar i projektet ska vara av god kvalitet (Fitzgerald et al., 2006). Enligt Cao et al. (2009) fungerade kollektivt ägande väl i samtliga projekt och projektmedlemmar berättar om hur de känt sig mer delaktiga.

Enligt Cao et al. (2009) finns det tre centrala beståndsdelar som ska uppfyllas för att kvalitetssäkring ska uppnås vid användandet av *XP*. Även Iivari & Iivari (2011) stärker dessa centrala beståndsdelar, som främst grundas i företags- och ledningsstruktur, och presenteras här i tabell 2:

Tabell 2: Centrala beståndsdelar för kvalitetssäkring i *XP* (Cao et al., 2009; Iivari & Iivari, 2011)

Id	Arbets titel	Central beståndsdel
A	Högre chef	Ge fullständigt stöd för användning av agila arbetsmetoder
B	Projektledare	God kunskap och förståelse för varje enskild individ och dess unika personlighet
C	Utvecklare	Ta ansvar för sitt ökade självstyre

Cho et al. (2011) listar 14 centrala beståndsdelar vid användandet av *Scrum* för att säkerställa kvaliteten i projektet. Återigen så styrker Iivari & Iivari (2011) de centrala beståndsdelar som grundar sig i företags- och ledningsstruktur. Vidare så styrker Dingsøy et al. (2012) de centrala beståndsdelar som lutar mot personligheter och kunskapsspridning. Slutligen så styrker

Hoda et al. (2011) de centrala beståndsdelar som påverkas av kommunikation till kund samt ökat självstyre bland utvecklare. Dessa centrala beståndsdelar presenteras i tabell 3:

Tabell 3: Centrala beståndsdelar för kvalitetssäkring i Scrum (Cho et al., 2011; Dingsøy et al., 2012; Hoda et al., 2011; Iivari & Iivari, 2011)

Id	Central beståndsdel
1	Längden och reglerna för det dagliga <i>Scrum</i> -mötet bör hållas strikt, för att undvika irrelevanta samtalsämnen. Längden på övriga <i>Scrum</i> -möten bör dynamiskt förändras utefter agendan.
2	Projektledare bör uppmuntra samarbetet mellan utvecklare och <i>quality assurance</i> -personal, fortsättningsvis QA. Utvecklarna bör kunna göra enhetstest på andra utvecklarens kod, samt ha nära samarbete med QA-personal under integrationstest.
3	Organisationer bör uppmuntra utvecklare att samtlig kod bör vara testbar samt designad därefter.
4	Formell kod-standardisering ökar läsbarheten samt förståelsen av andras kod, men ett för stort antal standardiseringar hindrar utvecklarens initiativ.
5	För lite dokumentation är en källa till problem, framförallt i större och mer komplexa projekt. Filosofin med att minska dokumentationen avsevärt bör istället skräddarsys efter specifika projekt.
6	Placera rätt utvecklare i rätt projekt.
7	Formell kodgenomgång är vital och kritisk för hög kvalitet. Organisationer bör etablera ett effektivt tillvägagångssätt att utföra formell kodgenomgång.
8	Planeringspoker är ett enkelt, användbart och effektivt tillvägagångssätt att utvärdera projekt (Molokken-Ostvold & Haugen, 2007). Utvecklare bör bryta ned stora projekt till minsta möjliga delar för att enklare estimeras uppgifterna.
9	Användarfall är viktiga specifikationer som höjer en utvecklarens förståelse av projektet dom arbetar i. Både utvecklare och kund behöver veta hur man skriver tydliga användarfall.
10	Kundinvolvering är viktigt för ett projekts framgång. Organisationen bör bjuda in kunden att delta i beslutsfattande, men även till diverse <i>Scrum</i> -möten.
11	Om kunden är utspridd på ett flertal platser så bör organisationen anordna kundkonferens en till två gånger om året för att visa upp nya produkter samt samla in användarfeedback.

12	Öppet kontorslandskap uppmuntrar teamwork och kommunikation, men organisationer bör komma fram till metoder för att undvika att utvecklarna blir distraherade under arbetstiden.
13	I stora och komplexa projekt bör organisationen uppmuntra utvecklarna att ha samtliga kopplingar mellan moduler i projektet i åtanke under utvecklingens gång.
14	Om ett projekt kräver gedigen dokumentation, mycket planering, eller mycket initial design, kan <i>Scrum</i> fungera mindre bra om det inte används i samband med ett annat arbetssätt, exempelvis <i>Unified Process</i> (Schach, 2004).

2.3 Teoretiskt ramverk

Vi valde, utifrån de teorier och befintliga ramverk vi tagit upp tidigare (kapitel 2.2), att skapa ett eget teoretiskt ramverk eftersom befintliga ramverk inte fokuserar på specifikt kvalitetssäkring. Andra ramverk inom området hade inte heller det fokus som vi eftersträvade, exempelvis *Agile methods and QA* (Huo et al., 2004) som fokuserar på agila arbetsmetoder i allmänhet. Vårt ramverk baserades på tidigare nämnda centrala beståndsdelar och använder dessa som grund för formuleringen av våra intervju- samt enkätfrågor. Vi tog hänsyn till vad tidigare studier nämner som brister och undersöker dessa närmare i praktiken. Vårt teoretiska ramverk presenteras i tabell 4:

Tabell 4: Teoretiskt ramverk

Kvalitetsområde	Central beståndsdel	Arbetsmetod
Arbetskvalitet	[B] Projektledare: God kunskap och förståelse för varje enskild individ och dess unika personlighet.	Parprogrammering, Intern kund, Planning Game
	[C] Utvecklare: Ta ansvar för sitt ökade självstyre.	Kollektivt ägande
	[2] Projektledare bör uppmuntra samarbetet mellan utvecklare och QA-personal. Utvecklarna bör kunna göra enhetstest på andra utvecklarens kod, samt ha nära samarbete med QA-personal under integrationstest.	Parprogrammering, Små produktsläpp med testdriven utveckling, Formell kodgenomgång

	<p>[12] Öppet kontorslandskap uppmuntrar teamwork och kommunikation, men organisationer bör komma fram till metoder för att undvika att utvecklarna blir distraherade under arbetstiden.</p>	<p>Kollektivt ägande, Formell kodgenomgång, Intern kund</p>
	<p>[13] I stora och komplexa projekt bör organisationen uppmuntra utvecklarna att ha samtliga kopplingar mellan moduler i projektet i åtanke under utvecklingens gång.</p>	<p>Parprogrammering, Kollektivt ägande, Små produktsläpp med testdriven utveckling, Formell kodgenomgång</p>
	<p>[14] Om ett projekt kräver gedigen dokumentation, mycket planering, eller mycket initial design, kan <i>Scrum</i> fungera mindre bra om det inte används i samband med ett annat arbetsätt, exempelvis <i>Unified Process</i> (Schach, 2004).</p>	<p>Planning Game</p>
<p style="text-align: center;">Kodkvalitet</p>	<p>[3] Organisationer bör uppmuntra utvecklare att samtlig kod bör vara testbar samt designad därefter.</p>	<p>Parprogrammering, Små produktsläpp med testdriven utveckling, Formell kodgenomgång</p>
	<p>[4] Formell kod-standardisering ökar läsbarheten samt förståelsen av andras kod, men ett för stort antal standardiseringar hindrar utvecklarens initiativ.</p>	<p>Formell kodgenomgång</p>
	<p>[7] Formell kodgenomgång är vital och kritisk för hög kvalitet. Organisationer bör</p>	<p>Parprogrammering, Formell kodgenomgång</p>

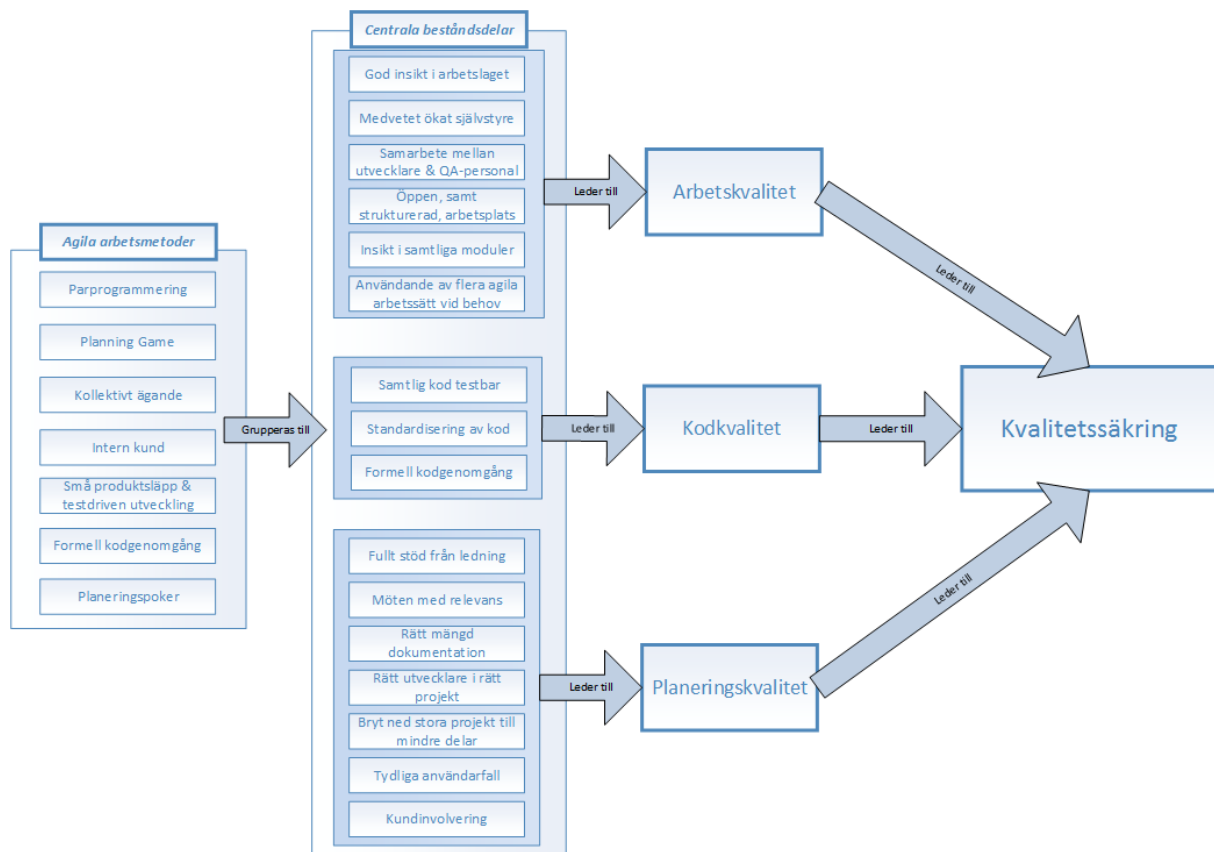
	etablera ett effektivt tillvägagångsätt att utföra formell kodgenomgång	
Planeringskvalitet	[A] Högre chef: Ge fullständigt stöd för användning av agila arbetsmetoder	Parprogrammering, Planning Game, Kollektivt ägande, Intern kund, Små produktläpp med testdriven utveckling, Formell kodgenomgång, Planeringspoker
	[1] Längden och reglerna för det dagliga <i>Scrum</i> -mötet bör hållas strikt, för att undvika irrelevanta samtalsämnen. Längden på övriga <i>Scrum</i> -möten bör dynamiskt förändras utefter agendan.	Planning Game, Planeringspoker
	[5] För lite dokumentation är en källa till problem, framförallt i större och mer komplexa projekt. Filosofin med att minska dokumentationen avsevärt bör istället skräddarsys efter specifika projekt.	Planning Game, Planeringspoker
	[6] Placera rätt utvecklare i rätt projekt	Parprogrammering, Kollektivt ägande, Intern kund
	[8] Planeringspoker är ett enkelt, användbart och effektivt tillvägagångsätt att utvärdera projekt (Moloken-Ostvold & Haugen, 2007). Utvecklare bör bryta ned stora projekt till minsta möjliga delar för att enklare estimerar uppgifterna.	Planning Game, Planeringspoker

	[9] Användarfall är viktiga specifikationer som höjer en utvecklarens förståelse av projektet som arbetar i. Både utvecklare och kund behöver veta hur man skriver tydliga användarfall	Planning Game, Intern kund, Små produktsläpp med test-driven utveckling
	[10] Kundinvolvering är viktigt för ett projekts framgång. Organisationen bör bjuda in kunden att delta i beslutsfattande, men även till diverse <i>Scrum</i> -möten.	Planning Game, Intern kund

2.3.1 Flygplansmodellen

Ramverket ovan visualiserar vi till en modell som kompletterar genom att visa strukturen samt sammanhanget mellan agila arbetsmetoder, centrala beståndsdelar och kvalitetssäkring. Vi valde att namnge den flygplansmodellen då den påminner om ett flygplan till utseendet. Förklaring till modellen kommer efter figur 1.

Figur 1: Flygplansmodellen



Modellen läses från vänster till höger, där de agila arbetsmetoderna vidare grupperas in i tre grupper med centrala beståndsdelar. En agil arbetsmetod kan grupperas till flera centrala beståndsdelar, exempelvis intern kund grupperas till god insikt i arbetslaget, öppet kontorslandskap, fullt stöd från ledning, rätt utvecklare i rätt projekt, tydliga användarfall, och kundinvolvering (tabell 4). De tre kvalitetsområdena arbetskvalitet, kodkvalitet och planeringskvalitet är direkt framtagna ur de centrala beståndsdelar som grupperats utifrån de agila arbetsmetoderna. Modellen visar vilka centrala beståndsdelar av ett projekt som behövs för att uppnå fullständig kvalitetssäkring enligt vårt ramverk.

3 Metod

I detta avsnitt motiverar vi vårt metodval, som utförs i form av en kvalitativ studie. Vi motiverar sedan i detalj val av företag som låg till grund för studien, samt urvalet av intervjuobjekt. Vidare beskriver vi utförandet av intervjuerna och hur de gick till väga. Varje intervjufråga motiveras sedan mot ramverket (kapitel 2.3). Sedan förklaras de åtgärder vi tog för att säkerställa undersökningskvalitet, sedan motiveras de etiska aspekter vi haft i åtanke vid utförandet. Vidare beskrivs studiens validitet samt reliabilitet, och slutligen förklaras genomförandet av analysen.

3.1 Metodval

Det finns två etablerade tillvägagångssätt vid genomförandet av en forskningsstudie; kvalitativa och kvantitativa studier (Denscombe, 2000). Den kvalitativa studien, ofta i form av intervjuer, fokuserar på insamlandet av djupgående data från ett fåtal nyckelpersoner, och den kvantitativa studien, ofta i form av enkäter, samlar istället in data med mindre djup från ett stort antal personer.

Då vi förväntade oss att svaren på frågorna skulle vara unika föredrog vi möjligheten att kunna ställa spontana följdfrågor, och motiverade därmed valet att fokusera på en kvalitativ forskningsstudie (Berg et al., 2004). Därmed kunde vi forma följdfrågor samt ställa bredare, mer omfattande frågor vilket kräver en kvalitativ studie (Jacobsen, 2002). Ett annat motiv till valet av kvalitativ forskningsstudie är att det ger en helhetsbild av situationen (Holme & Solvang, 1997).

Vid val av intervjuform så utgick vi ifrån Lantz (2013) definition av en öppen intervju. Intervjuns kontext är empatisk och utgångspunkten är förförståelse. Vidare beskriver Lantz (2013) intervjuns upplägg där följdfrågor ställs för att fördjupa förståelsen av det aktuella fenomenet.

3.2 Urval

Vi kom i kontakt med Avensia, ett mindre IT-företag med cirka 70 anställda beläget i Lund som arbetar med att utveckla e-handelslösningar åt diverse kunder i Norden. Företaget intresserade oss då de använde sig av de agila arbetsmetodikerna *Scrum* och *XP*, utvecklade mjukvara, befann sig lokalt, samt var öppna för kontakt och medverkan till vår studie.

Totalt fem informanter från två separata projekt intervjuades i vår kvalitativa studie, och informanterna presenteras i tabell 5:

Tabell 5: Informanttabel

Informant	Arbetsroll	Projekt
P1, 35år, Kvinna	Frontendutvecklare	Projekt 1
P2, 27år, Man	Backendutvecklare	Projekt 2
P3, 28år, Kvinna	Backendutvecklare	Projekt 1
P4, 29år, Man	Lösningssarkitekt	Projekt 1
P5, 39år, Kvinna	Projektledare	Projekt 1

Vi valde att intervjua personer med olika projektroller och bakgrund för att få en omfattande helhetsbild. Vi valde även informanter som arbetade i två olika projekt för att säkerställa att svaren inte var projektspecifika. Vi såg även till att få informanter av varierande erfarenhet, ålder och kön. Överblicken av projekten presenteras i tabell 6:

Tabell 6: Projektöverblick

Projekt	Arbetsroller till antal
Projekt 1	1x Projektledare 1x Lösningssarkitekt 7x Backendutvecklare 4x Frontendutvecklare Totalt: 13
Projekt 2	1x Projektledare 1x Lösningssarkitekt 5x Backendutvecklare 1x Frontendutvecklare Totalt: 8

Som projektöverblicken visar var det ett antal arbetsroller som ingick i varje projekt och vi ansåg således det viktigt att intervjua minst en representant från varje arbetsroll. Vi ville även få med projekt av varierande storlekar för att se om svaren skilde sig åt på grund av antal projektmedlemmar. Projekt 1 utvecklade ett e-handelssystem åt ett företag i Norge och projekt 2 utvecklade ett e-handelssystem åt ett företag i Sverige. Företaget i Norge var större än det i Sverige (sett till antal anställda) och även projektet samt slutprodukten var av en större skala.

3.3 Genomförande av intervjuer

Intervjuerna utfördes på Avensia i inbokade konferensrum. Vi ville att intervjuerna skulle kunna ske naturligt utan irritation eller påverkan av arbetet, därför motiverades intervjuplats utifrån att informanterna skulle kunna återvända till sitt arbete efter genomförd intervju. Vi valde att utföra fysiska intervjuer istället för att utföra dessa via telefon eller elektroniskt eftersom vi ville skapa en personlig kontakt och öka seriositeten av vår studie (Myers & Newman, 2007).

3.3.1 Motivering av intervjuguide

Vår intervjuguide grundar sig i det teoretiska ramverk vi tagit fram i tidigare avsnitt (kapitel 2.3). Frågorna är utformade och motiverade för att beröra varje kvalitetsområde som i sin tur bygger på en rad centrala beståndsdelar (kapitel 2.2). Nedan i tabell 7 visas de intervjufrågor vi tog fram samt vilket kvalitetsområde de berörde:

Tabell 7: Motivering av intervjuguide

Intervjufråga	Kvalitetsområde
Fråga 1. Hur pass engagerad och insatt anser du att dina överordnade är till projektet du arbetar i för tillfället?	Planeringskvalitet
Fråga 2. Har du personligt intresse i de projekt du arbetar i?	Planeringskvalitet
Fråga 3. <i>Projektledare</i> : Hur pass god insikt i varje projektmedlems kunskapsgrad/personlighet anser du att du har? <i>Projektmedlem</i> : Hur pass god insikt i din kunskap/personlighet anser du att din projektledare innehar?	Arbetskvalitet
Fråga 4. Finner du att du tar egna initiativ eller inväntar du delegation från överordnad?	Arbetskvalitet
Fråga 5. Hur ofta har ni projektmöten?	Planeringskvalitet
Fråga 6. Hur fungerar samarbetet mellan utvecklare och kvalitetssäkringspersonal?	Arbetskvalitet

Fråga 7. Vilka eventuella riktlinjer följer ni för att skriva testbar kod, har ni någon kodstandard?	Kodkvalitet
Fråga 8. Sker det något formell kodgenomgång?	Kodkvalitet
Fråga 9. <i>Utvecklare</i> : Har du övriga kodmoduler i åtanke vid din egen utveckling?	Arbetskvalitet
Fråga 10. Hur ser dokumentationen ut i era projekt?	Planeringskvalitet
Fråga 11. Hur angriper ni större projektuppgifter?	Planeringskvalitet
Fråga 12. Vem formulerar era användarfall?	Planeringskvalitet
Fråga 13. Hur involverad är kunden i era projekt?	Planeringskvalitet
Fråga 14. Hur fungerar ert kontorslandskap?	Arbetskvalitet

Nedan motiveras varje fråga. Då frågorna är indelade i olika kategorier samlas dessa under tillhörande kvalitetsområde för att förtydliga sambandet mellan ramverk och intervjufrågor.

Arbetskvalitet

För att kontrollera projektledarens insikt i varje projektmedlems individuella kunskaper och personlighet (tabell 3:B) utformades en fråga som ställdes beroende på informantens arbetsroll (tabell 7:Fråga 3). Vore informanten utvecklare var frågan formad att efterfråga den uppfattade insikten av informantens personlighet och kunskap från projektledaren. Vore informanten projektledare vändes frågan till att istället undersöka informantens grad av insikt i arbetslaget. Det teoretiska ramverket säger att det även är viktigt att kontrollera i vilken utsträckning projektmedlemmarna tar egna initiativ och ansvar för sitt resultat, därför efterfrågades informantens grad av initiativtagande kontra delegationsgrad från överordnad samt vad informanten personligen föredrog (tabell 3:C; tabell 7:Fråga 4).

Kvalitetssäkringspersonal är en viktig del av ett projekt som vill leverera hög kvalitet (tabell 4:2). Vi ville därför undersöka om det fanns specifikt utsatt kvalitetssäkringspersonal och om inte, ifall företaget ansåg att det fanns ett behov av sådan (tabell 7:Fråga 6). En fråga som berör medvetandet om övriga kodmoduler ställdes (tabell 7:Fråga 9) för att undersöka hur inkapslat utvecklare tänker under utvecklingstiden (tabell 4:13). Detta för att kontrollera om utvecklaren exempelvis skapar onödiga beroenden eller om utvecklaren gör det enkelt att bygga vidare på modulen. Det finns olika kontorslandskap, och agil utveckling förespråkar ett öppet

kontorslandskap för att främja kommunikation och samarbete (tabell 4:12). Vi ville undersöka hur företagets kontorslandskap såg ut, samt hur det ansågs påverka det dagliga arbetet eftersom olika individer påverkas på olika sätt av deras arbetsmiljö (tabell 7:Fråga 14).

Kodkvalitet

För att undersöka kodkvaliteten inom projekten formades frågor utifrån hur utvecklare skrev testbar kod, det vill säga om de har testbarhet i åtanke vid utveckling (tabell 4:3; tabell 7:Fråga 8). Vi kontrollerade om det skedde någon formell kodgenomgång efter utvecklandet av moduler och funktioner (tabell 4:7; tabell 7:Fråga 8). Vi ville även se om projektet följde någon kodstandard, det vill säga en gemensam överenskommelse gällande utformningen av kod (tabell 4:4; tabell 7:Fråga 7).

Planeringskvalitet

Vi ville skapa en omfattande och strukturerad helhetsbild av företaget. Detta involverade att ta reda på hur pass engagerade och intresserade av projekten som den övre ledningen var (tabell 3:A). Detta mätte vi genom att fråga informanterna om deras åsikt och uppfattning gällande ledningens engagemang (tabell 7:Fråga 1). Vi ville även mäta individens intressenivå av det projekt den arbetar i för tillfället, samt om det eventuellt fanns andra projekt de hellre suttit med (tabell 7:Fråga 2). Detta gjordes för att kontrollera om rätt person placerats i rätt projekt (tabell 4:6). Relevanta och välplanerade möten är väsentliga för att skapa en produkt av god kvalitet (tabell 4:1) vilket vi undersöker genom att fråga om informantens åsikt kring frekvens, relevans och planering av projektmöten (tabell 7:Fråga 5).

För lite dokumentation kan vara en källa till problem (tabell 4:5), och därför formades en fråga för att undersöka dokumentationsgraden i informantens projekt (tabell 7:Fråga 10). Vi ville även undersöka om den mängd dokumentation avsågs vara tillräcklig. Vissa organisationer använder sig av planeringspoker för att hantera större projektuppgifter (tabell 4:8), och för att undersöka detta frågade vi öppet om hur informanten hanterar stora projektuppgifter (tabell 7:Fråga 11). Användarfall är nödvändiga specifikationer inom projekt och det är viktigt att både utvecklare och kund kan formulera dessa på ett tydligt vis (tabell 4:9). Vi formulerade en fråga för att kontrollera hur detta sker, samt om individerna är bekväma att formulera dessa (tabell 7:Fråga 12). Agila projekt kräver en god kundinvolvering för att säkra hög kvalitet (tabell 4:10), därför formulerade vi en fråga för att ta reda på hur pass involverad kunden är i informantens specifika projekt samt informantens åsikt gällande om kundens involvering borde öka (tabell 7:Fråga 13).

För att få ett bredare perspektiv och mer givande svar formulerades även ett antal följdfrågor, som samtliga presenteras med sina huvudfrågor i vår intervjuguide (bilaga 7.2).

