



# LUNDS UNIVERSITET

## Ekonomihögskolan

*Institutionen för informatik*

---

# Systemdokumentation

En studie om inställning till och hantering av systemdokumentation

Kandidatuppsats 15 hp, kurs SYSK02 i informationssystem

Författare: Fredrik Alserud  
Marcus Nilsson

Handledare: Odd Steen

Examinatorer: Agneta Olerup  
Mirella Muhic

# Systemdokumentation: En studie om inställning till och hantering av systemdokumentation

Författare: Fredrik Alserud och Marcus Nilsson

Utgivare: Inst. för informatik, Ekonomihögskolan, Lund universitet

Dokumenttyp: Kandidatuppsats

Antal sidor: 63

Nyckelord: systemdokumentation, mjukvaruutveckling, inställning, hantering, problem

Sammanfattning (Max. 200 ord):

Att det finns problem med dokumentation, inklusive systemdokumentation är ett känt faktum inom IT-världen. Den här uppsatsen har identifierat några huvudproblem, som att dokumentation inte hålls uppdaterad, att den inte prioriteras och att mjukvaruutvecklare har en negativ inställning till systemdokumentation för att sedan undersöka hur företag ställer sig till dessa problem och hur de hanterar dem. I undersökningen intervjuades två mjukvaruproducerande företag av olika storlek kring hur de arbetar med dessa problem, finns det till exempel några riktlinjer inom organisationen för systemdokumentation eller tycker de inte att problemen är värda att hantera överhuvudtaget. Efter intervjuerna genomfördes en analys av där den insamlade empirin. Uppsatsen fann att de som intervjuades höll med om majoriteten av de identifierade problemen, men ansåg att de inte var värda att lägga för mycket resurser på att hantera. Detta styrktes av att det fanns ytterst få riktlinjer kring just systemdokumentation i båda företagen. Den största anledningen som den här uppsatsen fann till att hantering av problemen inte prioriteras var att utvecklarna har en negativ inställning gentemot systemdokumentation och därför tas det inte upp som ett problem i en organisation som prioriterar att producera värde till sina kunder.

## Innehåll

1	Introduktion.....	1
1.1	Problemområde.....	2
1.2	Forskningsfråga .....	2
1.3	Syfte.....	2
1.4	Avgränsningar .....	2
2	Litteraturgenomgång.....	3
2.1	Vad är dokumentation och systemdokumentation?.....	3
2.2	Utvecklingsmetoders synsätt på dokumentation .....	4
2.2.1	Traditionell utveckling .....	4
2.2.2	Agil dokumentation.....	5
2.3	Systemdokumentation .....	7
2.3.1	Inline & Extern dokumentation.....	7
2.3.2	Dokumentationsverktyg och hjälpmedel.....	8
2.3.3	Användning av systemdokumentation .....	8
2.3.4	Problem kring systemdokumentation.....	9
2.3.5	Konsekvenser av dessa problem .....	10
2.3.6	Software Crisis och dokumentation .....	11
2.4	Sammanfattning av litteratur .....	11
3	Metod.....	13
3.1	Kvalitativ undersökningsmetod.....	13
3.2	Val av organisation och intervjupersoner.....	13
3.3	Intervjuer .....	14
3.3.1	Intervjuguide .....	14
3.4	Analys av intervjuer.....	14
3.4.1	Huvudkategorier.....	15
3.5	Undersökningskvalitet .....	16
3.5.1	Validitet.....	16
3.5.2	Reliabilitet .....	16
3.6	Etik.....	17
4	Resultat .....	18
4.1	Studiens undersökningsobjekt .....	18
4.1.1	Företag A.....	18
4.1.2	Företag B .....	18