3.4 Undersökningskvalitet

För att säkerställa att ingen information gick förlorad spelades samtliga intervjuer in med tre separata mobiltelefoner med tillhörande diktafonapplikationer. För att ljudupptagningen inte

skulle ta med vibrationer i bordet lades två av maskinerna på dämpande underlag. Ljudisolerade och enskilda rum bokades för varje intervju för att minimera distraktioner och eventuella avbrott.

Istället för att boka in intervjuerna på löpande band såg vi till att ha gott om tid emellan till att reflektera och ladda batterier inför nästa informant. Vi såg över frågor med tillhörande svar från första intervjun och valde att finslipa betoningen och formuleringen på tabell 7:Fråga 1, för att säkerställa att svaret gällde den övre ledningen. Varje intervju inleddes på ett avslappnande vis för att mjuka upp stämningen och avdramatisera intervjun, i ett steg att inge förtroende och få ut ärliga svar. Varje intervju avslutades även med en slutgiltig och öppen fråga, utifall informanten kände att det var någon information den ville få sagt innan intervjun avslutades.

3.4.1 Etik

Varje intervju inleddes med en presentation av oss: vem vi var, vad vi studerade, och vad det var för forskningsämne på vår uppsats. Vi var tydliga med att förklara vårt huvudsakliga fokus och syfte med intervjun samt dess betydelse för vår undersökning, detta gjordes för att undvika all form av tvetydighet. Det var viktigt för oss att informanterna kände sig bekväma att besvara våra frågor på ett ärligt vis, och att inte undanhålla relevant information. Därför informerade vi informanterna om att deras transkript skulle skickas till dom för godkännande och att de hade möjligheten att ta tillbaka eventuell information som dom ansåg vara felaktig eller känslig. Varje informant informerades även om att deras identitet, förutom ålder, kön och arbetsroll, skulle hållas anonym för att skydda deras personliga integritet och anseende på företaget. Även kunden i deras projekt anonymiserades för att undvika eventuella problem med kundrelationer.

3.4.2 Studiens validitet och reliabilitet

För att uppnå hög objektivitet hade vi format frågorna så öppna som möjligt och vid endast ett fåtal tillfällen ombads vi av informanten att utveckla frågan. I vissa fall hade informanten omedvetet besvarat en följdfråga som vi valde att upprepa för att säkerställa frågans validitet. Vi valde även att hålla intervjuerna med enskilda personer för att undvika att yttre faktorer påverkade deras svar.

Under intervjuernas gång såg vi till att hålla en låg profil för att ge informanten tid på sig att ge ett utförligt svar (Jacobsen, 2002). Genom att ta en kortare paus på cirka tre till fem sekunder efter varje besvarad fråga lät vi informanten tala till punkt och få sagt det den ville utan att känna stress.

3.5 Datainsamling och analyser

Transkripten gavs transkriptnummer och radnumrering för att enkelt kunna hänvisa till rätt transkript och rad vid redovisandet av resultatet. Om det uppstod komplikationer med ljudfilen under transkriberingen fanns det två ljudfiler till med olika ljudformat att använda sig av. Under transkriberingen av intervjuerna har vi valt att ta bort irrelevanta ljud eller avbrott.

Vid analysen av data övervägde vi användandet av mjukvaran *QSR NVivo* men ansåg att mängden data gjorde att detta program kändes överflödigt, då *NVivo* riktar sig åt studier av en större karaktär. Analysen utgick istället från det ramverk som låg till grund för intervjuerna, och svaren kategoriserades utefter de centrala beståndsdelarna i ramverket. Eftersom svaren varierade mellan informanterna valde vi att mer ingående redogöra varje svar, istället för att återge ett generellt svar.

3.6 Trovärdighet

För att förbättra kvaliteten och fokus på uppsatsen har vi endast utgått ifrån artiklar som håller sig inom ramarna för informationssystem samt artiklar som har blivit referensgranskade, så kallade *peer-reviewed*. Vårt ramverk grundas på forskning utförd av etablerade och erkända aktörer inom ämnet, detta för att säkerställa relevansen och validiteten. De två författare vars forskning vårt ramverk influeras mest utav är Juyun Cho samt Lan Cao. Cho har en Ph.D. inom Management Information Systems vid Utah State University och har publicerat nio *peer-reviewed* artiklar inom ämnet informatik i bland annat *Issues in Information Systems* och *Journal of Computer Information Systems*. Cao har en Ph.D. inom Computer Information Systems och har publicerat över 20 *peer reviewed* artiklar inom ämnet informatik i bland annat *European Journal of Information Systems* och *Journal of Management Information Systems*. Detta ramverk byggde vi sedan vår metod och intervjumall utefter.

4 Resultat av empirisk data

Detta kapitel inleds med en kort presentation av vår uppfattning av företaget utifrån hur det presenterades för oss under vår studie. Vidare presenteras resultatet av de intervjuer vi utfört på företaget i form av vad informanterna anser vara fördelar samt nackdelar med deras arbetssätt. Sedan redogörs det vilken roll varje central beståndsdel från vårt ramverk (kapitel 2.3) spelar i praktiken enligt informanterna.

4.1 Företaget Avensia

Följande presentation av Avensia grundades i information vi samlade vid utförandet av den kvalitativa studien. Samtliga projekt på Avensia arbetade utefter ett agilt arbetssätt, och företaget arbetade i ett flertal projekt vid samma tillfälle. Varje projekt hade en rad olika arbetsroller med projektledare, lösningsarkitekt, backend- samt frontendutvecklare. Projekten satt tillsammans i grupper i ett öppet kontorslandskap där separata konferensrum kunde bokas, och kontoren var belägna i centrala Lund samt Helsingborg. Kommunikation inom projekten relaterat till förändringskrav och buggrapportering, både internt och mot kund, skedde via systemen JIRA och TFS (*Team Foundation Server*).

4.2 Generell upplevelse och attityd

Samtliga fem informanter hade en positiv inställning till det agila arbetssättet på Avensia, dock ansåg de själva att det eventuellt fanns vissa problem vid kvalitetssäkring. Vi uppfattade det som att merparten av de begrepp och frågor som vi använde inte ansågs vara främmande för informanterna. I de fåtal fall informanterna inte förstod frågan uttryckte de detta och vi var då noga med att specificera frågan ytterligare. Vår uppfattning var att informanterna var intresserade av ämnet och gladeligen besvarade våra frågor, vi ansåg även att intervjuerna var avslappnade och vänskapliga. Vi uppfattade det som att vi fick gott om tid, och vi kände aldrig att vi behövde pressa informanterna på svar. Informanterna var införstådda i att intervjuerna fick ta den tiden de tog och vi kände oss aldrig stressade eller att någon informant inte ville vara där. Alla informanterna godkände utan invändningar att intervjuerna spelades in och ingen av informanterna hade några invändningar på transkripten vi skickade till dem för godkännande.

4.3 Centrala beståndsdelar

Varje central beståndsdel från ramverket (kapitel 2.3) presenteras nedan utifrån informanternas uppfattning. En sammanfattning följer efter varje avslutat kvalitetsavsnitt.

4.3.1 Arbetskvalitet

Nedan följer intervjuresultat från de centrala beståndsdelar relaterat till kategorin arbetskvalitet (kapitel 2.3).

God insikt i arbetslaget

Fyra av informanterna tillfrågades hur bra insikt de ansåg att deras projektledare hade i deras personlighet och kunskap. Svaren varierade från ”inte jättebra” (P2, rad 14) till ”våldigt bra” (P3, rad 14), men det nämndes även att det var ”jättesvårt” (P4, rad 23) samt att det inte är ”så svart eller vitt” (P2, rad 16). En informant nämnde även att det var deras första projekt tillsammans med projektledaren och att ”i början var det en del upplärningstid innan man kände varandra” (P1, rad 14), men att det ändå gavs ”mycket beröm och kritik” vid fram- och motgångar (P1, rad 16). Projektledaren tillfrågades om den ansåg att den hade god insikt, gällande kunskap och personlighet, i sitt arbetslag, något den besvarade med ”god koll, men skulle absolut kunna bli bättre” (P5, rad 13). Projektledaren hade även en idé om att formalisera kompetensvetenskapen genom att ”tillsammans göra en kompetensmatris” (P5, rad 15), där individer kan öppet beskriva sina kompetenser.

Medvetet ökat självstyre

På frågan om informanterna tog egna initiativ var svaren relativt eniga. I vissa fall var informanterna bestämda såsom informanten som svarade ”jag tar egna initiativ, det tror jag att de flesta gör” (P2, rad 20) och en annan som svarade ”egna initiativ är jätteviktigt för det går inte att ha en projektledare som går och håller dig i handen” (P4, rad 27). Andra informanter svarade ”i den mån det går tar jag gärna egna initiativ” (P1, rad 18) samt ”jag försöker ta eget initiativ” (P3, rad 18). En annan informant tillade även att ”ibland finns det inte jättemycket möjlighet att ta egna initiativ” (P1, rad 20). Även projektledaren svarade i enighet med de andra ”jag tycker att jag tar mest egna initiativ” (P5, rad 17).

Vid frågan om de föredrog att ta egna initiativ eller att bli beordrad att göra en viss uppgift svarade de med att ”det måste finnas en jämnvikt” (P2, rad 22), ”både och. En kombination” (P3, rad 20) och ”självklart måste det ju finnas en riktpinne på vad vi ska koncentrera oss på” (P4, rad 29).

Samarbete mellan utvecklare & QA-personal

Informanterna tillfrågades om hur de anser att samarbetet mellan QA-personal och utvecklare fungerar. Frågan besvarades på liknande vis av alla informanterna i projekt ett, exempelvis; ”Vi har inte direkt någon kvalitetssäkringspersonal” (P1, rad 26), ”vi är vår egen kvalitetssäkringspersonal” (P4, rad 40). Svaret från informanten som arbetar i ett annat projekt än de övriga skiljde sig lite då han beskrev att ”vi sitter varje fredag en halv dag och gör testning. Vi har fyra timmar dedikerat till det” (P2, rad 30), vilket tyder på att det projektet har mer fokus på kvalitetssäkring. Lösningssarkitekten berättade att de lade över mycket av acceptanstestningen på deras kunder och lät dem göra den (P4, rad 40). En annan faktor lösningssarkitekten tog upp för att argumentera emot att ha personal dedikerad till kvalitetssäkring var att projektet skulle vara tvungna att utbilda den personalen i hur alla funktioner skulle fungera och vad som var okej eller inte (P4, rad 42).

Informanterna fick frågan om de tyckte att det fanns ett behov av kvalitetsäkringspersonal. En informant tyckte det skulle varit bra ”att någon har det på deltid eller att man tar in någon några dagar i slutet av en sprint” eller ”att vi själva får mera tid att kvalitetssäkra varandras uppgifter” och ”vi skulle behöva göra det mer” (P1, rad 30). En informant i projekt två tyckte det skulle vara ”bättre med renodlad” (P2, rad 32) kvalitetspersonal. En informant i projekt ett tycker samtidigt att ”det är kanske inte jättenödvändigt, men det är ju alltid bra om det finns.” (P3, rad 32). Projektledaren besvarade frågan med ”nej, jag tycker nog inte det” (P5, rad 29).

Öppen, samt strukturerad arbetsplats

För att undersöka om Avensia hade ett öppet kontorslandskap frågade vi hur deras arbetsplats såg ut. Vi ville även veta om informanterna ansåg att deras kontorsmiljö påverkade dem positivt eller negativt. Samtliga informanter besvarade frågan med att de har ett öppet kontorslandskap (P1, rad 62; P2, rad 60; P3, rad 58; P4, rad 84; P5, rad 67). Fyra av informanterna ansåg att ljudnivån var hög, men att det inte störde dem personligen (P1, rad 62; P2, rad 60; P3, rad 58; P4, rad 84). Projektledaren ansåg att det var alldeles för trångt och inte fungerade så bra, och att hon ibland går undan och stänger in sig för att fokusera (P5, rad 67). En annan informant antydde även att det fanns andra anställda som har problem med ljudnivån (P1, rad 62). Backendutvecklaren i projekt ett var betydligt mer optimistiskt inställd till kontorslandskapet, och angav även att det var en av anledningarna till att den valde att arbeta där (P3, rad 58).

På frågan om kontorslandskapet påverkade deras dagliga arbete positivt eller negativt var svaren unisona; alla ansåg att det bidrog positivt till deras dagliga arbete. Även om projektledaren ansåg att det var trångt och ostrukturerat så gav den ett positivt svar. ”Jag tror det bidrar positivt. Att man har alla omkring sig, jag gillar det. Man är med, och sitter inte i ett rum och ugglar liksom. Alla är med!” (P5, rad 69).

Insikt i samtliga moduler

Vi undersökte ifall utvecklarna hade övriga kodmoduler i åtanke vid sin egen utveckling, bland annat för att undvika onödiga beroenden, frågade vi fyra av informanterna om detta. Den femte informanten, som var projektledare, tillfrågades inte eftersom arbetsrollen inte involveras direkt i utvecklingen. Samtliga utvecklare angav att de ser modultänk som en självklarhet (P1, rad 38; P2, rad 38; P3, rad 38; P4, rad 52). ”Självklart! Det försöker man ha så mycket man kan, men det är det att man inte vet hur allting ser ut och det är alltid svårt att veta” (P4, rad 52). En av informanterna visade även upp en egenritad modell över modulerna i projektet (P2, rad 38). Backendutvecklaren i projekt ett poängterade att som ny i branschen var det inte alltid lätt, men att den försökte ha det i åtanke (P3, rad 38).

Sammanfattning

Majoriteten (tre av fem) av informanterna ansåg att det fanns god insikt i deras arbetslag från projektledaren. När det gällde viljan att ta egna initiativ i arbetet ansåg samtliga informanter att det var något de föredrog. Gällande samarbete mellan utvecklare och QA-personal var sva-

ren eniga om att det inte existerade något samarbete, eftersom det inte existerade någon specifik QA-personal. På frågan om de ansåg att deras arbetsplats var öppen och strukturerad var fyra av informanterna positiva, och endast projektledaren var obestämmd i sitt svar. Slutligen ansåg samtliga av de tillfrågade utvecklarna att de hade övriga kodmoduler i åtanke vid sin egen utveckling.

4.3.2 Kodkvalitet

Nedan följer intervjuresultat från de centrala beståndsdelar relaterat till kategorin kodkvalitet (kapitel 2.3).

Samtlig kod testbar och Standardisering av kod

För att undersöka om Avenia arbetade aktivt med att skriva testbar kod frågade vi om de följde några riktlinjer eller någon kodstandard. Frontendutvecklaren svarade med ”vi försöker köra med kodstandarder. Men jag känner i alla fall att vi inte har kommit hela vägen med detta” (P1, rad 32). Frontendutvecklaren tillade även att det var väldigt svårt att anpassa sig till någon specifik kodstandard eftersom det kommer så mycket nytt hela tiden och att det finns vansinnigt många standarder att välja mellan (P1, rad 32). De övriga informanterna svarade med att de inte följde någon kodstandard när de utvecklade (P2, rad 34; P3, rad 34; P4, rad 44). Backendutvecklaren i projekt två berättade att ”man försöker bygga saker modulärt, avgränsa saker och försöka att inte ha beroenden i onödan” (P2, rad 34), även om de inte körde efter en specifik kodstandard. Den andra backendutvecklaren berättade att det hade varit upp för diskussion inom arbetslaget att de borde följa någon kodstandard men att det inte blivit av ännu (P3, rad 34).

Lösningssarkitekten sa att projekten ibland kom överens om att köra efter en viss standard men att utvecklarna inte alltid vet vad dem innebär och att de då snabbt faller tillbaka till hur de brukar göra (P4, rad 44). Projektledaren, som inte skriver någon kod själv, svarade kort med att de inte följde någon kodstandard och att det inte var bra (P5, rad 31).

Formell kodgenomgång

Vi ville undersöka ifall Avenia använde sig av någon formell kodgenomgång i sina projekt, det vill säga att utvecklarna bistår varandra med kontroll av skriven kod för att undvika tunnelseende samt att sprida kunskap. Svaren var ganska skilda, och i ett av projekten användes det i någon form, men syftet med kodgenomgång verkar vara underförstått hos samtliga informanter. Backendutvecklaren i projekt två svarar att de använder sig av kodgenomgång, och att det även är bra för att sprida kunskap, och ”på krångliga delar av projektet så måste man gå igenom det med någon annan” (P2, rad 36). Även backendutvecklaren i projekt ett svarar med ”ja, vi har bestämt att någon annan ska kolla igenom ens kod innan man skickar in” (P3, rad 36). Frontendutvecklaren hävdar däremot helt kort att de inte använder sig av någon formell kodgenomgång (P1, rad 36). Lösningssarkitekten menar att det är olika från projekt till projekt, men att det vanligtvis används för att sprida kunskap och kontrollera vanliga fel (P4, rad 46). ”Det är mer när man gör en stor funktion eller stor ändring, då kanske det blir att folk gör det

men det är också en vana och man måste få folk trygga med det” (P4, rad 46). Lösningssarkitekten hade helst sett att kodgenomgången var mer informell, och användes som enkla, små och snabba kodgranskningar (P4, rad 48).

Projektledaren menade att det har pratats om att börja med det: ”Det har vi också pratat om att ha så kallade *code reviews*, en miniprocess för det där man kvalitetssäkrar sin egen kod genom att någon annan alltid ska titta på koden” (P5, rad 33). Projektledaren tror dock att det saknas disciplin, att utvecklarna inte använder deras interna system JIRA tillräckligt, och att i deras nuvarande projekt har kunden bett dem att använda deras system istället (P5, rad 39). På frågan om projektledaren tror att det är ofta som kunden ställer sådana krav svarar den med att det inte händer så ofta, och när det händer så är det endast när större kunder är involverade (P5, rad 41).

Sammanfattning

Resultatet visade att tanken om att skriva testbar och standardiserad kod fanns, men att det inte skedde i nuläget. Formell kodgenomgång visade sig vara någonting som alla var införstådda med innebörden av, men även denna centrala beståndsdel var bara i tankestadiet i projekt ett. Det skedde ingen formell kodgenomgång. Projekt två däremot använde sig av formell kodgenomgång.

4.3.3 Planeringskvalitet

Nedan följer intervjuresultat från de centrala beståndsdelar relaterat till kategorin planeringskvalitet (kapitel 2.3).

Fullt stöd från ledning

Vi vill undersöka om medlemmarna inom projekten ansåg att de fick fullt stöd från företagets ledning, därför formade vi en fråga angående detta samt hur de visade sitt eventuella engagemang. Svaren var relativt lika varandra men skilde sig på vissa punkter. Informanten som var frontendutvecklare ansåg att engagemanget från projektledaren var mycket stor men att informanten inte visste hur pass involverade den högre ledningen var (P1, rad 6). Även informanten från projekt två ansåg att projektledaren var engagerad trots att denne var ny i projektet (P2, rad 6), och tillade att engagemanget visades genom att projektledaren ställde frågor samt var aktiv under möten (P2, rad 8). Detta backades upp ännu en gång av backendutvecklaren i projekt ett. ”De har koll på det när man pratar med dem. De visar även intresse genom att ställa frågor om hur det går och liknande” (P3, rad 8).

Informanten som var lösningssarkitekt menade dock att platscheferna och VD:n inte hade möjlighet att vara involverade på detaljnivå i alla projekt då de ansvarade för flertalet projekt parallellt. ”Man får ju göra det rimligt liksom. Du kan ju inte ha att en person som är platschef som har kanske åtta till tio projekt under sig, ifall de ska hinna med att göra någonting annat så kan ju inte dem vara så detaljerat insatta utan det är ju oftast att de kollar liksom vad det är som händer i projektet, det är ju framförallt om någonting strular i projektet” (P4, rad 10).

Projektledaren ansåg att chefernas engagemang inte var särskilt stort utan att då de visade intresse så handlade det främst om de ekonomiska aspekterna (P5, rad 7). Projektledaren påpekar även att denne önskat att intresset från ledningen vore större (P5, rad 7), speciellt då kunden är en "väldigt strategisk kund som Avensia tjänar bra med pengar på, alltså en bra kund och en bra referenskund", något som projektledaren även avslutar med att "jag skulle nog vilja säga att jag önskar att intresset vore större speciellt för en så stor kund" (P5, rad 7).

Möten med relevans

En fråga ställdes angående hur relevanta mötena ansågs vara, frontendutvecklaren besvarade med att denne "är inte jätteintresserad av hur backendsystemen fungerar" och att "sitta och lyssna på detaljerade genomgångar av hur systemen fungerar är inte jätteintressant" (P1, rad 24), men även att "det är en form av kunskapsspridning" (P1, rad 24). Backendutvecklaren från projekt två ansåg att mötesrelevansen var "bra", men även att "jag tror att många upplever dem som inte bra" (P2, rad 26). Informanten besvarade även att "ofta har vi delat upp det på frontend- och backendmöten" och att "det är viktigt att tänka på att man inte slösar tid" (P2, rad 26). Backendutvecklaren ur projekt ett ansåg att "de är väldigt relevanta och följer planen" (P3, rad 26). Lösningarkitekten ansåg att "de flesta försöker säga nej till möten om de inte känner att de är givande" (P4, rad 38). Projektledaren ansåg att mötena "borde vara ännu mer välplanerade och ha ett ännu tydligare syfte" (P5, rad 23). Informanten ansåg även att "man skulle absolut kunna ha planerat dem lite bättre" för att "få ut mer" (P5, rad 23).

Rätt mängd dokumentation

Vi frågade informanterna angående mängden dokumentation i deras projekt samt om de ansåg att den var tillräcklig. Informanterna tycktes alla vara överens om att det inte fanns någon stor mängd dokumentation inom projekten. "Vi försöker kommentera vår kod men någon separat dokumentation görs ej" (P1, rad 43). Informanterna var överens om att det inte fanns någon överenskommelse om hur dokumentationen skulle utföras. "Vi har tid i projektet utsatt till just dokumentation men vi har inte specificerat vad för sorts dokumentation vi kommer prestera" (P5, rad 46). En informant nämnde även att bristen på dokumentation kan vara på grund av kunden, då kunder ofta vill skära ner projekt så mycket som möjligt för att spara pengar så brukar saker som dokumentation skäras bort (P4, rad 54), att dokumentera handlar om en vana, de flesta är inte vana vid att dokumentera överhuvudtaget (P4, rad 54). En informant menade på att dokumentationen fanns inom utvecklarna i form av kunskapen de höll och att det således var upp till kunden att ställa frågor och själva dokumentera vad de ansåg relevant (P4, rad 56). Det påpekades även av samma informant att det är svårt att hålla dokumentation relevant och uppdaterad (P4, rad 56).

Då vi frågade om dokumentationen de hade i nuläget var tillräcklig rädde det skilda meningar. En informant menade att mängden dokumentation kan bero på antalet användare av slutprodukten (P1, rad 43). En informant ansåg dokumentationen tillräcklig om den endast skulle användas för att få användaren att kunna använda produkten, däremot skulle det kunna uppstå svårigheter om koden går vidare till andra utvecklare för att upprätthållas (P2, rad 42). Dokumentation kan ske på olika sätt, på en informanternas föregående arbetsplatser dokumenterades det i detalj hur man återskapade varje enskild bugg medan dokumentationen informanten

skrev i dagsläget var mer allmän om hur programmet fungerade (P3, rad 40). Trots de brister som påpekas angående dokumentationen ansåg alla fyra informanter med utvecklarroller att dokumentationen var tillräcklig enligt dem. ”Dokumentationen är tillräcklig tycker jag, fokus ligger dock inte på den som jag har förstått det” (P3, rad 42). Informanten med rollen projektledare däremot höll inte med utan ansåg att det inte var tillräckligt med dokumentation inom projekten. ”Vi hade behövt mer dokumentation som inte är så detaljerad och teknisk utan mer om flöden och processer. Jag tycker att en lösningsarkitekt borde sätta sig och gå igenom det och dokumentera” (P5, rad 45).

Rätt utvecklare i rätt projekt

För att undersöka om rätt utvecklare sattes i rätt projekt frågade vi först varje informant om de kände att de hade ett personligt intresse i projektet de arbetade i och även om det fanns några andra projekt på deras arbetsplats som de hellre arbetat i. Det var endast en informant som tvekade kring sitt personliga intresse av projektet. ”Nja, just projektspecifikt skulle jag nog inte säga att jag har ett personligt intresse, visst man kanske har ett personligt intresse för tekniken, men inte den projektspecifika sidan” (P2, rad 10). Tre av de fyra resterande informanterna kände alla att de utan tvekan fann ett personligt intresse (P1, rad 10; P3, rad 10; P5, rad 9) och den sista menade att det var oundvikligt att inte få ett personligt intresse för något man spenderar så mycket tid med. ”Ja, det blir så, man sitter och engagerar sig så pass mycket med någonting så tillslut blir det så, ja det gör man” (P4, rad 16).

Vid frågan om det fanns andra projekt de hellre arbetat i var svaren mer blandade. Två av informanterna svarade konkret att det inte fanns (P2, rad 12; P5, rad 11). Två av de tre andra informanterna gav mer vaga svar, en informant spekulerade i om gräset var grönare på andra sidan (P4, rad 18), en annan tyckte det var svårt att säga då informanten visste för lite om de andra projekten (P3, rad 12). Endast en av informanterna visade intresse för övriga projekt. ”Jag vet att det är ett par som startat den senaste tiden som jag är lite sugen att hoppa in i, men inte kunnat gå in i då jag är lite inlåst i mitt nuvarande projekt” (P1, rad 12).