---

4.2	Individens inställning till dokumentation .....	19
4.2.1	Användning .....	19
4.3	Organisationsriktlinjer och hantering av dokumentation .....	20
4.3.1	Storlek .....	20
4.3.2	Arbetsmetod & Riktlinjer .....	21
4.3.3	Stöd från ledning .....	21
4.4	Hantering av problem .....	22
4.4.1	Negativ inställning .....	23
4.4.2	Intressekonflikt & Prioriteringsfråga .....	23
4.4.3	Utdaterad dokumentation .....	23
4.4.4	Hantering av problem .....	24
4.5	Sammanfattning .....	26
5	Diskussion .....	27
5.1	Individens inställning till dokumentation .....	27
5.1.1	Användning av Inline-dokument .....	28
5.2	Organisationsriktlinjer och hantering av dokumentation .....	28
5.2.1	Hantering varierar efter storlek .....	28
5.2.2	Brist på riktlinjer .....	28
5.2.3	Utvecklingsprocessens påverkan .....	29
5.2.4	Kunder får ledningen att agera .....	30
5.3	Hantering av problem .....	30
5.3.1	Prioriterings- & Intressekonflikt .....	30
5.3.2	Utdaterad dokumentation .....	31
5.3.3	Negativ inställning .....	31
5.4	Sammanfattning .....	32
6	Slutsats .....	33
	Bilaga 1 – Intervjuguide .....	35
	Bilaga 2 – Intervjuperson A1 .....	36
	Bilaga 3 – Intervjuperson A2 .....	41
	Bilaga 4 – Intervjuperson B1 .....	49
	Bilaga 5 – Intervjuperson B2 .....	55
	Referenser .....	62

## Figurer

Figur 2.1 – Waterfall Model (SoftwareTestingHelp, 2016).....	5
Figur 2.2 – Agile document Late SDLC (Ambler, 2012). ....	6
Figur 2.3 – Agile Document Continuously SDLC, Ambler (2012) .....	6

## Tabeller

Tabell 4.1 – Resultat i undersökningsmodell angående individens inställning. ....	20
Tabell 4.2 – Resultat i undersökningsmodellen för en organisations riktlinjer.....	22
Tabell 4.3 – Resultat i undersökningsmodell för hantering av kända problem.....	25



# 1 Introduktion

Dokumentation anses vara en viktig del av utvecklingen av ett system och dess fortsättning. I grund och botten är det en typ av kommunikation, vars mål är att stödja projektets utvecklare och övriga intressenter (Ambler, 2012). Skapandet av dokumentation kan ske av varierande anledningar. Det kan skapas för att underlätta kommunikation mellan externa grupper och deltagare, då det inte är säkert att alla deltagare i ett projekt kan utföra sitt arbete samtidigt på samma arbetsplats, eller för att stödja underhållet av ett system, likt ett organisatoriskt minne. Dock kan det förekomma dokumentation som skapas för att det finns formella krav, men utan något precist syfte med vad eller vem dokumentationen är till för (Ambler, 2012).

Om dokumentationen sker på ett korrekt och effektivt sätt, kommer det att minska utvecklingstiden och resultera i ett mer effektivt arbete. Dokumentation är ett viktigt element för att skapa en mjukvara och hålla mjukvaran fungerande och anpassningsbar (Raskin, 2005). I uppsatsen *Användningen och nyttan av systemdokumentation* (Magnusson, 2004) menar undersökningens intervjupersoner att dokumentation är till hjälp när de ska sätta sig in i ett nytt system, samt om dokumentationen är övergripande. Mer detaljerad information vill de ha i inline-dokumentation, direkt i källkoden (Magnusson, 2004).

Mjukvaruexperter från Prause & Durdiks studie (2012) upplever dock att det existerar problem med dokumentationen och trots att dokumentation anses vara viktig, finns det en kritisk inställning till den hos mjukvaruutvecklare. Dokumentationsaktiviteterna anses inte vara nödvändiga, då de enbart har en indirekt påverkan på projektets slutprodukt. Ytterligare anledningar till att det finns en kritisk inställning till dokumentationen kan vara tidsbrist hos utvecklarna, otydliga mål med dokumentationen, samt brist på kunskap (Prause & Durdik, 2012). Även om det finns lösningar som anses vara effektiva, kan problemet med att det brister i dokumentationen ha en kulturell aspekt; att utföra dokumentering parallellt med utvecklandet ligger inte i linje med den kultur som existerar i organisationen (Raskin, 2005). Raskin (2005) menar även att problemet kan ligga i utbildning, då lärare sällan har med dokumentation i betygskalan i en programmeringskurs.