Bryt ned stora projekt till mindre delar

För att undersöka om Avensia hanterar stora projekt genom att bryta ned dessa till mindre delar frågade vi hur de angrep större projektuppgifter. Vi frågade även om det fanns något annat tillvägagångssätt än deras nuvarande som de hade föredragit. Bland informanterna rådde det skilda meningar kring hur en stor projektuppgift borde angripas. Två av informanterna rekommenderade att bryta ner det i mindre delar, den ena lade fokus på planering och prioritering (P1, rad 48), den andra informanten förklarade vikten av att bryta ner de större projektuppgifterna. ”Det måste du göra för att kunna göra det parallellt, för att kunna göra det övergripligt och för att kunna göra estimat så måste man bryta ner uppgiften om den är för stor” (P4, rad 58). De andra informanterna hade lite andra lösningar, till exempel med parprogrammering (P3, rad 44), eller genom att syna all kod för att undersöka hur olika saker kommer påverka varandra (P2, rad 44). Projektledaren berättar att de har en så kallad *change management*-process som innebär att man kontakter leveransansvarig för att se till att få fram resurser och backa upp en så stor förändring (P5, rad 47).

När informanterna blev tillfrågade om de ansåg att det fanns något annat tillvägagångssätt så svarade alla informanterna på liknande vis. Informanten som föreslog parprogrammering ansåg att det tillvägagångssättet är bra (P3, rad 46). De informanter som förespråkade nedbrytandet till mindre delar ansåg att det var ett bra tillvägagångssätt; det hjälpte en att kategorisera sina uppgifter mentalt (P4, rad 60), eventuellt hade det kanske varit smart att planera tillsammans med någon annan som kunnat agera bollplank (P1, rad 50). Projektledaren ansåg inte att det fanns något annat tillvägagångssätt att föredra (P5, rad 49). Även informanten som förespråkade att man i lugn och ro gick igenom sin kod var nöjd med sitt tillvägagångssätt. "Det är nog viktigt att inte agera förhastat, man vill nog gärna bara kasta sig ner och börja koda direkt, men det är viktigt att tänka efter" (P2, rad 46).

Tydliga användarfall

Vi frågade informanterna om deras användarfall i projekten med tre intresseområden: vem som formulerade användarfallen, om informanten ansåg dem vara användbara, samt om informanten hade känt sig bekväm med att skriva användarfall. Tre av informanterna angav att användarfallen skrevs av lösningsarkitekten och projektledaren under förstudien (P1, rad 52; P2, rad 48; P4, rad 62). Backendutvecklaren i projekt ett ansåg att alla i projektet var involverade i skrivandet av användarfallen (P3, rad 48), och projektledaren ansåg att det endast var en senior utvecklare som var lösningsarkitekt som skrev dem (P5, rad 51). "Det blir den som är huvudansvarig för dokumenten, sen är det ju alltid kanske några inspel av andra utvecklare, men det är ju i första hand lösningsarkitekten, och i andra hand frontend- eller backendutvecklare" (P5, rad 51).

På frågan om användarfallen var användbara skilde svaren sig åt. Två av informanterna ansåg dem vara väldigt användbara (P2, rad 50; P4, rad 72), och två av dem tyckte att det varierade från projekt till projekt eller beroende på informationen (P1, rad 54; P3, rad 50). Projektledaren ansåg att de var användbara till viss del: "Hos oss på Avensia så förutsätter man ju att användarfallen bara ska gå att plocka och utveckla direkt, men så är det aldrig" (P5, rad 53).

På frågan om informanterna var bekväma med att skriva användarfall ansåg två av dem sig vara det (P2, rad 52; P3, rad 52), och två av dem var bekväma men tyckte inte det var vidare intressant (P1, rad 56; P4, rad 74). "Att skriva ett användarfall är inte svårt, det svåra är att bena ut vad det faktiskt var de ville ha" (P4, rad 74). Projektledaren ansåg sig inte vara bekväm med att skriva användarfall (P5, rad 55).

Kundinvolvering

För att undersöka hur kundinvolveringen såg ut på Avensia samt hur denna ansågs påverka det dagliga arbetet inom projekten ställde vi en fråga angående detta. Informanternas svar var lika varandras. Den generella uppfattningen var att det var viktigt att kunden var väldigt involverad för att produkten skulle bli så bra som möjligt, men att detta även kräver en viss kunskap från kunden. "I det här projektet så är de väldigt involverade. Men ibland har man varit i projekt där kunden inte förstår överhuvudtaget ett IT-projekt eller en webbsida" (P1, 58). Två av informanterna hävdade att kunderna till deras projekt var väldigt involverade och dessutom var på arbetsplatsen varje vecka för att följa upp (P2, rad 54; P3, rad 53). Informanten som var

projektledare hävdade också att kunden var väldigt involverade och poängterade även att detta krävdes (P5, rad 59). En av informanterna svarade kortfattat att kundinvolveringen var olika från projekt till projekt (P4, rad 76).

Vid frågan om kundinvolveringen borde öka eller minska var svaren lite mer skilda. ”Det beror helt och hållet på vilken kund man har. Det beror helt på deras kompetens och möjligheter och sådär” (P1, rad 60). Detta styrktes även av en av backendutvecklarna som svarade att det var väldigt projektspecifikt och att man inte kunde ha någon direkt standard för hur involverad en kund bör vara i ett projekt (P2, rad 56). En av informanterna ansåg att kundinvolveringen de hade i sitt projekt i dagsläget var på en lagom nivå, det vill säga en gång om dagen. Informanten menade att om det sedan skulle dyka upp något akut så var det bara att kontakta kunden (P3, rad 55). Informanten med rollen som lösningsarkitekt gav däremot ett bestämt svar. Informanten menade att generellt borde kundinvolveringen öka (P4, rad 80). Projektledaren svarade att det var viktigt att lyssna på kunderna och deras idéer (P5, rad 63). Däremot om kunden tar sig på för stort allvar kan det bli negativt med för mycket kundinvolvering (P5, rad 65).

Sammanfattning

Resultatet visade att informanterna inte var eniga angående ledningens fulla stöd, två informanter ansåg att ledningen inte stödde tillräckligt medan två var obestämde i frågan och endast en svarade att de ansåg den tillräcklig. Mötena ansågs relevanta, användarfallen ansågs överlag tydliga och det ansågs att rätt utvecklare placerats i rätt projekt. Däremot angående frågorna om det ansågs finnas rätt mängd dokumentation eller om stora projekt bröts ner i mindre delar var informanterna mindre eniga. Generellt ansågs det som att stora projektuppgifter bröts ner till mindre delar även om informanterna valde att beskriva det med andra ord. Dokumentationen rådde det skilda meningar om, hälften ansåg att den nuvarande var tillräcklig medan den andra hälften påstod motsatsen.

4.3.4 Tillägg

I slutet av intervjuerna frågade vi informanterna om de hade någonting de ville tillägga som vi inte hade berört med våra frågor. Fyra av informanterna hade inte någonting att tillägga till intervjun (P1, rad 68; P2, rad 64; P3, rad 64; P5, rad 71). Lösningsarkitekten berättade att kvalitetssäkring var någonting många tyckte ”man måste få tid till” (P4, rad 88), men att det istället resulterar i att det behövs mindre tid senare i projektet för att gå igenom tidigare skriven kod än om man hade gått igenom den direkt. Lösningsarkitekten förklarade att ”folk måste inse att man får ge lite tid för att det ska generera att det kommer ta mindre tid i slutändan... folk har inte den slutledningsförmågan utan tänker ’nu har vi väldigt bråttom’ och så kapar man av saker som man tycker är *over head*” (P4, rad 88).

4.4 Sammanfattning

Då varje fråga var kopplad till en central beståndsdel gjorde vi efter varje fråga en individuell tolkning utifall informanten ansåg att den centrala beståndsdelens uppfylldes i deras projekt. Nedan följer en sammanfattande tabell där vi enkelt staplar upp varje enskilt svar på varje fråga. Därefter följer två diagram där datan har staplats upp på ett enkelt och överskådligt vis.

4.4.1 Sammanfattande tabell

De centrala beståndsdelarna ur vårt teoretiska ramverk (kapitel 2.3) låg till grund för våra frågor i intervjuguiden (bilaga 7.1) som även motiveras i kapitel 3.3.1. Efter tidigare analys (kapitel 4.3) gick vi igenom punkt för punkt för att avgöra om informanten ansåg att den centrala beståndsdelens uppfylldes. Detta visas nedanför i tabell 8:

Tabell 8: Sammanfattande tabell över intervjusvar

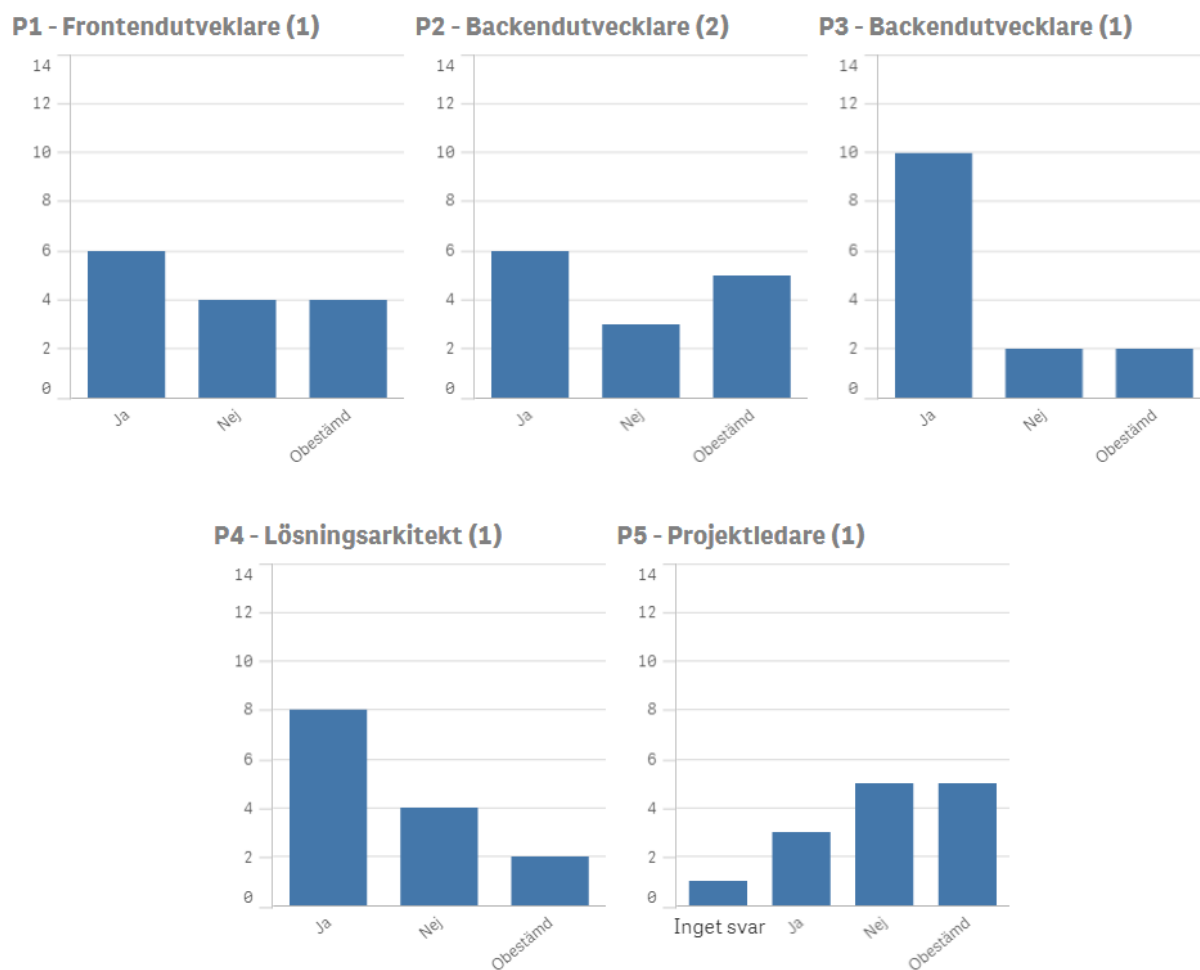
Central beståndsdel	P1 – frontutvecklare, projekt ett	P2 – backutvecklare, projekt två	P3 – backutvecklare, projekt ett	P4 – lösningsarkitekt, projekt ett	P5 – projektledare, projekt ett
God insikt i arbetslaget	Ja	Nej	Ja	Obestämd	Ja
Medvetet ökat självstyre	Ja	Ja	Ja	Ja	Ja
Samarbete mellan utvecklare och QA-personal	Nej	Nej	Nej	Nej	Nej
Öppen, samt strukturerad arbetsplats	Ja	Ja	Ja	Ja	Obestämd
Insikt i samtliga moduler	Ja	Ja	Ja	Ja	Inget svar
Samtlig kod testbar och standardisering av kod	Nej	Nej	Nej	Nej	Nej

Formell kodgenomgång	Nej	Ja	Ja	Obestämd	Nej
Fullt stöd från ledning	Obestämd	Obestämd	Ja	Nej	Nej
Möten med relevans	Obestämd	Ja	Ja	Ja	Obestämd
Rätt mängd dokumentation	Nej	Obestämd	Ja	Ja	Nej
Rätt utvecklare i rätt projekt	Ja	Obestämd	Ja	Ja	Ja
Bryt ned stora projekt till mindre delar	Ja	Obestämd	Obestämd	Ja	Obestämd
Tydliga användarfall	Obestämd	Ja	Obestämd	Ja	Obestämd
Kundinvolvring	Obestämd	Obestämd	Ja	Nej	Obestämd

4.4.2 Sammanfattande diagram

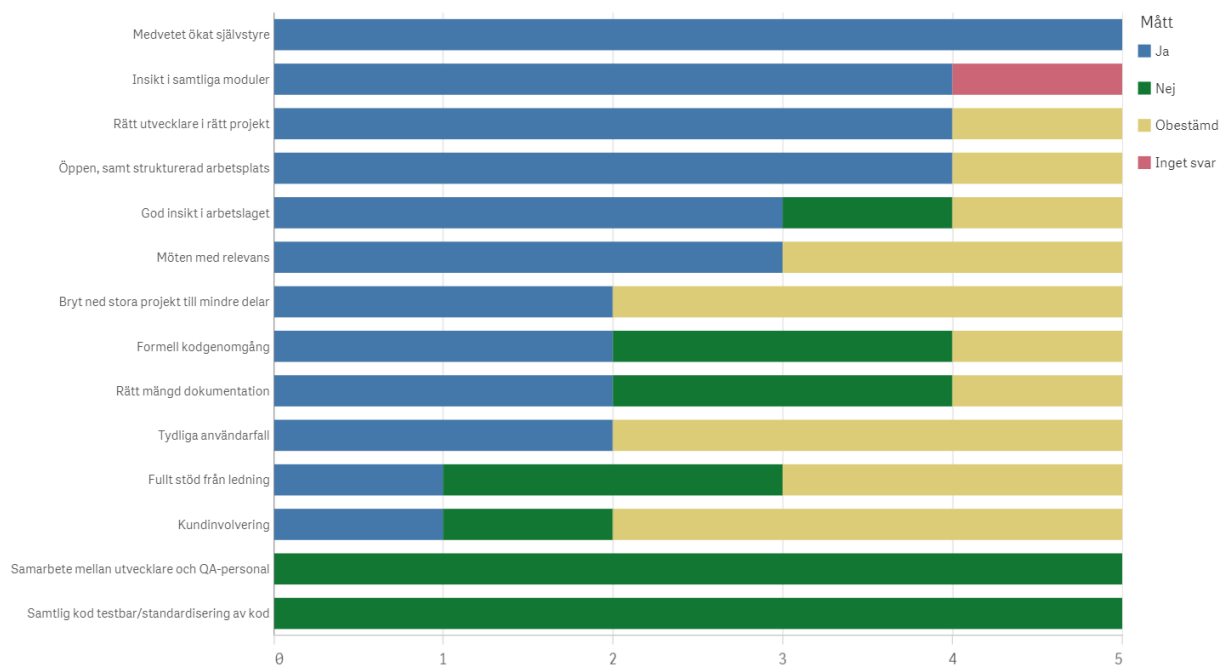
Utifrån den sammanfattade datan som presenteras i tabell 8 visualiserar vi två grafiska diagram med hjälp av verktyget *Qlik Sense*.

Figur 2: Statistik över informanternas svarsfrekvens



I figur 2 visualiseras frekvensen av ja, nej, obestämd och inget svar bland informanterna. Detta för att ge en tydligare bild av hur de enskilda informanternas generella uppfattning av hur de centrala beståndsdelarna uppfylldes i deras projekt. Informanternas projektnummer anges som siffra i parentes efter informantens arbetstitel.

Figur 3: Statistik över uppfyllandet av centrala beståndsdelar



I figur 3 visualiserar varje central beståndsdel på y-axeln med tillhörande frekvens av svar. Detta för att ge en överblick av vilken grad varje central beståndsdel anses uppfyllas av informanterna.

5 Analys & Diskussion

I detta kapitel analyserar och diskuterar vi vårt resultat som presenterades i det föregående kapitlet. Vi ställer även våra informanternas svar mot vårt teoretiska ramverk (kapitel 2.3), för att därmed uppnå en givande diskussion med teori kontra praktik. Strukturen på detta kapitel påminner om vårt tidigare kapitel med resultat (kapitel 4), där vi sorterar utifrån de centrala beståndsdelar som vår studie fokuserade på.

5.1 Centrala beståndsdelar

Nedan presenteras varje central beståndsdel under tillhörande kvalitetsgruppering, det vill säga arbetskvalitet, kodkvalitet och planeringskvalitet.

5.1.1 Arbetskvalitet

Nedan analyseras och diskuteras de centrala beståndsdelar som tillhör arbetskvalitet.

God insikt i arbetslaget

På frågan angående hur pass god insikt i projektets medlemmar som respektive projektledare hade var svaren varierande (kapitel 4.3.1). Detta anser vi kan bero på att det är en personlig fråga, varje individ har en egen personlig relation till projektledaren och svaren kan därför skilja sig. En annan variabel som bör beaktas är projektets längd samt hur länge varje projektmedlem varit involverad i projektet. En projektmedlem som inte varit med sedan projektstart kan exempelvis vara svårare för projektledaren att skapa en överskådlig bild av dennes kunskap. Eventuellt pendlar projektledarens uppfattade insikt under projektets livscykel, vilket kan bero på en rad olika orsaker, som exempelvis projektledarens arbetsbelastning under den specifika tiden. Projektledarens expertis kan även spela en roll i hur pass insatt denne verkar vara, en projektledare med en bakgrund som backendutvecklare kan te sig mer intresserad av projektmedlemmar som hanterar punkter av den karaktären. Projektledaren nämnde under intervjun att en formalisering och sammanställning av varje enskild projektmedlem i form av en kunskapsmatris skulle kunna underlätta arbetsfördelningen. En sådan matris hade eventuellt fungerat under vissa förhållande men hade enligt oss begränsat varje projektmedlems förmåga att utvecklas och prova sig på nya områden. Enligt oss låter en sådan matris även opersonlig då projektledaren fokuserar endast på kunskapen och inte på individen. Denna centrala beståndsdel (tabell 2:B) ska uppmuntra kommunikation och förståelse för varje individs kunskap och personlighet (Cao et al., 2009; Iivari & Iivari, 2011), och inte hålla den strikt formell.

Eventuella kvalitetssäkringsproblem härstammar inte från denna centrala beståndsdel i vår studie, då både projektledaren i projekt ett och två av projektmedlemmarna anser att det är god insikt i deras projekt, där både beröm och kritik ges till projektmedlemmarna. Den tredje medlemmen i projekt ett var obestämmd, men ansåg att det var väldigt svårt för projektledaren

att ha sådan insikt. Informanten från projekt två ansåg att projektledaren inte hade god insikt i hans kunskap eller personlighet. Han menar samtidigt att det inte är svart eller vitt och menar att det är omöjligt att projektledaren har väldigt bra eller väldigt dålig insikt i samtliga projektmedlemmars kunskap och personlighet. I hans fall kan det även bero på att projektledaren var ny.

Medvetet ökat självstyre

Svaren vi fick på frågan om informanterna tog egna initiativ eller inväntade delegering av arbetsuppgifter från ovanstående chef var ungefär samma från samtliga informanter (kapitel 4.3.1). Informanterna svarade att de tog egna initiativ, eller att de försökte ta egna initiativ då tillfälle gavs. Även denna fråga blev personlig då informanterna inte hade samma bakgrund eller arbetserfarenhet. Även om personerna blev informerade om att de skulle hållas anonyma anser vi att frågan ändå kunde få ett missvisande svar då den berörde hur individen arbetar. Utifrån egen erfarenhet drar vi slutsatsen att människor inte alltid vill berätta om vad de är mindre bra på. Att inte klara av att ta egna initiativ kan anses vara något negativt på grund av att personen inte är självgående, utan behöver få delegation från överordnad för att utföra sitt arbete.

Frontendutvecklaren förklarade att det inte alltid fanns utrymme för att ta egna initiativ utan man fick göra de uppgifter som man blev tilldelad att göra genom företagets interna system (kapitel 4.3.1). De övriga informanterna uttryckte sig på ett liknande vis; att initiativ togs när möjligheten gavs. Detta tyder på att viljan att ta egna initiativ fanns, men att det inte alltid fanns möjlighet att göra det. Denna centrala beståndsdel (tabell 2:C) menar att utvecklare ska ta ansvar för sitt ökade självstyre för att projektets produkt ska uppnå kvalitet (Cao et al., 2009; Iivari & Iivari, 2011). Eftersom samtliga informanter tar ansvar för sitt ökade självstyre, alltså tar egna initiativ då det finns möjlighet, finner vi inte att det är i denna centrala beståndsdel problem med kvalitetssäkring uppstår.

Samarbete mellan utvecklare & QA-personal

På frågan angående hur samarbetet fungerade mellan utvecklare och QA-personal var svaren eniga. Någon renodlad QA-personal fanns det inte i något av projekten (kapitel 4.3.1). Några av informanterna svarade dock att utvecklarna var sin egen kvalitetsäkringspersonal då de i mån av tid testade och gick igenom den kod som skrivits. Lösningssarkitekten berättade att de lade över delar av kvalitetssäkringen av produkten till kunden, något vi tolkar som en oklar form av QA-personal. Backendutvecklaren i projekt två menade också att dem själv var deras egen QA-personal men till skillnad från informanterna i projekt ett hade de avsatt tid endast för att kvalitetssäkra. Detta anser vi vara ett litet steg på vägen mot att ha specifik QA-personal. Genom att utvecklarna i båda projekten anser sig vara sin egen QA-personal anser vi att samarbetet mellan QA-personal och utvecklare är god, då de både känner sig själv och arbetar tillsammans med de övriga i projektet dagligen. Denna centrala beståndsdel innefattar inte bara att det ska vara bra samarbete mellan QA-personal och utvecklare utan även att utvecklare bör kunna göra enhetstester på andra utvecklares kod (tabell 3:2; Cho et al., 2011; Iivari & Iivari, 2011; Dingsøy et al., 2012; Hoda et al., 2011).

Det visade sig att det fanns en tanke att utvecklarna skulle testa varandras kod men att det inte fanns tillräckligt med tid till det. Lösningssarkitekten tog även upp att det fanns en ekonomisk faktor bakom varför det inte fanns någon QA-personal. Faktorn var att eventuell QA-personal skulle behöva läras upp i hur systemen fungerade, samt hur det var tänkt att varje del skulle fungera, och därför menade han på att det var bättre att arbetslaget utförde testning (kapitel 4.3.1). Vi anser dock att utbilda QA-personal är bättre i längden då utvecklarna kan lägga full fokus på att skriva bra kod och därmed överlåta kvalitetssäkringen åt någon annan.

Trots att ”samarbetet” mellan QA-personal (de är sin egen) och utvecklare är bra saknas det helt och hållet renodlad QA-personal, vilket gör att denna centrala beståndsdel kan vara en bidragande faktor till att god kvalitet inte hålls.

Öppen, samt strukturerad arbetsplats

Eftersom samtliga informanter sitter på samma kontor var svaren liknande från samtliga; de har ett öppet kontorslandskap. En aspekt vi som grupp hade diskuterat redan innan intervjuerna var att det måste vara hög ljudnivå när man har öppet kontorslandskap, och detta verkade stämma då samtliga informanter påpekade att ljudnivån var hög (kapitel 4.3.1). Även om den centrala beståndsdelen trycker på att det öppna kontorslandskapet ska göra det enklare att kommunicera och arbeta i lag (tabell 3:12; Hoda et al., 2011) anser vi att den höga ljudnivån kan upplevas som en stor nackdel. Trots ljudnivån verkade samtliga informanter förutom projektledaren inte störas av ljudnivån då de enkelt kunde skärma av med musik i hörlurar. Som projektledaren säger tror vi också det är viktigt att det finns utrymmen dit man kan gå för att få sitta i fred när man behöver det, det är ju trots allt inte alla människor som klarar av att arbeta under konstant hög ljudnivå.

Utveckling i allmänhet anser vi handla om problemlösning vilket gör att man ibland behöver kunna lägga full fokus på en specifik uppgift och inte ha distraktionsmoment runt omkring sig. På projektledaren lät det inte som att arbetsplatsens struktur var optimal då vissa projekt inte kunde sitta samlade och de tvingades delas upp bland andra projektgrupper. Eftersom kontoret där informanterna sitter har öppet kontorslandskap, och att endast en av fem informanter hade någonting att invända vad gäller strukturen, anser vi att det inte är på grund av kontorslandskapet som eventuella problem med kvalitetssäkring uppstår. Utifrån egna erfarenheter tycker även vi att ett öppet kontorslandskap förenklar kommunikationen mellan människor, och det blir enklare att fråga om hjälp eller diskutera problem. Istället för att behöva knacka på en kontorsdörr och känna att man stör tar man tag i personen man vill prata med på tu man hand.

Insikt i samtliga moduler

Det visade sig att samtliga informanter hade andra delar av systemet i åtanke när de utvecklade moduler. Den centrala beståndsdelen säger att utvecklarna ska ha samtliga kopplingar mellan moduler i projektet i åtanke under utvecklingens gång (tabell 3:13; Cho et al., 2011; Iivari & Iivari, 2011; Dingsøy et al., 2012; Hoda et al., 2011), och eftersom samtliga tillfrågade informanter svarade att de hade det är det inte heller här eventuella kvalitetssäkringsproblem uppstår. Backendutvecklaren i projekt två tog under intervjun fram en bild han hade

skissat upp över hur olika moduler såg ut, vad de hade för inparametrar samt vad de returnerade. Detta visar hur de verkligen arbetar med att få hela projektet att gå ihop trots att det är olika personer som utvecklat varje del.

Tidigare arbetserfarenhet anser vi spela en större roll gällande uppfyllandet av denna centrala beståndsdel, då en mindre erfaren person förmodligen inte har samma helhetsbild som en mer erfaren person har. Det är nog först när personen arbetat ett tag med problemlösning och utveckling som denne får en helhetsbild över hur allting hänger samman. Backendutvecklaren i projekt ett var ny i branschen men försökte ändå ha de andra modulerna i åtanke när hon utvecklade, men hon poängterade just vad vi diskuterat också, att det inte var så lätt då hon var ny i branschen. Sammanfattningsvis anser vi att den centrala beståndsdelens inte är en bidragande faktor till eventuella kvalitetssäkringsbrister i vår studie.

5.1.2 Kodkvalitet

Nedan analyseras och diskuteras de centrala beståndsdelar som tillhör kodkvalitet.

Samtlig kod testbar och Standardisering av kod

En formell kod-standardisering är enligt teorin något som ökar läsbarheten samt förståelsen av andras kod (tabell 3:4; Cho et al., 2011; Iivari & Iivari, 2011; Dingsøy et al., 2012; Hoda et al., 2011), och gör även det enklare för nya medlemmar i arbetslaget att hoppa in i projektet. Samtliga informanter svarade att de inte följde någon standardisering av kod i sina projekt, men ett antal av dem menade att det var någonting de hade diskuterat och även försökt få igång, dock utan resultat (kapitel 4.3.2). Frontendutvecklaren menade att problemet kunde ligga i att det ständigt kom nya standardiseringar, vilket gjorde det svårt att välja en specifik. Den förklaringen ser vi som trolig gällande just frontendutveckling, då det enligt oss hanterar ett mer levande programmeringsspråk som förändras efter vad som anses vara populärt för tillfället.

Lösningssarkitekten menade att även ifall projekten bestämde sig för att använda en specifik standard så gick utvecklarna snabbt tillbaka till ”sin egen” standard för att de inte förstod vad det innebar att använda sig av den tilltänkta standardiseringen. Agil utveckling förespråkar individens egenmakt och ansvar (Lindstrom & Jeffries, 2004; Mangalaraj et al., 2009), vilket förklarar varför lösningssarkitekten svarade som han gjorde. Även ifall projektledaren svarade med att det inte följdes någon standardisering, och att det inte var bra, så värderas individen och dess arbetssätt högre i ett agilt utvecklingsprojekt.

Vi anser ovan resultat ha en stor inverkan på den allmänna kvalitetssäkringen i vår studie, eftersom utvecklare i projekten följer sin egen standardisering. Moduler som ska påminna om varandra till sättet kan i slutändan se ut, och fungera, på helt olika vis eftersom det är olika utvecklare som står bakom koden. Det gör det även enligt oss betydligt svårare att skriva tester till funktioner, där en funktion beter sig på ett sätt och en annan på ett helt annat sätt. Enligt oss blir det även problematiskt att introducera nya medlemmar till projektet, eftersom en rad olika standardiseringar existerar att förhålla sig till. Teorin menar att det är viktigt att inte ha

alltför strikt standardisering eftersom det kan påverka individens initiativ och egentänkande, men att någon form av standardisering bör existera (tabell 3:4; Cho et al., 2011; Iivari & Iivari, 2011; Dingsøy et al., 2012; Hoda et al., 2011). Därmed anser vi att avsaknaden av den centrala beståndsdelen påverkar kvalitetssäkringen negativt i vår studie.

Formell kodgenomgång

Teorin ser användandet av formell kodgenomgång som vital och kritisk för att hålla hög kvalitetssäkring, samt att organisationer bör etablera ett effektivt tillvägagångssätt för att utföra kodgenomgång (tabell 3:7; Cho et al., 2011). Svaren från våra informanter var lite mer skilda på denna fråga (kapitel 4.3.2), där båda backendutvecklare menade att någon form av formell kodgenomgång existerade, frontendutvecklaren och projektledaren ansåg att det inte fanns någon kodgenomgång, och lösningsarkitekten ansåg att det varierade från projekt till projekt. Backendutvecklaren från projekt ett, som ansåg att det existerade en formell kodgenomgång, hade troligtvis den åsikten eftersom hon är ny i branschen och har desto större anledning att be sina kollegor att se över hennes kod. Lösningsarkitekten var av åsikten att kodgenomgångar borde utföras mer informellt och på mindre delar för att det ska gå snabbt och effektivt, något som backendutvecklaren från projekt två går emot och menar att kodgenomgångar behövs göras på krångliga delar.

Projektledaren säger att det har pratats om det, och hon vill formalisera processen desto mer genom att ha så kallade *code reviews*, där samtlig kod ska kvalitetssäkras av en annan utvecklare innan den anses vara färdig. Att projektledaren vill införa kodgenomgång som dessutom är desto mer formaliserad kan förklaras med att hon redan nu anser att disciplinen bland utvecklarna är för låg, och att deras interna verktyg JIRA inte används i den mån som krävs enligt henne.

Vi anser att avsaknaden av formella, men även informella, kodgenomgångar är en anledning till avsaknad av kvalitetssäkring i vår studie. Endast en person från projekt ett ansåg att det existerade kodgenomgång i deras projekt, vilket troligtvis kan förklaras med att hon är ny i branschen och är därför i större behov av kontroll av hennes kod. Kunskapsspridningen som resulteras från kodgenomgång hade gynnat samtliga involverade i projekten, och teorin anser att kodgenomgång är en vital och kritisk del för att skapa mjukvara av hög kvalitet (tabell 3:7; Cho et al., 2011; Iivari & Iivari, 2011; Dingsøy et al., 2012, Hoda et al., 2011).

5.1.3 Planeringskvalitet

Nedan analyseras och diskuteras de centrala beståndsdelar som tillhör planeringskvalitet.

Fullt stöd från ledning

För att uppnå hög kvalitet anser teorin att det inte räcker med stöd från projektledaren till projektmedlemmarna utan att det även krävs stöd från den övre ledningen bestående av bland annat platschefer och VD (tabell 2:A; Cao et al., 2009; Iivari & Iivari, 2011). Svaren vi samlade in från informanterna var relativt lika men skilde sig på vissa punkter. Samtliga informanter

från båda projekten var överens om att de fick tillräckligt stöd från respektive projektledare (kapitel 4.3.3), men när det gällde den övre ledningen var svaren mer varierade.

Endast backendutvecklaren från projekt ett tyckte sig se ett tydligt stöd från den övre ledningen, och övriga informanter var mindre positiva gällande den punkten. Återigen kan detta bero på att just den backendutvecklaren var ny i branschen, och hade en mer positiv inställning till engagemanget från de övre cheferna. Lösningssarkitekten hade åsikten att det inte var möjligt för de övre cheferna att vara involverade i detaljnivå på grund av arbetsbördan och tyckte att de endast visade något intresse när det uppstod problem, vilket man får ta hänsyn till.

Projektledaren, som förmodligen är den i projektgruppen som har mest direktkontakt med de övre cheferna och vars åsikt väger tyngst, tyckte att intressenivån från de övre cheferna inte var särskilt stor och att det främst handlade om de ekonomiska aspekterna. Vi anser att det är av stor betydelse för kvalitetssäkringen av projekten i vår studie att projektledaren anser att ledningens intresse är svalt gällande deras projekt, och även att hon önskar att intresset bland dem vore större. Det är även sådant intresse som kan visa sig gentemot kunden när den exempelvis är på besök, som kan ge ett negativt eller fel intryck till kunden, något som även projektledaren påpekar i sitt svar till oss.

Möten med relevans

Teorin är bestämd med att de möten som sker bör hållas relevanta och i lagom längd (tabell 3:1; Cho et al., 2011). Våra informanter i vår studie var generellt positiva på denna fråga, där tre av de tillfrågade ansåg att deras möten höll hög relevans och bidrog till kunskapsspridning (kapitel 4.3.3). Endast frontendutvecklaren och projektledaren var obestämda på frågan, där frontendutvecklaren menade på att hon inte var väldigt intresserad av möten där det fokuserades för mycket på backend. Projektledaren menade på att deras möten borde vara ännu mer välplanerade och ha ett ännu tydligare syfte.

Vi kan förstå frontendutvecklarens åsikt i att känna sig utanför när det sker samtal i områden som inte berör henne direkt, men hon påpekar även att det är en form av kunskapsspridning när det uppstår i deras möten. Backendutvecklaren från projekt två menade även att i hans projekt så delade de upp sig inför sina möten för att hålla det skilt mellan backend och frontend. I detta fall anser vi att det är bättre att blanda folk från båda läger för att öka kunskapsspridningen som frontendutvecklaren också ansåg, även ifall det ibland innebär en del irrelevans för individen. Vi förstår också projektledarens vilja av att ha ännu mer välplanerade möten, men att generellt så är mötena i vår studie av en acceptabel kvalitetsgrad och har ingen större negativ påverkan på den allmänna kvaliteten. Sammanfattningsvis anser vi att den centrala beståndsdelen uppfylls och är inte en bidragande faktor till eventuella brister av kvalitetssäkring i vår studie.

Rätt mängd dokumentation

För att säkerställa att ett projekt är av hög kvalitet är det viktigt att det under projektets gång har utförts rätt mängd dokumentation (tabell 3:5; Cho et al., 2011; Iivari & Iivari, 2011;

Dingsøy et al., 2012, Hoda et al., 2011). Dokumentationen är väsentlig för att kunden ska kunna tolka produkten på korrekt vis. Om en kund inte förstår hur den ska använda produkten och därmed går miste om eventuell funktionalitet så anser vi inte detta projekt vara av hög kvalitet. Dokumentation är även viktig om det är mer än en utvecklare som arbetar med samma kod, eller om denna kod senare ska överlätas till en annan utvecklare. Med korrekt dokumentation blir risken för missförstånd mindre och därmed chansen för en lyckad övergång större. Ett projekt som har för mycket dokumentation anses dock även av oss som en kvalitetsbrist då värdefull projekttid dedikerats åt onödig dokumentation som sedan inte används. Det är av stor vikt att dokumentationen som utförs är av kvalitet, om den inte kan tolkas eller kommer att tolkas är den helt onödig och även då en brist på kvalitet (Cho et al., 2011).

Samtliga informanter var eniga om att mängden dokumentation som utfördes inom projekten de arbetade i var låg, däremot var de inte överens om den låga mängden var tillräcklig eller inte. Informanterna var även överens om att det inte fanns någon gemensam överenskommelse angående hur dokumenteringen skulle utföras. Detta innebar att alla utvecklare dokumenterar på olika sätt vilket enligt oss enkelt kan leda till missförstånd. En informant nämnde även att bristen på dokumentation kan bero på kunden. Om en kund involverar sig i resursfördelningen inom projektet kan den eventuellt inte förstå vikten av dokumentation och således skala ner den för att spara pengar och tid. Två av informanterna talade om tidigare arbetsplatser där det dokumenterades betydligt mer, men vi uppfattade det inte som att de ansåg att det var bättre. Informanten som var projektledare ansåg också att det behövdes mer dokumentation i annan form än de hade i dagsläget. Informanterna verkar enligt oss inte anse att dokumentation är en särskilt viktig del av projekten.

Utifrån resultatet i vår studie anser vi att avsaknaden av en standard eller överenskommelse angående hur dokumentationen borde utföras med stor sannolikhet sänker den potentiella kvaliteten på slutprodukten. Utan en standard för hur dokumentationen bör utföras blir det svårt, om inte omöjligt, att bedöma mängden dokumentation som utförts. Förslagsvis bör detta gås igenom vid projektstart för att undersöka vilken sorts dokumentation som kommer vara av relevans tillsammans med kunden. Projektledaren nämner att dokumentationen de innehar för tillfället är väldigt teknisk och svår att förstå. Detta tror vi är svårt för kunden att ta nytta av vilket därmed renderar dokumentationen mindre användbar än om den skrivits på ett anorlunda vis. Vi anser därmed inte att den centrala beståndsdelen uppfylls vilket leder till kvalitetssäkringsbrister enligt vår studie.

Rätt utvecklare i rätt projekt

För att uppnå hög kvalitet inom agila projekt är det viktigt att rätt utvecklare sätts i rätt projekt (tabell 3:6; Cho et al., 2011; Iivari & Iivari, 2011; Dingsøy et al., 2012, Hoda et al., 2011). Varje utvecklare är unik och har unika egenskaper och preferenser, vilket även innebär att vissa utvecklare är bättre lämpade för vissa miljöer än andra. Detta kan handla om personligt intresse i projektet från utvecklarens sida, men även om tolkningen samt fördelningen av uppgifter till varje utvecklare från projektledarens sida. Vi undersökte den centrala beståndsdelen

genom att fråga informanterna om de ansåg att de hade ett personligt intresse i det projekt de arbetar i samt om de hellre hade arbetat i något annat projekt som fanns på deras arbetsplats.

Svaren från informanterna var även här lika då alla utom en svarade att de hade ett starkt intresse för deras projekt. Vi ansåg att informanterna verkade svara ärligt och verkligen fann ett intresse för sitt arbete. Även om de, som en informant påpekade, rent projektspecifikt kanske inte var särskilt intresserade så var de intresserade av tekniken och sina arbetsuppgifter relaterade till projektet. Vi anser att detta troligen beror på arbetsfördelningen samt landskapet inom företaget. Om utvecklarna blir tilldelade eller känner sig välkomna att i största möjliga grad kunna fokusera på områden de känner sig intresserade av kommer kvaliteten på slutprodukten vara hög.

Vi anser enligt vår studie att detta centrala begrepp uppfylls på Avensia. Inga informanter antydde enligt oss på att de blivit malplacerade utan de verkade anse sig själva vara inte bara placerade i rätt projekt utan även i rätt plats inom projektet.

Bryt ned stora projekt till mindre delar

En väsentlig del för att kunna kvalitetssäkra ett projekt är att kunna hantera större projektuppgifter, och det finns olika strategier för att hantera uppgifter av detta slag. Det viktigaste vid hanteringen av en stor projektuppgift är enligt teorin att bryta ned dessa i mindre hanterbara delar (tabell 3:8; Cho et al., 2011; Iivari & Iivari, 2011; Dingsøy et al., 2012; Hoda et al., 2011). Det är således av ytterst vikt att varje projektmedlem är familjär med denna strategi och är underförstådd i hur detta bör utföras. Vi ville inte fråga direkt om Avensia arbetade efter denna modell utan vi frågade istället "hur angriper ni större projektuppgifter?". Tanken var att undersöka huruvida detta tillvägagångssätt var etablerat och användes. Två av informanterna svarade direkt att de bröt ner större projektuppgifter i mindre delar, de andra informanterna svarade att fokus borde ligga på planering och prioritering. Risken fanns att samtliga informanter hade svarat att de hade hanterat det genom att bryta ner uppgiften i mindre delar om vi ställt en ledande fråga, då det enligt oss kan låta som en självklarhet.

Vår fråga kan även vara svår att besvara på plats då vissa saker man gör på daglig basis blir rutin och man är inte van vid att förklara eller formulera det i tal. En av informanterna som inte nämnde att bryta ner projektuppgiften i mindre delar nämnde till exempel att man skulle syna all kod för att undersöka hur olika saker påverkar varandra, vilket vi anser indirekt innebära en uppdelning av uppgiften. En annan informant besvarade att det borde hanteras genom parprogrammering, vilket även kräver en uppdelning av uppgiften. Projektledaren nämnde sin *change management*-process där man kontaktar leveransansvarig för att få fram resurser. Vi anser även detta angripande vara en del av att bryta ned större projektuppgifter till mindre delar. Sammanfattningsvis anser vi att Avensia hanterar stora projektuppgifter i enlighet med vad vår studie kommit fram till är det optimala tillvägagångssättet, dock är de eventuellt inte medvetna om detta. Kanske vore det bra för övre ledning och projektledare att sätta ord på deras taktik och standardisera detta tillvägagångssätt då det inte verkade vara en självklarhet.

Vi anser enligt vår studie att detta centrala begrepp inte är ett kvalitetssäkringsproblem i dagsläget, men att det, om det kom in en ny utvecklare med ett annat tillvägagångssätt, eventuellt skulle kunna bli ett problem då det inte fanns någon standard eller underförstått tillvägagångssätt.

Tydliga användarfall

För att säkerställa kvaliteten inom ett projekt är det viktigt att ha klara och tydliga användarfall (tabell 3:9; Cho et al., 2011; Iivari & Iivari, 2011; Dingsøy et al., 2012; Hoda et al., 2011). Det är viktigt att det är bestämt vem som formulerar användarfallen, hur de bör formuleras samt att de anses användbara av de som ska använda dem. Vi fick skilda svar angående vem som skrev användarfallen. Tre av informanterna angav att de skrevs av lösningsarkitekten och projektledaren under förstudien vilket vi ansåg intressant då detta inte stämde överens med vad projektledaren svarade. Projektledaren ansåg att det endast var lösningsarkitekten som skrev användarfallen under förstudien. Vid frågan om de kände sig bekväma med att skriva användarfall svarade alla utom projektledaren att de var bekväma med att skriva dessa även om två av informanterna poängterade att de inte var intresserade av att göra det. Vi ansåg att den gemensamma uppfattningen om användarfall från alla informanter utom projektledaren var att det var en banal uppgift, vilket vi ansåg intressant och något motsägelsefullt då samtliga informanter samtidigt verkade anse att användarfall var en viktig del av deras projekt.

Den uppfattade användbarheten av användarfallen skilde sig åt mellan informanterna också; två av informanterna svarade att användarfallen var bra och tydliga, två svarade att det varierade beroende på projektet och informationen. Projektledaren ansåg att de var användbara till viss del, men att det rådde vissa tvivel kring hur de borde tolkas. Detta tyder enligt oss på att de är något otydliga. Vi anser att det troligen behövs fastställas vem som skriver användarfall samt hur de bör tolkas och användas. Om en projektmedlem inte anser användarfallet tillräckligt tydligt utan känner att denne måste styrka vissa aspekter av det borde det rimligtvis ha skrivits tydligare från början.

Sammanfattningsvis anser vi enligt vår studie att det centrala begreppet inte fullständigt uppfylls och att det således eventuellt finns vissa kvalitetssäkringsbrister i utförandet samt användandet av användarfall på företaget.

Kundinvolvering

Teorin anser att kundinvolvering i agila projekt bör vara hög, och att kunden även bör vara med och bestämma (tabell 3:10; Cho et al., 2011; Iivari & Iivari, 2011; Dingsøy et al., 2012; Hoda et al., 2011). Vår uppfattning av informanternas svar var att kundinvolveringen i deras nuvarande projekt var hög, men att kundens kunskap och grad av insikt varierade. På frågan om informanterna ansåg att kundens involvering borde öka eller minska ansåg de att det varierade från kund till kund. Överlag anser vi att informanterna var nöjda med graden av kundinvolvering.

Därmed anser vi att den centrala beståndsdelen inte har någon negativ påverkan på kvalitetssäkringen i vår studie.

5.2 Urvalsproblematik

Då studien är begränsad till ett företag kan det diskuteras ifall resultatet hade haft större variation om fler företag hade ingått i studien. Vår studie inkluderar två fall med separata projektmedlemmar, strukturer och förhållningssätt och vi anser att detta gav oss tillräcklig data för vår undersökning. Vi anser att studien inte hade gynnats av ytterligare företag då vi utför en kvalitativ studie och då troligen hade nått en grad av svarsräktnad, det vill säga otillräcklig variation bland svaren för att bidra till ett annorlunda resultat.

5.3 Sammanfattning av analys och diskussion

Efter genomgående analys och diskussion av det empiriska resultatet kom vi fram till att en del centrala beståndsdelar uppfylldes medan en del uppfylldes delvis eller inte alls. När det rörde meningsskiljaktigheter mellan informanterna angående en viss central beståndsdel fick vi diskutera och bedöma vilka åsikter som vägde tyngst för att fatta ett beslut kring uppfyllandet. Vi sammanfattar detta i tabell 9:

Tabell 9: Sammanfattande tabell över centrala beståndsdelsuppfyllnader

Central beståndsdel	Uppfylls tillräckligt i vår studie
God insikt i arbetslaget	Ja
Medvetet ökat självstyre	Ja
Samarbete mellan utvecklare och QA-personal	Nej
Öppen, samt strukturerad, arbetsplats	Ja
Insikt i samtliga moduler	Ja
Samtlig kod testbar och standardisering av kod	Nej
Formell kodgenomgång	Nej
Fullt stöd från ledning	Nej
Möten med relevans	Ja

Rätt mängd dokumentation	Nej
Rätt utvecklare i rätt projekt	Ja
Bryt ned stora projekt till mindre delar	Ja
Tydliga användarfall	Nej
Kundinvolvering	Ja

Samarbetet mellan utvecklare och QA-personal uppfylldes inte på grund av att det inte fanns personal som var fullt dedikerade till att kvalitetssäkra, trots att det fanns ett uttalat behov av sådan personal. Vi anser att detta är ett projektspecifikt problem då avsaknad av QA-personal inte är någonting som är unikt för agila projekt samt att om QA-personal hade funnits kan vi inte se att problem direkt relaterade till agila arbetsmetoder skulle uppstå.

Problemet med att inte följa en gemensam kodstandard är däremot någonting vi anser vara direkt relaterat till agil utveckling, detta eftersom det helt saknas formella arbetsriktlinjer. Det är således viktigt att ha detta i åtanke eftersom det försvårar arbetet med att skriva tester samt introducera nya personer in i projektet. En av de uttalade nackdelarna med agil utveckling är att det inte förespråkar formell kommunikation (kapitel 2.1.3; Barlow et al., 2011; Cho et al., 2011) vilket enligt vår studie leder till att samtliga utvecklare har sin egen arbetsmetodik (kapitel 5.1.2). Även nästkommande centrala beståndsdel, formell kodgenomgång, uppfylls inte på grund av att det inte förespråkas någon formell kommunikation, vilket leder till sämre kvalitetssäkring (kapitel 5.1.2).

Avsaknad av fullt stöd från ledning anser vi inte vara specifikt för agila utvecklingsprojekt, men det kan vara lättare att få en bättre överblick i icke-agila projekt för ledningen att få insikt och intresse. En av de negativa aspekterna med agil utveckling (kapitel 2.1.3; Barlow et al., 2011; Cho et al., 2011) är att det initialt är svårt att estimerar tid och resurser, något vi i vår studie har funnit är av stort intresse hos den övre ledningen (kapitel 5.1.3). En annan nackdel med agil utveckling är att agila utvecklingsprojekt tenderar att ha bristfällig dokumentation (kapitel 2.1.3; Barlow et al., 2011; Cho et al., 2011) något även vår studie bekräftar (kapitel 5.1.3). En annan nackdel med agila arbetsmetoder är att krav inte är väldefinierade (kapitel 2.1.3; Barlow et al., 2011; Cho et al., 2011) vilket vår studie bekräftar då den centrala beståndsdel med tydliga användarfall inte uppfylls tillräckligt (kapitel 5.1.3).

6 Slutsats

I det slutgiltiga kapitlet väver vi samman vår studie för att besvara vår ursprungliga forskningsfråga. Vi tar upp de centrala beståndsdelar som vi ansåg bidrog till kvalitetssäkringsproblemen på Avenia och kopplar sedan dessa till den agila arbetsmetodiken för att dra slutsatser om problemen är företagsspecifika eller metodikspecifika. Därefter ger vi förslag på hur vidare forskning i ämnet kan utföras med vår studie som grund.

6.1 Hur påverkas kvalitetssäkring av agila arbetsmetoder?

Enligt vår studie uppstod det kvalitetssäkringsbrister i samtliga fall som arbetade utefter agila arbetsmetoder. Detta kom vi fram till efter att ha analyserat och diskuterat resultatet och konstaterade att en rad centrala beståndsdelar inte uppfylldes tillräckligt för att kvalitetssäkra slutresultatet.

Sammanfattningsvis är det de centrala beståndsdelar som nämns i kapitel 5.3 som agila utvecklingsprojekt bör ha i särskild åtanke för att säkerställa det slutgiltiga resultatets kvalitet, då de enligt vår studie är direkt relaterade till agila arbetsmetoder. Vi summerar dem i listan nedan:

- Samtlig kod testbar
- Standardisering av kod
- Formell kodgenomgång
- Fullt stöd från ledning
- Rätt mängd dokumentation
- Tydliga användarfall

Vårt resultat kan även appliceras i agila arbetsmiljöer för att undersöka hur resultatet påverkas när de ovan nämnda centrala beståndsdelarna får extra fokus. Vi anser även att flygplansmodellen (figur 1) kan användas som ett ramverk att förhålla sig till för att kvalitetssäkra produkten utvecklad i utvecklingsprojekt som arbetar utefter *Scrum* och/eller *XP*.

6.2 Förslag till vidare forskning

Med hjälp av ytterligare empiriska studier, förslagsvis genom en större kvantitativ studie som jämför agila arbetsprojekt med traditionella arbetsprojekt (exempelvis som följer vattenfallsmodellen), skulle vårt ramverk kunna styrkas av både praktik och teori.

Vidare forskning kan vara en uppföljning av ett projekt som applicerat vårt ramverk på ett pågående projekt för att undersöka konsekvenser samt effekter av ramverket.

7 Bilagor

7.1 Intervjuguide

7.1.1 Kort introduktion av författarna

Vi är tre studenter som läser det Systemvetenskapliga Kandidatprogrammet. Nu, under vår sista termin, skriver vi vår kandidatuppsats som fokuserar på kvalitetssäkring inom agila utvecklingsprojekt. Vår uppfattning av agila arbetsprojekt är att de förespråkar egenansvar och flexibilitet, men kan samtidigt upplevas som ostrukturerade samt svåra att tids- och kostnadsestimeras. Vi har därför intresserat oss i att undersöka kvalitetssäkring utifrån dessa förhållanden.

Intervjun kommer att spelas in för att undvika missförstånd under transkriberingen. Transkriptet kommer i god tid innan publikation skickas till dig för godkännande.

7.1.2 Kort introduktion av informant

1. Kan du berätta lite om dig själv? (Ålder, arbetsroll, utbildning)
2. Vad är din erfarenhet inom agila utvecklingsprojekt? (Nuvarande samt tidigare)

7.1.3 Intervjufrågor

1. Hur pass engagerad och insatt anser du att dina överordnade är till projektet du arbetar i för tillfället?
 - a. Hur visar dem sitt engagemang eller avsaknad därav?
2. Har du personligt intresse i de projekt du arbetar i?
 - a. Finns det andra projekt som intresserar dig mer som du hellre arbetat i?
3. *Beroende på arbetsroll:*
 - a. *Projektledare:* Hur pass god insikt i varje projektmedlems kunskapsgrad/personlighet anser du att du har?
 - i. Hur skaffar du denna insikt samt upprätthåller den?
 - b. *Projektmedlem:* Hur pass god insikt i din kunskap/personlighet anser du att din projektledare innehar?
 - i. Hur återspeglas detta till dig?
4. Finner du att du tar egna initiativ eller inväntar du delegation från överordnad?
 - a. Vad föredrar du?
5. Hur ofta har ni projektmöten?
 - a. Hur tycker du att dessa är, sett till relevans och planering?
6. Hur fungerar samarbetet mellan utvecklare och kvalitetssäkringspersonal?
 - a. Hur hade samarbetet kunnat förbättras?
 - b. Anser du att det finns ett behov av kvalitetssäkringspersonal?
7. Vilka eventuella riktlinjer följer ni för att skriva testbar kod, har ni någon kodstandard?
 - a. Hur följs dessa?
 - b. Sker det någon formell kodgenomgång?
 - i. Hur går denna till?
 - c. *Arbetsrollen utvecklare:* Har du övriga kodmoduler i åtanke vid din egen utveckling?
8. Hur ser dokumentationen ut i era projekt?

- a. Anser du att den är tillräcklig?
9. Hur angriper ni större projektuppgifter?
 - a. Finns det andra tillvägagångssätt som du hade föredragit?
 - i. Varför?
10. Vem formulerar era användarfall?
 - a. Är användarfallen användbara?
 - b. Känner du dig bekväm med att skriva användarfall?
11. Hur involverad är kunden i era projekt?
 - a. Anser du att kundinvolveringen borde öka eller minska?
 - i. Varför?
12. Hur fungerar ert kontorslandskap?
 - a. Hur påverkar det ditt dagliga arbete?
13. Tack för din tid! Har du något du vill tillägga?

7.2 Intervjuprotokoll

Samtliga protokoll presenteras i tabeller med fyra kolumner: Nr (radnumrering), Tema (relevant central beståndsdel i ramverket), Person (talande person), Fråga/svar (transkriberad text). Talande person är antingen en av författarnas initialer eller informantens ordningsnummer.

Under intervjuerna nämns begreppen JIRA och TFS. JIRA är ett internt administrationsverktyg som på Avensia används för att registrera användarfall, buggar, med mera. TFS står för Team Foundation Server, och liknar JIRA till arbets sättet men är i detta fall sammankopplat med kundens system.

7.2.1 Intervjuprotokoll 1

- Arbetsroll: Frontendutvecklare
- Ålder: 35
- Kön: Kvinna
- Datum för intervju: 2015-04-08

Nr	Tema	Person	Fråga/svar
1		CW	Kan du berätta lite om dig själv; hur gammal du är och vad du har för arbetsroll, vad du har för utbildning, och så vidare?
2		P1	Jag är 35 år gammal, har inte pluggat jättemycket, men en yrkesutbildning i ett år under år 2001. Började jobba direkt efter det, har jobbat jättelänge sen dess, alltså bara jobbat och jobbat, förutom ett år då jag pluggade på distans på högskola och Universitet. Det var mest allmänna programmeringskurser, alltså C och C#, Java och JavaScript, och HTML... ja, mest för att ha på pappret att jag faktiskt kunde de här grejerna. Sen har jag haft eget företag ett litet tag, och sen jobbat som konsult ett litet tag, och nu här på Avensia jobbar jag som Front End-utvecklare och har jobbat här i snart 4 år. Innan har det varit mycket blandat,

			både backend och frontend och lite allt i allo, men nu är det endast frontend.
3		CW	Vad är din erfarenhet inom just agila utvecklingsprojekt? Har det varit under en längre period, eller?
4		P1	Ja, lite till och från. Här på Avensia har jag bara kört det, och innan dess har det varit en hel del Vattenfallsmodellen. Jag har inte tänkt sådär vansinnigt mycket på vilken projektmodell det har varit, jag har bara jobbat och gjort det man blivit ombedd.
5	Fullt stöd från ledning	AO	Hur pass engagerad och insatt anser du att dina överordnade är till projektet du arbetar i för tillfället?
6		P1	Mycket. Tycker jag. Iallafall projektledaren, cheferna vet jag inte.
7	Fullt stöd från ledning	AO	På vilket sätt visar de att de är engagerade?
8		P1	Ja... de sitter mycket i möten? Och ja, det är daglig kontakt med både oss utvecklare och kunderna. Man får mycket information på båda hållen och... ja, man vet vad som händer i projektet genom projektledaren.
9	Rätt utvecklare i rätt projekt	JB	Känner du har ett personligt intresse i just de projekten du har arbetat i?
10		P1	Ja, man är ju alltid personligt engagerad och vill att det ska vara väl utfört och ska ha gjort ett bra jobb. Så proffsigt som möjligt.
11	Rätt utvecklare i rätt projekt	JB	Finns det några andra projekt som intresserar dig mer, som du hellre hade arbetat i?
12		P1	Jag vet att det är ett par som har startat den senaste tiden som jag är lite sugen att hoppa in i, men inte kunnat gå in i då jag är lite inlåst i mitt nuvarande projekt. Men annars tycker jag att det har varit ett roligt projekt.
13	God insikt i arbetslaget	CW	Hur pass god insikt i din kunskap och personlighet anser du att din projektledare har?
14		P1	Rätt så bra. Det är första projektet som jag jobbar med henne så det är ju inte 100 %. Och i början var det en del upplärningstid innan man kände varandra. Men nu när vi har jobbat sen i september, ett halvår, så lär man ju känna varandra mer och jag tycker nog att hon är rätt så engagerad, bryr sig, och känner mig rätt så väl.
15	God insikt i arbetslaget	CW	Återspeglas det här på något speciellt sätt? Till exempel hur uppgifterna fördelas?

16		P1	Ja, jag har ju fått mer ansvar efter tidens gång. Och kan få både beröm och kritik över sak man har gjort bra eller inte så bra.
17	Medvetet ökat självstyre	AO	Tar du egna initiativ eller inväntar du delegation från överordnad?
18		P1	Ja, i den mån det går så tar jag ju gärna egna initiativ. Man har ju sina uppgifter som man ska göra i JIRA och inom projektets ramar, och ibland finns det inte jättemycket möjlighet att ta egna initiativ utan man gör det man ska göra. Men i de tillfällen då det går så gör jag det gärna.
19	Medvetet ökat självstyre	AO	Så du föredrar att ta egna initiativ?
20		P1	Ja, får jag välja så gör jag det gärna.
21	Möten med relevans	JB	Hur ofta har ni projektmöten?
22		P1	Ja vi har ju stand-up varje morgon, där vi går igenom allas uppgifter och ser till så alla har något att göra under dagen, eller man kan säga till om man behöver hjälp med något särskilt. Sen kör vi ju sprintar på ungefär 3 veckor, så vi brukar köra i slutet av varje sprint dels en demo på kanske 3 timmar och dels en sprint-planering på 3 timmar, ungefär brukar det röra sig om. Där går man igenom projektet väldigt noggrant.
23	Möten med relevans	JB	Hur tycker du att dessa är sett till relevans och planering?
24		P1	Ja alltså det är ju en svår fråga, för att jag som "frontendare" är ju inte jätteintresserad av hur backend-systemen fungerar, och sitta och lyssna på alla dessa detaljerade genomgångar av hur systemen ska fungera är ju egentligen inte jätteintressant för mig. Men, det är ju väldigt svårt att sälla och säga "du ska sitta med på de här punkterna, men inte de här", så oftast blir det ju att alla sitter med på hela mötet. Och så får det ju vara, det är ju en form utav kunskapsspridning också och jag förstår bättre hur backend-systemen fungerar, men det kan kännas när man är pressad och har mycket att göra att behöva sitta och lyssna på de här mötena.
25	Samarbete mellan utvecklare och QA-personal	CW	Hur fungerar samarbetet mellan er utvecklare och kvalitetssäkringspersonal?
26		P1	Vi har inte direkt någon kvalitetssäkringspersonal.
27	Samarbete mellan utvecklare och QA-personal	CW	Anser du att det finns ett behov av någon kvalitetssäkringspersonal?
28		P1	Ja det tycker jag. Det tycker jag absolut. Vi är lite dåliga på det.

29	Samarbete mellan utvecklare och QA-personal	CW	Så du hade velat ha typ, "vi är utvecklare, men det finns personal som enbart utför kvalitetssäkring"?
30		P1	Ja, eller att någon har det på deltid, eller att man tar in någon några dagar i slutet av en sprint, eller någonting. Eller att vi själva får mera tid att kvalitetssäkra varandras uppgifter. Att vi har mer kodgenomgångar, eller parprogrammerar... jag tycker det finns så många alternativ för att få upp kvaliteten, och jag tycker att vi gör väldigt lite sådant. Vi skulle behöva göra mer.
31	Samtlig kod testbar/Standardisering av kod	AO	Vilka eventuella riktlinjer följer ni för att skriva testbar kod, har ni någon kodstandard?
32		P1	Nu kan jag ju bara svara för frontend, och vi försöker köra kodstandarder. Men jag känner iallafall att vi inte har kommit hela vägen med detta, och vi diskuterar detta under de frontend-möten som vi har varannan månad ungefär. Vi har diskuterat det mycket, och de flesta av oss har försökt köra efter någon standard, men eftersom det kommer så mycket nya hela tiden så blir det svårt att anpassa sig efter dem eller försöka få dem att anpassa sig efter oss. Det kan vara lika svårt åt båda hållen. Och alla har inte så många idéer om det, och just inom frontend finns det så vansinnigt många standarder att det är svårt att säga vilken vi ska köra efter.
33	Samtlig kod testbar/Standardisering av kod	AO	Ni har inte utvecklat någon egen standard, liksom "såhär gör vi"?
34		P1	Vi har försökt. Men... det har inte riktigt slagit igenom.
35	Formell kodgenomgång	AO	Sker det någon formell kodgenomgång i ert projekt?
36		P1	Nej.
37	Insikt i samtliga moduler	AO	När du utvecklar, har du övriga kodmoduler i åtanke?
38		P1	Rent generellt modultänk försöker jag ju ha, att kunna återutnyttja kod och bygga moduler som man kanske till och med kan använda i andra projekt. Men vi får ju inte jättemycket tid att dela sådant med andra projekt, speciellt inte vi i frontend. Vi har diskuterat också det mycket på frontend-möten, det här med att dela moduler och ha någon typ av modulbibliotek. Men det har inte heller hänt och inte blivit genomfört. Men alltså mellan vissa projekt har vi ju plockat kod från varandra, men det är ju sällan den koden då är kvalitetssäkrad.

39	Insikt i samtliga moduler	AO	Men när du utvecklar och om du varit i något projekt tidigare så vet du då att då kan du plocka med dig ditt?
40		P1	Ja precis, exakt.
41	Rätt mängd dokumentation	JB	Hur ser dokumentationen ut i era projekt?
42		P1	Dokumentation, vad är det? På frontend så gör vi inte jättemycket dokumentation iallafall, jag vet inte hur de gör i backend.
43	Rätt mängd dokumentation	JB	Så du helt enkelt anser att dokumentationen inte är tillräcklig, eller att den inte behövs?
44		P1	Ja alltså det beror ju på vilken typ av dokumentation som ska göras. När jag jobbade på en tidigare arbetsplats så gjorde vi väldigt mycket dokumentation, men det var ju mycket produktbeskrivningar. Såhär funkar den här appen eller sighten, och såhär ska den användas, men det berodde ju på att vi hade så många användare. Oftast var det inte jättekomplicerade grejer, åtminstone som jag satt med, och där var det väldigt mycket dokumentation. Men jag har inte överhuvudtaget suttit med dokumentation på Avensia.
45	Rätt mängd dokumentation	AO	Dokumentation kan ju också vara att kommentera kod och sådär också.
46		P1	Ja, i och för sig. Det är ju en grej, och det försöker vi ju göra. Det är klart. Men att skriva dokument är ju inget att vi gör, men vi försöker ju kommentera vår kod. En variant där är att skriva logisk kod som är lätt att förstå, alltså att koden förklarar sig själv genom att döpa variabler rätt. Det är väl så jag försöker göra, och att lägga kommentarer i koden.
47	Bryt ned stora projektuppgifter till mindre delar	CW	Hur angriper du större projektuppgifter?
48		P1	Jag börjar med att dels försöka bryta ned i mindre delar, och dels för att se om jag behöver hjälp av någon annan. Och sen att börja planera, och prioritera och se vad som är viktigast.
49	Bryt ned stora projektuppgifter till mindre delar	CW	Finns det något annat tillvägagångssätt som du kanske hade föredragit?
50		P1	Nja, eventuellt hade det kanske varit smart att kunna göra den här planeringen tillsammans med någon annan. Ha ett bollplank, och det gör jag väl om jag känner att jag behöver det. Men det

			har jag inte direkt känt än, men oftast sitter man inte ensam med en sådan stor grej.
51	Tydliga användarfall	AO	Vem formulerar era användarfall?
52		P1	Det vet jag inte riktigt. Det är något de gör på förstudien, och exakt vem det är har jag inte riktigt koll på.
53	Tydliga användarfall	AO	Är användarfallen användbara?
54		P1	Det är väldigt olika från projekt till projekt. I vissa fall vet jag faktiskt att det är kunden som formulerar användarfallen, och där kan det bli väldigt generellt... typ "åh vi behöver en sökfunktion", utan att specificera exakt vad de behöver: söka på vissa ord, eller ska det finnas "did you mean?", eller kategorisering, eller helt enkelt mer specifik beskrivning. För vår del är det ju mer specifik desto bättre, men det kan ju ha att göra med att den som skrivit användarfallen inte har koll hur en sådan modul fungerar. Och ibland specificerar de helt gålet, och jag har ibland varit med om "wow det här är ju superlätt", och ibland är det helt uppåt väggarna. Som i projektet jag sitter i nu så hade vi användarfall på samma sak i fyra olika sprintar. I ett annat projekt så var de så generella så vi fick sitta och gissa, och då kom det istället tillbaka change requests med "såhär ville vi inte ha det", och "såhär ville vi inte ha det". Det är jättesvårt att jobba efter.
55	Tydliga användarfall	CW	Hade du känt dig bekväm med att skriva användarfall? Eller likasom, förstår du tänket bakom dem?
56		P1	Jag hade nog kunnat göra det, men det är nog inte min favorituppgift som jag hade känt kul att göra. Jag förstår hur det funkar och tänket bakom såklart.
57	Kundinvolvering	JB	Hur involverad känner du att kunden är i era projekt?
58		P1	I det här projektet så är de väldigt involverade. Och det är jättebra, och det är en bra kund att ha att göra med också och de förstår systemen också. Men ibland har man varit i projekt där kunden inte förstår överhuvudtaget ett IT-projekt eller en webbsida. Men har man tur så är de lätta att vägleda, men har man otur så tror dem att de förstår fast dem inte gör det. Så det är lite olika.
59	Kundinvolvering	JB	Anser du då att kunden borde involveras mer eller mindre i projekt?
60		P1	Det beror helt och hållet på vilken kund man har. Det beror helt på deras kompetens och möjligheter och sådär.
61	Öppen, samt strukturerad, arbetsplats	CW	Hur fungerar ert kontorslandskap?

62		P1	Det fungerar ju bra sådär tycker jag. Det kan ju vara rätt hög ljudnivå och sådär, men jag har lätt för att skärma av. Men jag vet att det är vissa personer som inte har det, som kanske inte har det så lätt.
63	Öppen, samt strukturerad, arbetsplats	CW	Känner du att det påverkar ditt dagliga arbete?
64		P1	Inte för min del, för mig spelar det ingen roll. Men jag vet att andra hade svarat annorlunda.
65		AO	Innan vi avslutar; du tog upp tidigare att ni arbetar i JIRA. Vad är det för någonting?
66		P1	Det är där alla våra användarfall ligger och kan flytta runt dom, beroende på om vi har startat dem, eller avslutat dem, eller om de är pågående. Det är väldigt användbart och man kan dela upp det efter vilken utvecklare som jobbar med vilket användarfall.
67		AO	Är det någonting annat du vill tillägga?
68		P1	Nej.
69		AO	Då tackar vi för din tid!

7.2.2 Intervjuprotokoll 2

- Arbetsroll: Backendutvecklare
- Ålder: 27
- Kön: Man
- Datum för intervju: 2015-04-08

Nr	Tema	Person	Fråga/svar
1		CW	Kan du berätta lite om dig själv, din ålder, vad du har för utbildning?
2		P2	Jag är 27år och har utbildat mig på LTH, datateknik. Har arbetat i tre år snart som backend-utvecklare.
3		CW	Vad är din erfarenhet i agila utvecklingsprojekt, alltså din nuvarande samt tidigare?
4		P2	Vi jobbar ju ganska agilt här på Avensia. Så vi har nog arbetat med agil utveckling i tre år i ett antal olika projekt. Det finns olika former av agil utveckling men det vi håller på med här är nog alltid agil utveckling skulle jag vilja säga i alla fall.

5	Fullt stöd från ledning	AO	Hur pass engagerad och insatt anser du att dina överordnade är i projektet du arbetar i för tillfället?
6		P2	Vi har precis fått en ny projektledare som håller på att sätta sig in i detaljerna och så vidare, men hen är engagerad, absolut.
7	Fullt stöd från ledning	AO	Hur visar projektledaren att hen är engagerad?
8		P2	Vi har ju morgonmöten varje morgon, då ställer projektledaren frågor och är aktiv under mötena och det visar ju att man är engagerad.
9	Rätt utvecklare i rätt projekt	JB	Känner du att du har ett personligt intresse i det projektet du arbetar i?
10		P2	Nja, just projektspecifikt skulle jag nog inte säga att jag har ett personligt intresse, visst man kanske har ett personligt intresse för tekniken, men inte den projektspecifika sidan.
11	Rätt utvecklare i rätt projekt	JB	Finns det andra projekt som du hellre hade arbetat i här på Avensia?
12		P2	Nej, egentligen inte, inte just nu så, absolut inte.
13	God insikt i arbetslaget	CW	Hur pass god insikt i din kunskap eller personlighet anser du att din projektledare har?
14		P2	Inte jättebra tror jag inte, men det tror jag är mycket upp till utvecklaren själv. Om man är framåt, man pratar mycket, då får personer runt omkring en bra insikt i vad man kan och vad man vill. Om man är lite mer tillbakadragen och bara gör sin grej då tror jag inte att folk får så bra inblick i det. Det är utvecklarens eget ansvar att visa att man är framåt, skulle jag vilja säga.
15	God insikt i arbetslaget	CW	Du menar alltså att det inte ligger helt på projektledaren att ha god insikt utan mer på individen?
16		P2	Ja, eller båda två. Det är inte svart och vitt.
17	God insikt i arbetslaget	CW	Visar projektledaren på något sätt specifikt intresse för dig?
18		P2	Nej, det har jag inte varit med om speciellt ofta.
19	Medvetet ökat självstyre	AO	Tar du egna initiativ eller inväntar du delegation från projektledare eller överordnad?
20		P2	Jag tar egna initiativ, det tror jag att de flesta här gör. Vi har till och med en sådan där lista där man själv får komma med förbättringsförslag så gör vi det om vi har tid och kunden vill betala för det.

21	Medvetet ökat självstyre	AO	Föredrar du att ta egna initiativ eller föredrar du att bli delegerad?
22		P2	Det måste ju finnas en jämvikt där, man måste ju ha något att göra. Det måste ju finnas något att göra om man inte vet vad själv.
23	Möten med relevans	JB	Hur ofta har ni projektmöten?
24		P2	Vi har möten varje morgon klockan nio och sedan ett längre mer omgående varje fredag.
25	Möten med relevans	JB	Hur tycker du att dessa är sett till relevans och planering?
26		P2	Bra faktiskt, men jag tror att många upplever dem som inte bra. Det gäller att få med alla. Just relevans är viktigt så att man inte sitter på ett möte och pratar om något som inte berör en. Ofta har vi delat upp det på frontend- och backend-möten. Ofta behöver inte alla frontendutvecklare vara med på alla backend-mötena för det berör inte dem helt enkelt, vissa saker i alla fall. Så det är nog viktigt att tänka på så att man inte tråkar ut någon eller slösar tid.
27	Möten med relevans	CW	Så ni har mer eller mindre uppdelade frontend-möten och backend-möten?
28		P2	Ja, precis. Varje tisdag har vi ett renodlat backend-möte.
29	Samarbete mellan utvecklare och QA-personal	CW	Hur fungerar samarbetet mellan er utvecklare och kvalitetssäkringspersonal?
30		P2	I vårt fall får utvecklarna agera kvalitetssäkringspersonal. Vi sitter varje fredag en halv dag och gör testning. Vi har fyra timmar dedikerat till testning och går igenom allting vi gjort i veckan. Sedan har ju även kunden på sin sida anställda som är dedikerade till testning. Just i detta fall är den kopplingen ganska omständlig, de hittar ett fel, lägger upp det i deras system så har vi någon som läser och godkänner, och lägger upp det i vårt system och så vidare. Så det är ganska omständligt.
31	Samarbete mellan utvecklare och QA-personal	CW	Känner du att det att utvecklarna agerar kvalitetssäkringspersonal är bra eller hade du hellre sett att det fanns extern eller renodlad kvalitetssäkringspersonal?
32		P2	Jag tror att det hade varit bättre med renodlad faktiskt. För då ställer man krav på utvecklaren att specificera tydligt "Hur ska detta testas?" för om man testas det själv då specificerar man inte alltid bra hur det faktiskt bör testas och då när det kommer till kunden om det inte är upp-”specat” hur det ska testas så är det lätt att de missar saker.

33	Samtlig kod testbar/Standardisering av kod	AO	Vilka eventuella riktlinjer följer ni för att skriva testbar kod, har ni någon kodstandard?
34		P2	Nej, vi har ingen direkt kodstandard. Oftast blir det bra, men vi har ingen direkt kodstandard. Man får försöka bygga saker modulärt, avgränsa saker och försöka att inte ha beroenden i onödan.
35	Formell kodgenomgång	AO	Sket det någon formell kodgenomgång i ert projekt?
36		P2	Ja, det gör det faktiskt. På krångliga delar av projektet så måste man gå igenom det med någon annan. Det handlar egentligen bara om att man rycker tag i en kollega och säger "Nu ska vi kodgranska detta i tio minuter". Det är också bra för att sprida kunskap mellan projektmedlemmarna om hur saker fungerar.
37	Insikt i samtliga moduler	AO	När du utvecklar, har du övriga kodmoduler i åtanke?
38		P2	Ja, det har jag. [Visar ritning] Jag brukar rita upp såhär för att få en helhetsbild. Just detta specifikt är en funktion i projektet jag gjorde för två veckor sedan. Man måste ju tänka på hur olika saker påverkar varandra och tänka på allt som redan finns.
39	Rätt mängd dokumentation	JB	Hur ser dokumentationen ut i era projekt?
40		P2	Vi har lite standard för det och vi har avsatt tid för att skriva just dokumentation men det brukar inte följas upp speciellt bra. Det brukar bli lite hafsigt när man tänker "nu måste vi dokumentera" och skriver ihop något.
41	Rätt mängd dokumentation	JB	Anser du att dokumentationen är tillräcklig?
42		P2	Ja, faktiskt. Jag tycker inte att det behöver vara så mycket dokumentering, det är ju främst för användaren så de vet hur man ska använda produkten. Det skulle i så fall vara om produkten går vidare till andra utvecklare för att upprätthållas. Då kan det vara bra för dem att ha lite dokumentation också. Men skriver man kod på ett bra sätt med bra struktur så behöver man inte dokumentera så mycket tycker inte jag.
43	Bryt ned stora projektuppgifter till mindre delar	CW	Hur angriper du större projektuppgifter?
44		P2	Det blir att man sätter sig ner, läser igenom noga och tänker länge. Därefter ser man runt i koden och tänker ut vad det kommer påverka och så vidare. Helst ska man även läsa igenom användarfall som man ska göra ett par månader framåt och tänka

			på om detta kommer vara beroende på de kommande sakerna och hur för att få allt att lira tillsammans. Till slut får man sätta sig med en kollega och gå igenom, "så här hade jag tänkt lösa det, låter det vettigt?". Om man gör detta kommer man nästan alltid på något och då blir det bättre.
45	Bryt ned stora projektuppgifter till mindre delar	CW	Finns det andra tillvägagångssätt du hade föredragit när det gäller att angripa större projektuppgifter?
46		P2	Nej, det är nog väldigt viktigt att inte agera förhastat. Man vill nog gärna bara kasta sig ner och börja koda direkt men det är viktigt att tänka efter.
47	Tydliga användarfall	AO	Vem formulerar era användarfall?
48		P2	Bra fråga, jag tror att det är vår lösningsarkitekt och projektledare. Det är egentligen de två som är med på en förstudie där de går igenom med kunden innan projektet gått igång vad som ska göras och hur. Då skrivs dessa tillsammans med kunden.
49	Tydliga användarfall	AO	Är användarfallen användbara?
50		P2	Absolut, det är dem vi utgår ifrån hela tiden.
51	Tydliga användarfall	CW	Hade du kunnat formulera ett användarfall? Förstår du tankesättet bakom och hur det formuleras ner till hur man sedan skriver koden?
52		P2	Ja, absolut. Man tänker inte så mycket på koden när man skriver användarfall överhuvudtaget utan det är mer bara "detta ska funka" sedan får man göra koden bäst man vill.
53	Kundinvolvering	JB	Hur involverad känner du att kunden är i ert projekt?
54		P2	Bra, mycket skulle jag vilja säga. Kunden är här varje vecka.
55	Kundinvolvering	JB	Känner du att det är representativt för hur involverad kunden är i alla projekt eller bara just detta?
56		P2	Just det här projektet är väldigt krångligt, det är en massa småkludd överallt. Så kunden har suttit med för att kunna svara på affärsspecifika frågor. Det brukar inte vara så att kunden sitter med, men i detta fall har det varit så. Det är bra att man kan anpassa utifrån projektet. Just i detta fall behövdes det och då går det att fixa.
57	Kundinvolvering	JB	Anser du att kundinvolveringen bör öka eller minska?
58		P2	Det är som sagt väldigt projektspecifikt så det kan man nog inte ha en direkt standard för.

59	Öppen, samt strukturerad, arbetsplats	CW	Hur ser ert kontorslandskap ut och hur tycker du att det fungerar?
60		P2	Öppet landskap och ganska hög ljudnivå men det stör inte mig.
61	Öppen, samt strukturerad, arbetsplats	CW	Du tycker inte att det påverkar ditt dagliga arbete?
62		P2	Nej, absolut inte.
63		AO	Är det någonting annat du vill tillägga?
64		P2	Nej, det tycker jag inte.
65		AO	Då får vi tacka för din tid.

7.2.3 Intervjuprotokoll 3

- Arbetsroll: Backendutvecklare
- Ålder: 28
- Kön: Kvinna
- Datum för intervju: 2015-04-08

Nr	Tema	Person	Fråga/svar
1		CW	Kan du berätta lite om dig själv, din ålder, vad du har för utbildning?
2		P3	Ja, jag har pluggat samma sak som ni, så systemvetenskap, kandidatexamen. Jag är 28 år och jag har jobbat här på Avensia i fyra månader ungefär, så jag har inte så mycket erfarenhet om just E-handel, men jag har jobbat tre år tidigare som programmerare. Jag jobbar med backendutveckling.
3		CW	Vad har du för tidigare erfarenhet inom just agila utvecklingsprojekt?
4		P3	Inte sådär jättemycket. På mitt förra jobb jobbade vi i och för sig hyfsat agilt, men då hade vi en testgrupp som satt och testade och kvalitetssäkrade så vi som var programmerare satt ju inte alls med det.
5	Fullt stöd från ledning	AO	Hur pass engagerad och insatt anser du att den övre ledningen, alltså platschefen och VD:n är till projektet du arbetar i för tillfället?
6		P3	Jo, jag tycker att de verkar väldigt insatta och i och med att det är ett öppet landskap så blir det lite mer möjlighet att kunna diskutera med chef och VD.

7	Fullt stöd från ledning	AO	Hur visar de att de har engagemang för projektet?
8		P3	De har koll på det när man pratar med dem. De visar även intresse genom att ställa frågor om hur det går och liknande.
9	Rätt utvecklare i rätt projekt	JB	Känner du att du har ett personligt intresse i det projekt som du arbetar i?
10		P3	Ja, det är klart. Jag känner att jag har personligt intresse i allt jag gör, annars hade jag inte gjort det.
11	Rätt utvecklare i rätt projekt	JB	Känner du att det finns andra projekt här som intresserar dig mer som du hellre hade arbetat i?
12		P3	Det är svårt att säga, just nu sitter jag med just mitt projekt och det är ju det som intresserar mig. Det finns säkert andra intressanta projekt också men jag har för lite kunskap om dem för att kunna säga.
13	God insikt i arbetslaget	CW	Hur pass god insikt i din kunskap eller personlighet anser du att din projektledare har? Alltså dig som person.
14		P3	Jag tror att projektledaren har väldigt bra insyn och kunskap, Hon är en bra människokännare känns det som.
15	God insikt i arbetslaget	CW	Hur återspeglas detta? Hur märker du att hon har det om man säger så?
16		P3	Jo, det är sättet som projektledaren betar sig mot laget och hur hon pratar med lagmedlemmarna. Det verkar som att hon har koll på alla och vad de gör. Hon går ofta runt och frågar hur det går och man har möjlighet att prata med henne.
17	Medvetet ökat självstyre	AO	Tar du egna initiativ eller inväntar du delegation från projektledare eller överordnad?
18		P3	Jag försöker ta eget initiativ men om det är något jag är osäker på så går jag till en överordnad chef och frågar.
19	Medvetet ökat självstyre	AO	Vad föredrar du? Att ta egna initiativ eller att bli delegerad arbete?
20		P3	Både och. En kombination, men främst eget initiativ är väldigt positivt tycker jag, men att det ändå finns någon form av struktur och en mall på vad man ska jobba efter. Precis som vi har det nu, att vissa jobbar med JIRA och vissa i TFS.
21	Möten med relevans	JB	Hur ofta har ni projektmöten?
22		P3	Vi har ett "stand up" varje morgon om det är det du syftar på?

23	Möten med relevans	CW	Det är väl egentligen samtliga möten relaterade till projektet utöver de dagliga mötena?
24		P3	Ja, det finns ju andra möten också, nu i veckan till exempel ska vi gå igenom vår nuvarande lösning tillsammans och försöka hitta fler buggar och ge alla en tydlig överblick över hur den fungerar.
25	Möten med relevans	JB	Hur tycker du att dessa är sett till relevans och planering?
26		P3	Jag tycker att de är väldigt relevanta och följer planen.
27	Samarbete mellan utvecklare och QA-personal	CW	Hur fungerar samarbetet mellan er utvecklare och kvalitetssäkringspersonal?
28		P3	I projektet jag sitter i för tillfället är jag osäker på om vi har just kvalitetssäkringspersonal, det är nog lite att utvecklarna får agera det också. Det funkar ju bra eftersom det är samma personer.
29	Samarbete mellan utvecklare och QA-personal	CW	Enligt dig, tycker du att det borde finnas kvalitetssäkringspersonal som endast sitter och kvalitetssäkrar istället för att ni utvecklare gör det.
30		P3	Det är lite vårt eget ansvar som utvecklare att testa att det vi utvecklar funkar, så vi måste ju göra någon form av kvalitetssäkring innan vi skickar vidare.
31	Samarbete mellan utvecklare och QA-personal	CW	Så du tycker inte att det behovet finns?
32		P3	Det är kanske inte jättenödvändigt, men det är ju alltid bra om det finns.
33	Samtlig kod testbar/Standardisering av kod	AO	Vilka eventuella riktlinjer följer ni för att skriva testbar kod? Har ni någon kodstandard som ni går efter?
34		P3	Nej, inte riktigt. Det har varit uppe för diskussion inom laget att vi borde ha någon form av kodstandard och "code review" men det har inte blivit något av det. Vi har fokuserat på att arbeta med vår lösning istället, så det har inte funnits så mycket utrymme till det.
35	Formell kodgenomgång	AO	Sker det någon formell kodgenomgång av koden som skrivs i projektet? Gås den igenom av någon annan innan den skickas vidare?
36		P3	Ja, vi har bestämt att någon annan ska kolla igenom ens kod innan man skickar in.

37	Insikt i samtliga moduler	AO	Har du övriga kodmoduler vid åtanke när du sitter och programmerar så att det inte skapas onödiga beroende när du kodar?
38		P3	Ja, jag försöker ju tänka i de banorna men i och med att jag fortfarande är ny inom denna bransch så är det inte alltid lätt, men jag försöker ha det i åtanke.
39	Rätt mängd dokumentation	JB	Hur ser dokumentationen ut i era projekt?
40		P3	Det är ju lite blandat, det finns ju lite dokumentation samlad på vår gemensamma "workspace". Från mitt tidigare jobb är jag van vid att det fanns väldigt mycket dokumentation där alla buggar dokumenterades och så och så för att kunna återskapa buggen. Här kanske inte den typen av dokumentationen finns, men det finns annan dokumentation om t.ex. hur vissa funktioner fungerar och det är ju bra.
41	Rätt mängd dokumentation	JB	Anser du då att dokumentationen som görs här är tillräcklig?
42		P3	Ja, det tycker jag. Jag tror inte att fokus ligger på just dokumentationen som jag har förstått det.
43	Bryt ned stora projektuppgifter till mindre delar	CW	Större projektuppgifter, hur angrips dem? Om du blir tilldelad dem eller om en väldigt stor projektdel ska implementeras?
44		P3	Ja, jag har ju inte riktigt varit med om det ännu, men som jag har förstått det så blir det lite parprogrammering. Att det är fler som sätter sig i ett lag och diskuterar hur de ska gå till väga och lösa det.
45	Bryt ned stora projektuppgifter till mindre delar	CW	Har du någon idé på hur man skulle kunna sköta de större uppgifterna som du har observerat?
46		P3	Jag tycker att sättet som använts är ett bra sätt.
47	Tydliga användarfall	AO	Vem formulerar användarfallen i ert projekt?
48		P3	Det är lite blandat. Om vi upptäcker buggar så kan vi själva rapportera in dem och skriva en beskrivning så det är egentligen alla i laget.
49	Tydliga användarfall	AO	Är användarfallen användbara?
50		P3	Ja, det beror ju på hur mycket information man ger. Hur man kan återskapa ett fall. Oftast är de det.
51	Tydliga användarfall	AO	Känner du dig bekväm med att skriva användarfall?

52		P3	Ja, det gör jag.
53	Kundinvolvering	JB	Hur involverad är kunden i era projekt?
54		P3	Jag skulle säga att de är väldigt involverade. De vill ha dagliga möten för att följa upp och se vad som har hänt sedan senast.
55	Kundinvolvering	JB	Anser du att kundinvolveringen borde öka eller minska?
56		P3	Jag tycker att det är på en lagom nivå, det vill säga en gång om dagen. Sedan om det skulle dyka upp något akut så är det ju bara att kontakta oss eller att vi kontaktar dem. Så jag tycker att kommunikationen fungerar bra.
57	Öppen, samt strukturerad, arbetsplats	CW	Hur ser ert kontorslandskap ut och hur tycker du att det fungerar?
58		P3	Som jag kanske nämnde tidigare så är det ett öppet landskap vilket är en av anledningarna till att jag valde att arbeta här. Jag tycker att det är väldigt spännande och jag tycker det fungerar bra. Det är bra att kunna ha alla kollegor på samma plats.
59	Öppen, samt strukturerad, arbetsplats	CW	Skulle du då säga att det har en mer positiv påverkan på ditt dagliga arbete?
60		P3	Ja, det skulle jag säga. Det är inte så att man behöver knacka på någon dörr och fråga om man stör eller liknande utan här är alla på samma ställe så man ser alla.
61		AO	Du nämnde tidigare "JIRA" och "TFS". Kan du förklara dessa begrepp?
62		P3	"TFS" står för "Team foundation server" som är kopplat till "Visual Studio" där man helt enkelt kan rapportera in buggar och "tasks" samt "scenarion". Egentligen är det kundens miljö men vi använder det också för att fixa buggar som de rapporterar in. "JIRA" är vår egen miljö som vi använder för att rapportera in buggar och liknande.
63		AO	Innan vi avslutar, är det något du vill tillägga?
64		P3	Nej, det känner jag inte. Hoppas det var till någon hjälp.
65		AO	Absolut, det var det. Tack för att du tog din tid.

7.2.4 Intervjuprotokoll 4

- Arbetsroll: Lösningssarkitekt
- Ålder: 29

- Kön: Man
- Datum för intervju: 2015-04-08

Nr	Tema	Person	Fråga/svar
1		CW	Kan du berätta lite om dig själv; din ålder, arbetsroll, utbildning och så vidare?
2		P4	Ja, jag är 29 år gammal om en månad. Jag pluggade också här på LTH, på datateknik den femåriga master. Jobbat på Avensia sedan jag slutade, så det är snart tre år. Vad var det mer jag skulle svara på?
3		CW	Din arbetsroll här och nu.
4		P4	Ja, jag började som utvecklare, det är väl liksom fortfarande min formella titel. Det har väl blivit mer från att bara sitta med utveckling till att ansvara mer övergripande i projekt, är med i förstudier och hjälpa kunder att se sina behov och vad de behöver och sådana saker liksom. Men jag försöker hålla mig kvar lite vid utvecklingen fortfarande.
5		CW	Det är backendutveckling då?
6		P4	Precis, så att det är mycket C# och den typen och så sitter jag med mitt specialområde InRiver PIM.
7		CW	Din erfarenhet inom agila utvecklingsprojekt, alltså nuvarande och tidigare, hur ser den ut?
8		P4	Ja det är ju de tre åren som jag har jobbat här. Annars är det ju skolprojekt och innan när jag har jobbat har det inte varit så stora agila projekt utan det har oftast varit mindre projekt mellan kompisar. Skulle man titta företagsmässigt så är det ju de tre åren jag har varit här.
9	Fullt stöd från ledning	AO	Hur pass engagerad och insatt anser du att den övre ledningen, alltså platschefen och VD:n är till ditt projekt som du arbetar i för tillfället?
10		P4	Dem är ju så pass insatta så att de frågar "Hur går det?". Man får ju göra det rimligt liksom. Du kan ju inte ha att en person som är platschef som har kanske åtta till tio projekt under sig, ifall de ska hinna med att göra någonting annat så kan ju inte dem vara så detaljerat insatta utan det är ju oftast att de kollar liksom vad det är som händer i projektet, det är ju framförallt om någonting strular i projektet. Till exempel: vi har problem med det här, eller det här problemet är tungt för oss eller kunden strular med det här. Det är dem grejerna dem tittar på, tänker jag. Men sen

			vet jag inte exakt hur pass insatta de är, men det är min uppfattning det att just det dem tittar på är problemområden så får dem försöka sätta sig in i dem områdena liksom.
11	Fullt stöd från ledning	AO	Så då visar dem sitt engagemang just när det händer struliga grejer?
12		P4	Det blir ju oftast att... projektet försöker ju alltid lösa sina grejer internt. När man känner att det har blivit en konflikt eller blivit någonting som projektet har problem med att hantera själv, alltså vi har kommit till någon schism med kunden som vi inte riktigt kan lösa, då är det ju alltid det här med att man eskalerar det till en ledning. Kunden går över till sin projektledare på sin sida och pratar med den som är ytterst ansvarig där och då går det också upp ett steg och de vill inte prata med projektledaren här, då är det oftast högre roller. Dem personerna har ju också väldigt sällan koll på projektet. Dem har inte detaljkunskapen så ibland tar dem med sina projektledare för att de behöver detaljkunskap, men oftast när det är en konflikt så är inte alltid det intressant utan då tittar man.. "okej vi skiter i vad det är för konflikt, vad är det vi ska göra för att lösa den eller vad är det nu bråkar om nu, "ah det är pengar eller det är tid, nu har jag missat någonting, vad är det vi ska göra för att sluta bråka liksom". Så att oftast när de behöver ta in VD:n och de här posterna liksom då behöver du inte veta detaljerna om projektet. Det är liksom inte intressant att han sa si och han sa så.
13	Rätt utvecklare i rätt projekt	JB	Känner du att du har ett personligt intresse i det projekt som du arbetar i?
14		P4	Det blir det, man sitter och engagerar sig så pass mycket med någonting så tillslut blir det så, ja det gör man.
15	Rätt utvecklare i rätt projekt	JB	Så du känner inte att det finns andra projekt som intresserar dig mer, som du hellre haft eller det..?
16		P4	Alltså föregående projekt som man har haft blir ju lite som en vaggasom man tar han om så fort som någon gör någonting i ett gammalt projekt som man kanske satt jättelänge i så blir det liksom att man reflekterar mycket till det projektet. "Vad gjorde vi i det projektet, vad fungerade bra, vad fungerade dåligt, och liksom vad blev resultatet av det projektet" Det blir ju väldigt naturligt om man har suttit i ett år varje dag och vare sig om man säger att detta är ett jobb eller inte så liksom spenderar du hälften av din vakna tid med att göra någonting så kan du inte låta bli att att man på något sätt blir engagerad även om man försöker skjuta ifrån och göra någonting annat. Jag tror att det är vetligt, om du ska tycka det är kul, det du gör, så måste du våga bli lite engagerad. Så att, ja det blir jag. Det gäller ju även tidigare projekt som man gjort så blir man också engagerade i, man fortsätter på något sätt att vara engagerade i dem.

17	Rätt utvecklare i rätt projekt	CW	Det är inte så att du känner att ett annat projekt här på Avensia skulle passa dig bättre?
18		P4	Det är ju alltid det här med att gräset alltid är grönare på andra sidan. På ett sätt tror jag att det kan vara lite skiftande. På något sätt är det rätt skönt när man varit i ett projekt länge och man vet bakgrundsinformation och man vet mycket sådana grejer. När du suttit i ett projekt ett längre tag har du kommit in i grejerna, du vet vilka personer du ska prata med hos kunderna och du har börjat hitta i projektet och vet lite hur det är strukturerat. Men samtidigt när du har suttit i det ett tag blir det ju tråkigt, ifall du känner att man gör bara samma sak liksom och då kanske man vill vidare till ett annat projekt för att man känner att "nu kan jag dem här grejerna, nu vill jag till ett annat projekt för att det är ny teknik eller någonting annat som händer där liksom. Tidigare har vi haft väldigt mycket att vi gör projekt och sen är vi färdiga under ett års tid max. Nu har vi kommit till kunder som har mycket större projekt där folk blir att dem efter ett eller två år har så pass mycket att göra hela tiden så det känns som att jag skulle kunna sitta i detta projekt i två år eller mer för att det finns så pass mycket att göra då kunden har en kontinuerlig beställning, och då kan jag tänka mig att personer känner att de vill göra någonting annat. Det är nog personligt hur man är funtad tror jag.
19	God insikt i arbetslaget	CW	Hur pass god insikt i dig, i din kunskap och personlighet, anser du att din projektledare har?
20		P4	Hur jag är som person...
21	God insikt i arbetslaget	CW	Ja men, hur god insikt anser du att projektledaren har?
22	God insikt i arbetslaget	JB	Även vilken kunskap du har, vad du är extra bra på.
23		P4	Det är ju väldigt personligt, hur bra personen är på att känna av folk och liksom. Min projektledare hon är ju duktig på att, hon känner ju folk hyfsat bra, hon lär snabbt och ser. Hon vill ju gärna se vad folk klarar och inte liksom. Men sen är det ju väldigt svårt, har du en person som har varit anställd här två-tre år då vet dem nästan alla personer på det kontoret den jobbar på. Den vet lite vad de olika personernas expertområden ligger på, men kommer det en projektledare och har arbetat här i ett halvår eller ett par månader då blir det väldigt svårt. För dels vet de knappt vilka system som är intressanta i det här området och sen veta vad varje person kan och inte, det är jättesvårt.
24	God insikt i arbetslaget	CW	Upplever du att din projektledare nu tilldelar dig rätt uppgifter om man säger så utefter din kunskap?
25		P4	Ja, det är ju inte så med pekpinne, "var så god o gör", utan det är ju lite såhär "kan någon ta tag i den här uppgiften, kan någon göra det här". Men sen är det också att, till exempel mitt område

			är PIM, då har jag sagt att jag kan det här området men jag skulle gärna vilja lära mig lite av de andra grejerna också, och då kanske hon istället för att ge mig alla PIM grejer istället säger "kan du hjälpa någon annan igång med de här grejerna så kan du göra någonting annat". Det kanske inte alltid är det optimala men samtidigt sprider man ju kunskap och det skulle jag se som bättre i längden. Så att utnyttja vad folk är bra på det är ju liksom väldigt lätt att tänka "han eller hon har alltid gjort detta, dem är experter på det". Men det är kanske inte det som de ska göra för då blir det en "go-to-guy" och då har du plötsligt en person som är duktig på det och då är du ett känsligt läge, slutar den personen då är du körd.
26	Medvetet ökat självstyre	AO	Tar du egna initiativ eller inväntar du delegation från projektledare?
27		P4	Egna initiativ. Vi sitter i så små projekt där projektledaren inte har 100 procent koll. Det är vi som utvecklare som ser vad det är som händer och vad det är vi gör så vi måste ta egna initiativ och till exempel säga "ja, vi löste denna bugg men den kommer inte hålla i längden" eller "Detta är inte optimalt". Det händer ju jätte ofta att vi som utvecklare kommer till vår projektledare och säger "vi måste ta tag i den här biten" eller så säger vi inte någonting utan bara gör det. Det är ju det vanligaste, att man tar en timme och bara fixar vissa grejer, det tar man ju alltid alltför ofta. Så att egna initiativ är jätteviktigt för det går inte att ha en projektledare som går o håller dig i handen och pekar med pinnen och säger åt dig att du måste göra det här. Projektledaren måste ju ge övergripande direktiv över vilka punkter som är viktiga och vilka som ska arbetas med men under utvecklingens gång måste man ju vara självgående.
28	Medvetet ökat självstyre	AO	Så du föredrar egna initiativ i projekt?
29		P4	Mm. Men självklart måste du ju få en riktpinne på vad vi ska koncentrera oss på, vad är det kunden tycker är viktigt. "Här sitter jag och bygger en cool funktion", men de har inte frågat efter det. Så får man ju inte göra.
30	Möten med relevans	JB	Hur ofta har ni möten inom projektet?
31		P4	Ja vi försöker ju ha "daily standup". För att allihop ska mötas, men sen är det ju också utefter vem som behöver prata. Vad räknar du som ett möte? Ibland vänder jag mig om och säger "oh, jag måste prata med dig och dig för jag har en konstig grej här", är det ett möte? Folk försöker känna varandra och då tar du hjälp av varandra. Så.. möten, det är ju minst en gång om dagen men alltså att prata med folk det gör du ju hela tiden.
32	Möten med relevans	AO	Jag tänker inbokade möten, med satta tider under projektets gång.

33	Möten med relevans	CW	Där man har ett tema, vad man ska prata om på detta möte på denna tid.
34		P4	Menar ni då alla måste vara medbjudna?
35	Möten med relevans	CW	Ja, eller delar av projektet.
36		P4	Alltså det är ju verkligen hela tiden. Folk, en eller två personer har möten hela tiden med kund, till exempel jag har kanske ett test-möte med kunden samtidigt har någon annan i projektet ett möte angående profiltjänsten med någon annan hos kunden. Sen har någon annan möte om hur POS-hantering styrs osv. Det är ju dialoger hela tiden som vi håller med kunden, det är olika personer, ibland är man en och ibland är man fyra på mötet. Projektet jag sitter i nu är väldigt mötescentrerat skulle jag säga, det är mycket möten för att vi är många i teamet. Sitter du i mindre projekt blir det mindre formella möten.
37	Möten med relevans	JB	Hur tycker du att mötena ni har är, sett till relevans och planering?
38		P4	Jag tycker att folk oftast är rätt bra på att hålla fokus. Oftast har folk en känsla av att de vill tillbaka och jobba, de har grejer att göra. Om man har den känslan att man har saker att göra och känner att det som sägs inte bidrar då säger man nej för att man inte vill sitta och slösa bort sin tid, man vill hellre göra någonting vettigt. Så att jag tycker att det oftast blir en naturlig filtrering av att man slipper onödiga möten, men sen så blir det ju alltid när man ska samlas alla att det blir till exempel en back-end diskussion under exempelvis standup och då ser man front-end personerna eller andra som inte är insatta i den diskussionen bara: "jaha, måste ni snacka om detta nu?". Så att det kommer alltid vara under mötets gång att vissa personer känner sig mer engagerade och vissa mindre. Och då försöker man skala av mötena. Sen kommer det upp om man ska ha ett övergripande möte där man ska gå genom den övergripande strukturen på siten, då måste alla vara med för att vi behöver en helhetsbild men sen kommer det bara vara att man får sitta och ta in kunskap. Bottom line är att de flesta försöker säga nej till möten om de inte känner att de är givande.
39	Samarbete mellan utvecklare och QA-personal	CW	Hur fungerar samarbetet mellan utvecklare och er kvalitetssäkringspersonal?
40		P4	Vi är vår egen kvalitetssäkringspersonal. Vi har ingen testavdelning, utan vi utvecklar och testar, det är samma sak för oss. Vi skjuter mycket acceptanstestning till kunden för det är dem som ska göra det. Det är mycket från utvecklare till utvecklare hur mycket man testar sin grej innan man lämnar över den, vissa tycker att kunden ska testa det så varsågod. Man vill ju också ha

			lite "hygienkrav", det ska inte smälla på ful grejer utan dem ska ju testa att helheten fungerar. Vad var frågan igen?
41	Samarbete mellan utvecklare och QA-personal	CW	Hur samarbetet mellan utvecklare och kvalitetssäkringspersonal ser ut. Så om du säger att ni som utvecklare sköter kvalitetssäkringen, hade du velat se att ni tog in extern kvalitetssäkringspersonal som ett steg emellan utvecklare och kunden?
42		P4	Ja, jag tror det hade blivit bättre. Ibland är man lite för lat och för snabb och bara skyfflar genom grejer. Ibland tycker jag att saker man skickar iväg till kund är lite pinsamma. Man tänker "kunde vi inte sett det där innan vi lämnade bort det". Men det är också från projekt till projekt, i ett av de större projekten jag satt i innan började kunden undra "testar ni inte det här själva?" och vi började känna att vi inte levererade så jättebra kvalitet och då började vi ta in metodik i projektet för hur vi skulle göra bättre testning. Då gjorde vi till exempel att om jag blev klar med en grej så satte jag det till att nu får någon testa det här. Då är det någon annan i projektet som ska ta tag o testa det innan dem från börja med sin egen utveckling. Man fick tillbaka grejer relativt snabbt, oftast inom en dag hade någon annan testat det och då blir det inte att man "deployar" och två veckor senare kommer någon och säger att din grej inte fungerar och så tänker man "oh vad har jag gjort" utan då fick man feedback, du kunde enkelt sätta dig in i saker du gjort. Det tyckte jag var jättebra, dels spred du kunskapen inom teamet. För det är ju också det att om du har en bra "test-organisation" så måste du utbilda dem hur ska detta fungera och hur är det tänkt att fungera, vilka fall-gropar är okej och vilka är inte okej. Har du ett team är det ändå bra, om jag utvecklar en grej och du testar den är det bra för du kanske ska göra en påbyggnad till den kod jag gjorde och den kanske du gör om två veckor, då är det bra att du sett lite av den. Jag tänker att som team är det bättre, men finns det här klassiska att man blir stressad och så slutar man testa. Det är klart bra att ha en kvalitetssäkringsorganisation som hade höjt kvaliteten men jag tror att i de projekten vi sitter i nu, lite mindre projekt där man levererar till en kund som sitter och testar också, är det bättre att vi utbildar oss själva till att faktiskt genomföra testningen. Men då är det viktigt att det finns tid till att testa.
43	Samtlig kod testbar/Standardisering av kod	AO	Vilka eventuella riktlinjer följer ni för att skriva testbar kod, har ni någon standard?
44		P4	Nej. Det sägs i många projekt som börjar att "här ska vi köra TDD [Test driven development], nu ska vi köra testdrivet, vi ska göra det testbart." Men det är många som inte vet vad det innebär och så gör folk på sitt gamla vanliga sätt och så blir det inte av, eller 25 procent testas. Nej vi har inget som är satt.
45	Formell kodgenomgång	AO	Sker det någon formell kodgenomgång?

46		P4	Det är också olika från projekt till projekt. I mitt specifika projekt har vi sagt att vi ska ha "code reviews", att vi ska gå genom kod för att dels sprida kunskap och dels kontrollera vanliga fel. Det har vi sagt att vi ska göra och vi har satt upp en metodik för hur vi ska göra det men i arbetssättet som vi just nu befinner oss i, PRE-FAT, där vi sitter och gör dagliga deployer och vi har små bugg-fixar, då blir det lätt att folk struntar i att göra det, det blir inte av. Det är mer när man gör en stor funktion eller stor ändring, då kanske det blir att folk gör det men det är också en vana och man måste få folk trygga med det.
47	Formell kodgenomgång	AO	Den här metodiken ni hade tagit fram för just formell kodgenomgång, hur ser den ut, hur skulle ni gå tillväga?
48		P4	Vi sa att man skulle få så snabb feedback som möjligt, så fort du kände att "nu är jag färdig, nu vill jag checka-in det här", då ska man bara vända sig om och fråga närmaste utvecklare om vi kan köra en snabb code review. Det ska vara något som tar mindre än 10 minuter. Då sa vi att man får förklara vad man har gjort och den andra personen får titta lite på det och så diskuterar man kring det och får feedback och så jobbar man vidare på det utefter det. Det är alltså väldigt informell kodgranskning, inte att man går iväg till ett rum, utan man vänder sig om och säger "varsågod och titta på vad jag gjort". Så väldigt informell. Vi vill prioritera att folk skulle göra det, när man börjar med saker folk inte är vana vid då måste tröskeln dit vara enkel så att folk gör det. Det är återigen ett team där ingen står med en piska och säger "varsågod gör detta" utan vi måste själv göra det. Vi ville sprida kunskap, det skulle vara enkelt med små, snabba kodgranskningar, inte att vi samlas 10 personer och tittar tillsammans på detta.
49	Insikt i samtliga moduler	AO	När du sitter och utvecklar själv, har du de övriga kodmodulerna i åtanke?
50		P4	Vad menar du?
51	Insikt i samtliga moduler	AO	Så att det inte skapas onödiga beroende.
52		P4	Självklart! Det försöker man ha så mycket man kan, men det är det att man inte vet hur allting ser ut och det är alltid svårt att veta. Ifall man sitter fem personer och utvecklar kod, då hinner man inte läsa vad de andra fyra gör. Så man försöker alltid kolla om det är någon annan som gjort något liknande som jag kan använda. Och det där med beroenden det är sådana grejer det är bra att ha kodgranskning för, då har man en till person som tittar på det. Det är klart man vill göra det, men sen är det svårt.
53	Rätt mängd dokumentation	JB	Hur ser dokumentationen ut i ert projekt?

54		P4	Nada. Nej. Dokumentation, nu bad kunden om dokumentation men det är en klassisk sak att skära bort i scopet när kunden vill prioritera. När man säljer in till en kund skär de gärna bort dokumentation för att det ska bli billigare och sen inser de kanske senare att det kanske hade varit bra att ha, så då får man skriva till det senare. Vi har skrivit dokumentation för projektet för vilka moduler som finns och hur de ska se ut men det handlar åter igen om en vana, vad folk är vana vid. De flesta personer är vana vid att inte skriva någon dokumentation, man kanske gör en minimal kommentar i sin kod men det är ingenting som kunden ser. Så det är också det att, hur länge kommer denna dokumentation överleva? Hur länge kommer den vara relevant? För det handlar om en vana, du ska vara med om att ja, "nu ändrar jag någonting här och ska faktiskt uppdatera det någonstans". De flesta kommer bara ändra och inte tänka på att det finns en dokumentation.
55	Rätt mängd dokumentation	JB	Så, den mängd dokumentation ni har i dagsläget, anser du att den är tillräcklig eller?
56		P4	Tillräcklig för vadå? Jag ser det lite som att vi, vi som utvecklare är också dokumentationen i våra huvuden liksom. Det blir lite att dem kommer och frågar oss och vi har ju lite utbildning och kan säga såhär ska ni använda detta, såhär ska ni göra och sen blir det lite upp till kunden att de får skriva sin egen interna dokumentation för hur de ska använda det för att sen utbilda sin egen personal. Det svåra är att hålla dokumentationen levande, man skriver den en gång, fine, sen är det någon annan som kommer som ska veta att de ska uppdatera den.
57	Bryt ned stora projektuppgifter till mindre delar	CW	När det kommer större projektuppgifter, alltså om det är en stor förändring eller något liknande, hur angriper ni den då?
58		P4	Bryter ner den. Det måste du göra för att kunna göra det parallellt, för att kunna göra det övergripigt och för att kunna göra estimat så måste man bryta ner uppgiften om den är för stor.
59	Bryt ned stora projektuppgifter till mindre delar	CW	Och du tycker det fungerar i dem lägen där det skett?
60		P4	Ja. Självklart kan jag bara sätta igång och börja pilla på en sida och göra sin väg in, men oftast är det inte bra. Det gör man nästan mentalt också, får du en uppgift för att kunna "wrap your head around it" så måste man inse vad uppgiften innebär och då börjar man kategorisera, "jag ska göra det, jag ska göra det, jag ska göra det". Man gör det mentalt. Det tror jag de flesta gör om man ska lyckas få någon som helst överblick över det.
61	Tydliga användarfall	AO	Vem formulerar era användarfall?

62		P4	När vi börjar med ett projekt, då har vi en så kallad förstudie då man har en massa workshops, man sitter och diskuterar men kunden. Man försöker ta representanter från kundens olika avdelningar och så sitter man ett par personer från Avensia och pratar om vad det är dem vill göra, vad är det dem vill lyckas med, deras mål och svårigheter. Efter det skriver vi användarfallen utifrån de här diskussionerna och försöker mynta ner vad det är vi har suttit och diskuterat och sedan ta ner till faktiska användarfall. Så det är oftast vi som skriver dem. Sen sitter vi och estimerar dem och sen får kunden godkänna dem. Där har vi ett glapp för kunden har oftast svårt att.. kunden: "okej, vi satt och diskuterade om det här, representerar detta användarfallet det vi satt och diskuterade kring?" Det kan vara väldigt svårt för kunden att inse, men då försöker man ha skisser och annat till. Det är vi som skriver användarfallen och sen i takt med att projektet börjar bli mer och mer färdigt, kunden blir mer och mer bekväm med användarfallen så börjar dem skriva önskemål om vad de vill göra själva. Dessa blir då CR eller vidareutvecklingar, då kanske de börjar skriva användarfall.
63	Tydliga användarfall	AO	Är det några specifika roller i ert projekt som skriver användarfallen eller är det vem som helst?
64		P4	Det blir oftast den som är utsedd till lösningsarkitekt. Menar du de första användarfallen som skrivs?
65	Tydliga användarfall	AO	Ja precis.
66		P4	Ja precis, på de här workshopsen så är det alltid minst projektledare och lösningsarkitekt. Lösningsarkitekt ska kunna så pass mycket om alla system så den kan svara på tekniska frågor, vad som är möjligt och vad som inte är möjligt. Det är den personen som har den tekniska kompetensen som skriver användarfallen oftast med hjälp av projektledaren. Men det är den personen som har den tekniska kunskapen som skriver användarfallen. Vi har varit med om att en person som inte har kunskap skriver användarfall och så märker man sen att "det här kan vi inte alls göra". Då sitter man lite i skiten sen.
67	Tydliga användarfall	CW	Det är viktigt att rätt person skriver användarfallen?
68		P4	Ja. Jag har varit med om att en person som har haft lite bråttom skriver "ja men det ska fungera såhär" och sen när man kommer fram och ska utveckla det här så inser man att det inte går. Vi kan inte leverera det här systemet, de komponenterna vi har kan inte klara av det här. Då är det inte så kul att sitta och säga till kunden att vi inte kan leverera detta.
69	Tydliga användarfall	AO	Dem användarfallen ni har, är dem användbara?
70		P4	Vad menar du med användbara?

71	Tydliga användarfall	AO	Alltså följer ni dem när ni utvecklar?
72		P4	Ja. Det är det kunden tittar på. I slutändan är det ju det dem beställer. Då säger vi att ni får dem här användarfallen och det är dem vi baserat estimaten på och det är insäljningen till projektet innan man sätter igång och sen för varje sprint levererar vi några användarfall i taget och det blir dem de testat utifrån. Då märker dem om det är detta dem ville ha, så då blir det kanske förändringar på användarfallen eller diskussioner kring det. Dem är alltid bra att ha att diskutera kring, men ju längre projektet går desto mer blir original användarfallen svårare och svårare att använda för att det var så länge sedan dem skrevs och det var diskussioner kring dem, de blir snabbt irrelevanta. "Det här beställde ni för ett år sedan." - "ja men då visste vi inte det vi vet nu".
73	Tydliga användarfall	AO	Känner du dig bekväm med att skriva användarfall?
74		P4	Det har jag gjort. Att skriva radvis med text är inte min styrka. Det har jag aldrig tyckt var jätteroligt. Roligt eller ej, men det kan jag väl göra. Det viktiga är att du har grundkunskapen, att man var med på förstudien och vet lite vad de vill ha. Det svåra är när man suttit i en vecka och diskuterat kring massa möjligheter och inser att man tillslut inte vet vad de vill ha, det är jättesvårt. Att skriva ett separat användarfall, "som användare ska jag kunna filtrera på en söksida" - det är inte så svårt, då skriver du det. Att skriva ett användarfall är inte svårt, det svåra är att bena ut vad det faktiskt var de ville ha.
75	Kundinvolvering	JB	Hur involverad är kunden i era projekt?
76		P4	Det är också jätte olika från projekt till projekt. Vissa kunder är knappt med och sen när vi levererar så börjar de titta på det. Medan vissa är jätteengagerade för att det är "core of their business", så de sitter och vill vara med, man vill ju ha en engagerad kund för du får ett bättre slutresultat snabbare, med snabb feedback. Det är också från kund till kund hur kunden reagerar med ändringar och allt sådant. Man försöker alltid få kunden engagerad för att det gör ditt jobb enklare och de får ett bättre resultat. Man vill få kunden engagerad men hur pass engagerad kunden är, på vilken skala ska jag ange det? Vi försöker identifiera vilka personer som har koll på vad och sen prata med dem när det gäller deras område. Att få projektmedlemmarna att vara bekväma med kunderna, det är jätteviktigt för då känner man att är någonting konstigt skriver man ett mail eller ringer till kunden och man känner att kunden är engagerad samt att kunden känner att vi är engagerade i vad kunden vill ha och känna att "dem är så pass engagerade så om jag ringer till dem så kommer de prioritera det samtalet. Då kommer de få ett bättre slutresultat. Det är jätteolika från kund till kund. Även om kunden är engagerad så

			gäller det att få projektmedlemmarna att vilja ta kontakt med kunden.
77	Kundinvolvering	JB	Anser du att kundinvolveringen borde öka eller minska över lag?
78		P4	Under projektets gång menar du? Eller menar du vad man ska sträva efter?
79	Kundinvolvering	JB	Både och egentligen.
80		P4	Öka. Jag tror det blir bättre ju mer kunden är engagerad och så blir det oftast enklare för dem ju längre projektet går. Det blir mer konkret och när det närmar sig deadline blir de mer engagerade.
81	Kundinvolvering	CW	Det kan inte bli så att det blir för mycket kundinvolvering också?
82		P4	Jag har inte sett det än. Till exempel i ett annat projekt som vi har skickar de hit en person från kunden som satt här på kontoret i flera veckor, den satt med oss så man kunde vända sig om och fråga personen när som helst. Varje liten oklarhet gick att lösa väldigt snabbt. Sure, ibland kanske det är jobbigt när man säger rakt ut "oj, denna lösning vi gjort här var inte så snygg.". Då är det inte så jättekul att ha kunden precis jämte liksom. Jag tror generellt sett att det blir bättre om man har en bra relation med kunden. Har man en dålig relation, den är stel, det är mycket bråk, man bråkar om pengar och budget och hej och hå, då är det inte jättekul att ha kunden där. Då kanske man måste jobba sig runt vissa saker. Har man en vettig relation med kunden ser jag det bara som en bonus.
83	Öppen, samt strukturerad, arbetsplats	CW	Hur ser ert kontorslandskap ut och hur fungerar det för dig?
84		P4	Det är ett öppet landskap, man har en massa skrivbord som står i olika grupper och man försöker samla team i de här olika grupperna. Så har det till exempel inte sett ut i på vårt kontor i Helsingborg, utan där hade man mer rum där det stod skrivbord i och så delade man upp hur folk satt i de här olika rummen. Jag har suttit i 1,5 år i Helsingborg och 1,5 år i Lund så jag har sett lite båda världarna. Jag vet till exempel också att när de flyttade för ett år sedan i Helsingborg till ett större ställe där det blev lite mer landskapskänsla så var det många som klagade på att det blev mer högljutt. När jag kom till ett landskap och man började jobba där så är det klart att det är mycket intryck och det är folk som pratar och sådant men det blir väldigt socialt och väldigt trevligt. Det betyder att du kan ta hjälp och ta diskussioner med folk som inte sitter i ditt projekt. Framförallt för folk som sitter i mindre projekt så blir det kanske trevligare att sitta i ett landskap där det kanske är lite mer liv och rörelse, det händer grejer

			och man kan prata med folk som passerar. När man sitter i ett rum så blir man ju lite mer isolerad, man ska gärna vara ett par stycken för att det ska bli en god stämning men det blir samtidigt att du diskuterar mycket mindre med andra utanför projektet. Ljudnivån blir kanske högre men de flesta löser det med att ta på ett par hörlurar och så lyssnar de på musik när de verkligen behöver koncentrera sig. Det fungerar för vissa och vissa tycker att det blir högljutt men de flesta personer vänjer sig rätt snabbt.
85	Öppen, samt strukturerad, arbetsplats	CW	Hur tycker du att det påverkar ditt dagliga arbete med det här öppna kontorslandskapet? Du sa att du hade lite att jämföra med på hur det var i Helsingborg.
86		P4	Ja precis, när jag hade suttit i Helsingborg och suttit mer i rum så när vi flyttade till ett mer öppet landskap då kände jag själv att det blev högljutt och man reagerade plötsligt lite igen på att man vant sig men efter några veckor så tänker man inte mer på det, det gör jag inte heller när jag är i Lund heller. Jag tänker sällan på att det skulle vara högljutt, man filtrerar bort lite av bruset, sen om det påverkar en undermedvetet det kan jag inte svara på. Däremot känner jag att så fort man blir lite trött eller nu behöver jag fokusera så kör jag på musik, musik fungerar jät-tebra. Att ta på sig ett par stora hörlurar och så ignorerar man allting annat. Folk ska fortfarande känna att de kan prata med en, knacka på axeln eller vinka åt en.
87		AO	Innan vi avslutar, är det någonting du vill tillägga?
88		P4	Jag tror att någonting som man ofta säger, som jag tror jag också sa innan med angående testning och kvalitetssäkring, är att "man måste få tid till det". Ja, initialt måste du få en tid till att sätta igång din process men jag tror att helheten kommer göra att det tar mindre tid. Du behöver vid färre tillfällen sätta dig tillbaka in i gamla grejer som du gjort eller sitta och analysera grejer som du gjort. Om du får snabbare feedback på att någonting är galet så kan du lättare... Så är det alltid, ju längre och längre du kommer in i processen och får reda på att någonting är fel ute i produktionssystemet, då blir det rätt svårt att analysera vad det är som har gått fel. Om du hittar det när du debuggar då kanske du kan lösa det på fem minuter. Folk måste inse att man får ge lite tid för att det ska generera att det kommer ta mindre tid i slutändan. Ofta är det väldigt kortsiktigt, folk har inte den slutledningsförmågan utan tänker: "nu har vi väldigt bråttom" och så kapar man av saker som man tycker att det är "over-head". Jag tror att man förlorar på det i slutändan. Det är jätkligt lätt för projektledare och andra personer att inte se att det behövs, för man inte ser slutmålet. Vi är många utvecklare som tycker det är skittråkigt att testa och man tycker det är färdigt.

89		CW	Någon mer helhetsbild eller ett måltänkande och inte bara det man sitter med för tillfället.
90		P4	Precis. Det är också en grej som jag snackat lite om, det är med att arbeta med user stories och punkter som en projektmedlem älskar att kalla det, att när du sitter med en punkt så stirrar man sig blind på den lilla punkten och så gör man den utan att titta på hur den ska integrera i det stora hela. Du har estimerat tio timmar på denna punkt, och så tog den åtta timmar och så får du plocka undan den istället för att titta på hur den kommer fungera med resten av systemet och vad kan jag behöva fixa i ordning för att det ska bli bra helhet. Det är väl det som är den negativa grejen med att sitta med user stories för att du gör din lilla punkt och oftast så missar man helheten.
91		CW	Det blir insulärt på ett sätt kanske.
92		P4	Precis, det isolerar lite.
93		AO	Vi får tacka för din tid!

7.2.5 Intervjuprotokoll 5

- Arbetsroll: Projektledare
- Ålder: 39
- Kön: Kvinna
- Datum för intervju: 2015-04-09

Nr	Tema	Person	Fråga/svar
1		CW	Kan du berätta lite om dig själv? Din ålder, arbetsroll, utbildning, och sådär?
2		P5	Ja, jag ska fylla 39, så jag är född 1976. Jag har ju också gått Systemvetarutbildningen, Systemvetarprogrammet, och jag har en Magister där. Och jag gick ut 2001, och sen har jag jobbat sen dess i olika roller. Mitt största fokus har ju egentligen varit information management, alltså informationshantering i systemen som kör, inte direkt mjukvaruutveckling som man kör här på Avensia. Så det är ju en stor skillnad mot hur jag tidigare har jobbat, som då kanske var lite mer mot administrativa verksamhetssystem.
3		CW	Och din arbetsroll idag, skulle du säga är...?
4		P5	Den är ju IT-projektledare och teamledare.

5		CW	Vad är din erfarenhet inom agila utvecklingsprojekt, både nuvarande och tidigare?
		P5	Det är ju egentligen ett projekt vi hade hos min tidigare arbetsgivare, där man också fick dela upp leveransen. Det blev ju mer delleranser än agil projektmetodik kanske. Och här på Avensia så strävar man ju efter att hålla en agil metodik, men jag tycker nog att det snarare är ett mer dellerans-tänk än agilt 100 %. Det finns viss teknisk bas som man måste sätta upp som man inte klarar sig utan, och när man väl har byggt upp den så kan man ta språnget att jobba agilt med kunden.
6	Fullt stöd från ledning	AO	Hur pass engagerad eller insatt anser du att VD:n eller platschefen är i ditt projekt?
7		P5	Litet. Det handlar mest om ekonomi, när vi gör fakturering så det kommer in pengar. Så mer än så känner inte jag att han är intresserad liksom. Kanske frågar ibland om hur det går, och så svarar man kort. Så det är på den nivån. Så det är ju mer att vår kund är en väldigt strategisk kund som Avensia tjänar bra med pengar på, alltså en bra kund och en bra referenskund också. Så jag skulle nog vilja säga att jag önskar att intresset vore större speciellt för en så stor kund.
8	Rätt utvecklare i rätt projekt	JB	Känner du att du har ett personligt intresse i det projektet du arbetar i?
9		P5	Absolut! Jag har som person ganska svårt för att gå in i saker som jag inte har passion för eller inte är personligt engagerad i.
10	Rätt utvecklare i rätt projekt	JB	Finns det några andra projekt här på Avensia som intresserar dig mer, som du hellre skulle arbetat i?
11		P5	Nej.
12	God insikt i arbetslaget	CW	Hur pass god insikt i varje projektmedlems kunskapsgrad och personlighet anser du att du har som projektledare?
13		P5	Nu när man har gått igenom projektet, eller kört projektet så pass länge, så förstår jag ju hur de olika personerna i projektet fungerar, och vilken nivå de ligger på rent tekniskt. Så jag tycker nog att jag har en god koll, men den skulle ju absolut kunna bli bättre genom att kanske formalisera det lite under projektet. Man skulle kanske kunna mäta utvecklingsnivån för varje individ; att när man börjar så typ "du befinner dig här: på nivå 3, på skala 1-5", och sen när projektet är slut så "tycker du att du har kommit upp i en annan nivå, eller befinner du dig på samma? fick du göra det du ville?". Så absolut, där tycker jag att vi kan bli mycket bättre. Jag har ju ett litet tänk i bakhuvudet, att man använder personer till det dem tycker om att göra, och att

			varje individ kan utvecklas under projektets gång. Men jag tycker att där kan vi bli mycket bättre, absolut.
14	God insikt i arbetslaget	CW	Har du några egna idéer på hur man skulle kunna göra det?
15		P5	Ja men kanske att man tillsammans gör typ en kompetensmatris, att det blir som en teamaktivitet där man kan vara ganska öppen med "jag är bra på det här, jag kan bidra med det här". Jag hade ett par intervjuer med nya personer som kom in under hösten, och då frågade jag ju "vad är du intresserad av? vad tycker du att du är bra på? så att jag placerar dig rätt". Så det är väl mer det som man kan formalisera lite mer, inför projektstart. Så att det kanske blir en teambuilding-aktivitet utav det också kanske. Så man också lär känna varandra, för annars ofta är det bara "du ska göra det, du ska göra det", och så kör. Men man har kanske ingen förståelse för varandra i teamet, vad man kan och inte kan. Så en kompetensmatris hade kanske inte varit fel.
16	Medvetet ökat självstyre	AO	Finner du att du tar egna initiativ, eller inväntar du delegation från dina chefer när det gäller projektet?
17		P5	Nej, jag tycker att jag tar mest egna initiativ. Sen får jag ju flagga upp om det uppstår problem och vi inte kommer vidare. Som exempelvis nu när projektet är avslutat så ska vi ju då uppsätta ett nytt avtal med kunden som är mer löpande räkningsarbete, ett SLA [Service-level agreement], och det har ju dragit ut på tiden. Så nu känner jag att jag måste liksom dra i handbromsen och flagga upp att "nu kommer vi inte vidare".
18	Medvetet ökat självstyre	AO	Du föredrar att ta egna initiativ då, eller?
19		P5	Ja, det föredrar jag. Men jag hade kanske önskat... men det beror ju också på belastningen och hur stor organisationen är också just nu. Man har inte den tiden att ägna, som man kanske hade behövt, som leveransansvarig. Då håller jag ju en dialog med honom som är leveransansvarig, och han försöker ju göra sitt. Men ska ju då också på visst sätt axla vår platschefs roll, som är pappaledig, eller vår tillfällige platschef som är också i Helsingborg. Så det finns ju liksom hela tiden orsaker till varför det ser ut som det gör. Men där tror jag också att man kan stärka som en projektledare genom att ställa krav.
20	Möten med relevans	JB	Hur ofta har ni projektmöten?
21		P5	Oj! För det första är det morgonmöten varje dag. Sen kanske man har lite större möten ungefär var tredje vecka. Så då brukar det finnas ett behov av att ha någon form av avstämning. Vi körde ett retrospektivt möte sist, och idag ska vi köra en genomgång av hela produkten. Jag upplever det som att det är en rätt rimlig nivå för att hålla ihop teamet, varannan eller var tredje vecka, att hålla större möten då.

22	Möten med relevans	JB	Hur tycker du att mötena är, sett till relevans och planering?
23		P5	Jag tycker väl kanske att de borde var ännu mer välplanerade och ha ett ännu tydligare syfte, att det har varit lite ad hoc. Man skulle absolut kunna ha planerat dem lite bättre, de stora mötena, för att få ut mer då. Som idag då, när vi ska gå igenom hela produkten: det kanske finns fler syften än att bara sitta och titta igenom buggar? Men just nu är det just det fokus eftersom vi befinner oss i test. Men man hade kanske kunnat ha möten där man diskuterar "hur skulle vi kunna sälja in mer till kunden?", "hur kan vi vidareutveckla vår produkt?". Så det finns ju absolut möjligheter att göra mer.
24	Samarbete mellan utvecklare och QA-personal	CW	Hur fungerar samarbetet enligt dig mellan utvecklarna i projektet och kvalitetssäkringspersonal?
25		P5	Ja... precis. Det är ju det som är ett litet bekymmer, att vi inte har kvalitetssäkringspersonal idag. Vi har ju utvecklare som kvalitetssäkrar, och det fungerar väl... okej ändå. Men det beror också mycket på att vi har fått in många externa konsulter som har jobbat mycket med just kvalitetssäkring, som har ställt krav på projektet och då har vi fått hantera det därifrån. Sen har vi ju också en kund som är van med att jobba i affärssystem, som enligt en viss process som påminner mer om vattenfallsmetod där man har en utvecklingsfas och sen en testfas, och sen gör man tester även i produktion. Där har vi med mindre kunder tidigare, där grejer har snabbare kommit ut i produktion kanske utan att behöva kvalitetssäkra så mycket. Ett mindre scope, mindre påverkan, och då går det snabbare också. Men jobbar man med större kunder, större system, och större påverkan, desto viktigare blir det att ha de här processerna med kvalitetssäkring. Så där har vi ju ett stort behov att göra mer.
26	Samarbete mellan utvecklare och QA-personal	CW	Anser du att det finns ett behov av att ta in kvalitetssäkringspersonal, som gör endast detta?
27		P5	Jag skulle nog vilja säga att det kanske handlar mer om testledning i första hand. Att försöka dela upp teamet, som vi har försökt göra i vårt nuvarande projekt, lite grann där vissa sitter med test. Så jag tror nog man hade kunnat använda utvecklare, men jag tror nog att man behöver någon i så fall som fungerar som en testledare, i synnerhet när det blir större när man har haft 10-15 utvecklare. För en projektledare är inte så detaljorienterad, och jag tycker lite att det är olika discipliner att jobba som projektledare eller som testledare, att man inte riktigt kan likställa de rollerna.

28	Samarbete mellan utvecklare och QA-personal	CW	Så du ser inte att det finns ett behov av kvalitetssäkringspersonal?
29		P5	Nej, jag tycker nog inte det. Men det kanske är för att jag är projektledare, hade jag varit testledare hade jag kanske gett ett annat svar. Men det är så jag ser det. Jag tror inte jag har jobbat direkt... eller jo, men det är ju på motsvarande på kundsidan, när kunden testar, men inte att det utförs så att säga tekniska tester av så kallade tekniska testare, utan jag har alltid haft scenariot där utvecklarna själv testar.
30	Samtlig kod testbar/Standardisering av kod	AO	Vilka eventuella riktlinjer följer ni för att skriva testbar kod? Har ni någon kodstandard?
31		P5	Nej. Det har vi ju inte. Så det är ju inte bra.
32	Formell kodgenomgång	AO	Sker det någon formell kodgenomgång?
33		P5	Nej. Det har vi också pratat om att ha så kallade code reviews, en miniprocess för det där man kvalitetssäkrar sin egen kod genom att någon annan alltid ska titta på koden. Men vi har inte kommit dit, det har inte blivit så. Men vi har absolut diskuterat det och har ambitionen, och jag tror att vi kommer försöka att göra det bättre under fortsatt utveckling. Att vi har lärt oss det och att det är viktigt, för det har varit ganska mycket slarv: tryckt in grejer för att det ska gå snabbt, och ja det ger ju dåligt resultat i slutändan.
34	Formell kodgenomgång	AO	Om ni skulle ha gjort en formell kodgenomgång; hur skulle den ha sett ut i så fall?
35		P5	Rent processmässigt, tänker du?
36	Formell kodgenomgång	AO	Ja exakt, hur det skulle gå till.
37		P5	Ja, men vi använder ju oss av Jira, som är projektadministrationsverktyget, egentligen internt, för oss på Avensia. Och där har man ju sitt användarfall som en utvecklare ska utveckla, och sen har man ju då satt den som "ready for test" som status, och då är ju tanken att innan man som ny utvecklare tar sig ann ett nytt användarfall så ska man hela tiden gå in och kolla: "är det någonting annat som någon annan har utvecklat som jag kan testa?". Och hittar utvecklaren då ett fall som är "ready for test" så måste jag testa det innan jag tar mig ann det nya användarfallet. Alltså utför en code review och testar och sätter statusen som "done" när det är färdigt. Först då kan utvecklaren gå vidare med sitt nya användarfall. Det samma gäller ju då att den som har utvecklat det första användarfallet, då är det samma sak för den att den tittar vidare om någon annan i teamet har gjort något annat användarfall som är "ready for test". Så vill vi

			jobba, men vi har inte riktigt lyckats med det i projektet än. Men förutsättningarna finns ju där i Jira.
38	Formell kodgenomgång	CW	Ser du att det finns någon speciell anledning till varför det inte har fungerat?
39		P5	Jag tror det är disciplin faktiskt. Jag tror folk saknar disciplin, och det gäller att någon har stenkoll på Jira. Och nu har vi gått från Jira lite och jobbar istället i TFS:n mot kunden, som ställer sina krav där genom det verktyget istället. Och där har vi ju inte alls samma förutsättningar för att följa upp projektuppgifter så som vi kan i Jira. Så där är en liten motsägelse mot hur vi vill jobba, och hur Jira funkar på Avensia, för det gillar vi ju. Medan kunden då ställer krav på hur vi ska jobba, och förminska våra möjligheter då. Så där får vi ju försöka hitta någon slags process för det sen.
40	Formell kodgenomgång	CW	Sker det ofta, tror du, att kunden säger att man ska arbeta på ett visst sätt? Även ifall ni påpekar att det kan påverka er?
41		P5	Jag tror inte det, men när det gäller den typen av större kund så har inte dem bara då vårt projekt i sin TFS, utan de har ju alla sina system där. Och skulle vi då börja jobba på något annat sätt så blir det omständligt för kunden. Så de ställer ju krav på sina underleverantörer att "ni måste rätta er efter oss".
42	Rätt mängd dokumentation	JB	Hur ser dokumentationen ut i ert projekt?
43		P5	Ja, den är väl blandad, skulle jag vilja säga. Men där har vi också egentligen inte jobbat så strukturerat med dokumentation. Vi har ju egentligen inte bestämt från början vilken typ av dokument som vi ska skapa. Och vi har satt 120 timmar, tror jag, som ett estimat för att göra dokumentation. Men vi har egentligen inte sagt till kunden att "det är de här dokumenten ni får", utan det är ju typ: hur ser miljön ut, vilken miljö har vi, hur ser de olika lösningarna ut. Och då ligger vi in detta i Confluence. Men det är ju väldigt ostrukturerat faktiskt. Så det är kanske också någonting att tänka på när man startar upp projektet, att faktiskt bestämma vilken typ av dokument vi faktiskt ska skapa, utifrån användarfallen. Man pratar ju om "epics", som egentligen är kategorier av användarfall. "Startsidan" till exempel, "produkt-sida", "checkout". Där skulle man kanske ha sagt att man skapar dokument för varje epics, och för varje är det en viss dokumentation med en viss nivå. Så det blir formellt, och det är det som man lämnar över till kunden.
44	Rätt mängd dokumentation	JB	Anser du att dokumentationen ni gör idag är tillräcklig?
45		P5	Nej. Nej, det tycker jag inte. Det skulle behövas mer... inte så detaljerad teknisk dokumentation, alltså om koden typ, utan då

			handlar det om att dokumentera flöden och processer på sidan. Hur det funkar, alltså på en högre nivå. En lösningsarkitekt, tänker jag, borde sätta sig och gå igenom det och dokumentera.
46	Bryt ned stora projektuppgifter till mindre delar	CW	Större projektuppgifter; hur angriper ni dem i projektet? Om det exempelvis kommer in önskemål på väldigt stora funktioner, eller annat.
47		P5	Ja, då har vi ju en "change management"-process... som är väl inte heller helt formaliserad, men där har ju kunden kommit in med väldigt stora grejer ganska ad hoc, exempelvis "ah, nu har inte en leverantör gjort vår huvudsida, så nu vill vi att ni också gör den". Och från början skulle vi ju bara göra deras e-handelsida, som är väldigt smal. Så då blir det ju liksom att gå in till leveransansvarig som kan se till att få fram resurser och backa upp en så stor förändring. Och så har det ju varit under hela vägen, och det är bara ett exempel.
48	Bryt ned stora projektuppgifter till mindre delar	CW	Finns det något annat tillvägagångssätt som du hade föredragit?
49		P5	Nja, jag vet inte. Det är svårt. Det är väl att vara kanske lite mer förutsägbar från kunden då, och säga att de måste flagga upp en change request av förändringar av teamet, alltså som påverkar teamet hos oss, och säga att det kommer påverka tidsplanen tydligt. För här har vi ju hela tiden jobbat på att hålla tidsplanen, oavsett vad vi kastar in liksom så ska tidsplanen vara den samma. Kanske ha lite framförhållning: kommer ni senare än en månad med en change request som påverkar projektets tidsplan, så kommer det påverka slutresultatet. Så där kan man också göra lite förbättringar... vara tydlig! Vara tydlig mot kunden.
50	Tydliga användarfall	AO	Vem formulerar era användarfall?
51		P5	Det gör ju den personen under förstudien som har rollen som lösningsarkitekt. Ofta så är ju det en senior utvecklare, som kan alla de system vi arbetar i, och som har ett brinnande intresse för e-handel. Så då försöker man hitta någon som skapar de användarfallen. Det blir den som är huvudansvarig för dokumenten, sen är det ju alltid kanske några inspel av andra utvecklare, men det är ju i första hand lösningsarkitekten, och i andra hand frontend- eller backendutvecklare.
52	Tydliga användarfall	AO	De användarfall ni har; är de användbara?
53		P5	Till viss del. Kanske för syftet att paketera en förstudie och skapa en bild av vad det är kunden ska få i slutändan. Men för en utvecklare tror jag ju inte att dem är tillräckliga. Där krävs ju ytterligare en, eller två, eller tre nivåer av detaljering och specificering av vad det är som faktiskt ska hända i systemet. Så det är ju lite olika syften. Hos oss på Avensia så förutsätter man ju

			att användarfallen bara ska gå att plocka och utveckla direkt, men så är det aldrig. Så där finns också ett stort behov... och även att vara tydlig mot kunden: vad är det vi gör nu i förstudien, och vad kommer vi behöva jobba igenom ytterligare under projektet.
54	Tydliga användarfall	CW	Känner du dig bekväm med att skriva användarfall?
55		P5	Nej.
56	Tydliga användarfall	AO	Men du förstår poängen med dem?
57			Absolut, ja det gör jag ju. Men jag tror ju också det att jag inte har jobbat med e-handel på den detaljnivån, jag har inte jobbat med e-handel innan jag kom hit. Så jag förstår ju användarfallen, absolut, men det skulle nog behöva något projekt till för att jag skulle kunna skriva egna.
58	Kundinvolvering	JB	Hur involverad är kunden i era projekt?
59		P5	Jag skulle säga att de är väldigt involverade. Det krävs att de är involverade. Men sen fungerar ju olika kunder på olika sätt. Vår nuvarande kund tycker jag att vi har fått väldigt mycket frihet och möjlighet att ta mycket eget ansvar. Vi har ju legat före dem, vilket är ju kanske naturligt för detta är ju helt nytt för dem. Men tidigare hade jag ett projekt där kunden var ännu skarpare på vissa delar av projektet, där till exempel en person som var e-handelschef som va expert på Google Analytics och olika verktyg, och hade järnkoll på tekniken. Och då blir ju det ett helt annat sätt att möta den kunden, som är extremt detaljkuserad och orienterad, än istället som vår nuvarande kund där vi på något sätt måste styra dem och ligga steget före hela tiden. Så det kan ju vara väldigt stor skillnad, men det krävs ju absolut ett engagemang.
60	Kundinvolvering	JB	Anser du att kundinvolveringen borde öka eller minska?
61		P5	Tänker ni på vår nuvarande kund, eller generellt?
62	Kundinvolvering	JB	Generellt.
63		P5	Någonstans så tror jag det är viktigt att lyssna på kunderna. Vi på Avensia ska ju va såna sjukt skarpa experter när det gäller e-handel, och vilka funktioner ska vi erbjuda och föreslå, men jag tror många gånger så borde vi bara hålla en dialog med kunden. Där vi ser kunden som en idéspruta också, hur vill de att sidan ska se ut? Vilka funktioner behöver de? Men jag tycker nog att involveringen borde öka ännu mer, och hjälpa oss att bli bättre. För så är det ju när man jobbar med SAP mycket, då har de många användarorganisationer som träffas en gång i halvåret eller kvartalet där man tar input från kunderna, för att vidareutveckla systemet. Så ser jag det också som att Avensia borde

			också jobba med kunderna: ta in dem och hålla ett forum för idéer för nya funktioner som kan behövas, och hur bör vi vidareutveckla våra system utifrån det.
64	Kundinvolvering	JB	Du känner inte att det kan finnas ett scenario där kunden blir för involverad?
65		P5	Jo, men kanske då med exempel med den kunden vi hade, där kunden tog sig själv på så stort allvar och så extremt teknik- och detaljfokuserade. Där blev det lite för mycket, för kunden tyckte sig veta bäst. Men det handlade just om att den kunden hade svårt att ta en dialog, så det handlar ju mer om personlighet och kanske inte just om kunden utan istället just den personen. Men då va det en person i den positionen som e-handelschef, så då blev det lite... ja.
66	Öppen, samt strukturerad, arbetsplats	CW	Hur ser ert kontorslandskap ut, och hur tycker du att det fungerar?
67		P5	Vi har ett öppet kontorslandskap, som är alldeles för trångt. Så just nu fungerar det inte så bra, men det kommer ju bli bättre eftersom vi har två bolag som kommer flytta så där kommer det ges nya platser och vi kan sprida ut oss. På ett sätt är det väldigt skönt med landskap, att man har tillgång till alla när som helst, och jag tror för ett team att sitta och utveckla tillsammans är det en viktig del för att hålla dialogen vid liv. Det är lätt att bara sträcka sig över skrivbordet så finns personen där. Men för mig som projektledare så hade jag gärna... jag går ju undan ibland och stänger in mig för att kunna fokusera. Det är ju skillnad kanske att sitta och utveckla mot att sitta och skriva rapporter och hålla dialog med kund. Men jag föredrar öppna landskap, där man har möjlighet att gå undan om man behöver det, än tvärtom. Efter sommaren kommer det nog bli optimalt här.
68	Öppen, samt strukturerad, arbetsplats	CW	Hur skulle du säga att det påverkade ditt dagliga arbete med det öppna landskapet?
69		P5	Jag tror det bidrar positivt. Att man har alla omkring sig, jag gillar det. Man är med, och sitter inte i ett rum och ugglar liksom. Alla är med!
70		AO	Innan vi avslutar; är det någonting du vill tillägga?
71		P5	Nej, inte vad jag kan komma på.
72		AO	Då får vi tacka för din tid!

8 Referenser

- Austin, R. and L. Devin (2009) "Weighing the Benefits and Costs of Flexibility in Making Software: Toward a Contingency Theory of the Determinants of Development Process Design, *Information Systems Research* (20)3, pp. 462–477.
- Barlow, J. B., Giboney, J., Keith, M. J., Wilson, D., Schuetzler, R., Lowry, P. B., & Vance, A. (2011). Overview and guidance on agile development in large organizations. *Communications of the Association for Information Systems*,29(2), 25-44.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). *Manifesto for agile software development*.
- Berg, B. L., Lune, H., & Lune, H. (2004). *Qualitative research methods for the social sciences* (Vol. 5). Boston, MA: Pearson.
- Braa, K., & Øgrim, L. (1995). Critical view of the application of the ISO standard for quality assurance. *Information Systems Journal*, 5(4), 253-269.
- Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems*,18(4), 332-343.
- Cho, J. (2008). Issues and Challenges of agile software development with SCRUM. *Issues in Information Systems*, 9(2), 188-195.
- Cho, J., Huff, R., & Olsen, D. (2011). Management guidelines for scrum agile software development process. *Issues in Information Systems*, 12(1), 213-223.
- Cockburn, A., & Williams, L. (2003). Agile software development: it's about feedback and change. *Computer*, 36(6), 0039-43.
- Denscombe, M. (2000): *Forskningshandboken – för småskaliga forskningsprojekt inom samhällsvetenskaperna*. Studentlitteratur, Lund.
- 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*, 85(6), 1213-1221.
- Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*, 15(2), 200-213.
- Hannay, J. E., Dybå, T., Arisholm, E., & Sjøberg, D. I. (2009). The effectiveness of pair programming: A meta-analysis. *Information and Software Technology*, 51(7), 1110-1122.
- Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120-127.
- Hoda, R., Noble, J., & Marshall, S. (2011). The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology*, 53(5), 521-534.

- Holme, I., Solvang, B. (1997): *Forskningsmetodik*. ISBN: 9144002114. 360 s.
- Holmström, H., Fitzgerald, B., Ågerfalk, P. J., & Conchúir, E. Ó. (2006). Agile practices reduce distance in global software development. *Information Systems Management*, 23(3), 7-18.
- Huo, M., Verner, J., Zhu, L., & Babar, M. A. (2004, September). Software quality and agile methods. In *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International* (pp. 520-525). IEEE.
- Iivari, J., & Iivari, N. (2011). The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology*, 53(5), 509-520.
- ISO. (2005). *ISO 9000:2005 – Quality management systems – Fundamentals and vocabulary*. International Standardization Organisation.
- Jacobsen, D. (2002): *Vad, hur och varför*. ISBN: 9144040962. 503 s.
- Kautz, K., Johanson, T. H., & Uldahl, A. (2014). The perceived impact of the agile development and project management method scrum on information systems and software development productivity. *Australasian Journal of Information Systems*, 18(3).
- Lantz, A. (2013). *Intervjumetodik*. ISBN: 9789144081236. 173 s.
- Leffingwell, D. (2007). *Scaling software agility: best practices for large enterprises*. Pearson Education.
- Lindstrom, L., & Jeffries, R. (2004). Extreme programming and agile software development methodologies. *Information systems management*, 21(3), 41-52.
- Lyytinen, K., & Rose, G. M. (2006). Information system development agility as organizational learning. *European Journal of Information Systems*, 15(2), 183-199.
- Mangalaraj, G., Mahapatra, R., & Nerur, S. (2009). Acceptance of software process innovations—the case of extreme programming. *European Journal of Information Systems*, 18(4), 344-354.
- Mathiassen, L., & Pries-Heje, J. (2006). Business agility and diffusion of information technology. *European Journal of Information Systems*, 15(2), 116.
- Melnik, G., & Maurer, F. (2004, August). Introducing agile methods: three years of experience. In *Euromicro Conference, 2004. Proceedings. 30th* (pp. 334-341). IEEE.
- Molokken-Ostfold, K., & Haugen, N. C. (2007, April). Combining estimates with planning poker—an empirical study. In *Software Engineering Conference, 2007. ASWEC 2007. 18th Australian* (pp. 349-358). IEEE.
- Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and organization*, 17(1), 2-26.
- Nabulsi, M. (2014). *Towards more effective testing of communications-critical large scale systems* (Doctoral dissertation, Brunel University, School of Information Systems, Computing and Mathematics).
- Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3), 303-337.

-
- Rising, L. and N.S. Janoff (2000) "The Scrum Software Development Process for Small Teams, *IEEE Software* (17)4, pp. 26–32.
- Royce, W. W. (1970). *Managing the development of large software systems*. In proceedings of IEEE WESCON (Vol. 26, No. 8).
- Sarker, S., C. Munson, and S. Chakraborty (2009) "Assessing the Relative Contribution of the Facets of Agility to Distributed Systems Development Success: An Analytic Hierarchy Process Approach, *European Journal of Information Systems* (18)4, pp. 285–299.
- Schach, S. R. (2004). *An introduction to object-oriented systems analysis and design with UML and the unified process*. McGraw-Hill/Irwin.
- Schümmer, T., Lukosch, S., & Slagter, R. (2006). Using patterns to empower end-users—The oregon software development process for groupware. *International Journal of Cooperative Information Systems*, 15(02), 259-288.
- Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum*, Upper Saddle River, NJ: Prentice Hall.
- Sumrell, M. (2007, August). *From Waterfall to Agile – How does a QA Team Transition?* In Agile Conference (AGILE), 2007 (pp. 291-295). IEEE.
- Willson, C. D. (2009). A brief introduction to SCRUM: An agile methodology. *Matincor Inc. Information Technology Management Consulting*.