Dock kan problemen vara värre än att oklarheter skapas, för att en person inte kan hitta korrekt information. Bristfälliga dokumentationsprocesser och dokumentation kan få större konsekvenser än vad en må tro. I en undersökning av International Data Corporation (IDC) upplever 75,9% av företagen seriösa affärsrisker eller varit med om allvarliga konsekvenser på grund av bristfälliga dokumentationsvanor (Boyd, Pucciarelli, Webster, 2012). Konsekvenserna som nämns är misslyckande avseende efterlevnadskrav, PR-kris och förlust av viktiga kunder, konsekvenser vilka kan bli betydligt mer kostsamma än investeringen i förbättrad dokumentation. Nämnvärt är att detta gäller dokumentationsprocesser överlag, inte enbart relaterat till systemdokumentation (Boyd m.fl., 2012).



För att få en förändring till stånd krävs det att chefer högre upp i hierarkin måste investera resurser och sitt egna intresse i frågan (Boyd, m. fl., 2012). Dock kan detta vara svårt. I en undersökning visar resultatet att ju mer erfarenhet, desto mindre problem såg personerna med dokumentation vid ett projekt. Detta skulle kunna förklaras med att personerna med mer erfarenhet arbetade längre ifrån implementationen och därför inte kände av brister i dokumentationen i samma utsträckning (Prause & Durdik, 2012).

## 1.1 Problemområde

Dokumentationen anses som nämnt ovan, vara en viktig del av ett projekt, och värdet av den avgörs av hur korrekt och relevant information den sparar på ett effektivt sätt. Att underhålla den kräver i sin tur tid och resurser (Correia m.fl, 2009), men det är ingen hemlighet att det finns problem med dokumentation. Vanliga problem med dokumentationen är den inte blir uppdaterad tillräckligt ofta, vilket leder till att den inte kan stödja utvecklarna, samt att den i vissa fall inte ens existerar. Dokumentationen kan även vara svårtillgänglig i form av att den är utspridd och varierar i format, eller att det finns för mycket dokumentation, något som försvårar uppdateringar och är tidskrävande. Det talas även om en negativ inställning kring dokumentation hos utvecklare, vilket är en bidragande faktor till att dokumentationen uteblir. (Anquetil m.fl, 2005; Prause & Durdik, 2012.; Lethbridge m.fl, 2003).

## 1.2 Forskningsfråga

Hur ser en IT-organisations hantering och inställning ut beträffande systemdokumentation i relation till problem med systemdokumentation?

## 1.3 Syfte

Syftet med uppsatsen är att identifiera problem med systemdokumentation via den befintliga litteraturen på området och sedan utföra en undersökning, genom att utföra intervjuer med mjukvaruutvecklande organisationer. Ämnet anses relevant då har skrivits om problemen inom området, exempelvis Prause & Durdik (2012) och Sulaiman m.fl (2002), men har inte diskuterats om organisationer gör något i reaktion till detta.

## 1.4 Avgränsningar

Studien kommer att avgränsa sig till systemdokumentation och fokusera på att behandla den typen av dokumentation som stödjer utvecklare vid skapande och underhåll av system. Studien fokuserar på att se hur en organisation behandlar systemdokumentation och kommer inte bidra med lösningsförslag hur organisationer bör agera för att uppnå en kvalitativ dokumentation, eller vad som är den mest optimala dokumentationslösningen.

## 2 Litteraturgenomgång

*I detta kapitel ges definitioner av vad dokumentation innefattar och dess roll inom informatik och mjukvaruutveckling. Kapitlet tar även upp faktorer som påverkar utformningen av systemdokumentation, åsikter från tidigare studier i ämnet, samt vilka problem som kan existera.*

### 2.1 Vad är dokumentation och systemdokumentation?

Uppsatsen bygger sin definition av dokumentation och systemdokumentation på boken “Introduction to Software Engineering” av Ian Sommerville (2011). En mjukvara är inte bara själva programmet, utan också all tillhörande dokumentation. För utan dokumentationen går det inte att använda programmet fullt ut, hävdar Sommerville (2011) och fortsätter med att säga att skrivandet av dokumentation är en av de stora skillnaderna mellan en professionell och en amatörmjukvaruutvecklare. En amatörs program ska kanske inte användas av någon annan än denne själv och därför behövs inte struktur- eller användardokumentation skrivas. Vid professionell mjukvaruutveckling är det däremot flera utvecklare som arbetar på samma program och de behöver således ytterligare information över själva koden för att kunna bygga vidare på programmet. Dokumentation är ett samlingsord för flera typer av dokument och kan delas upp i två kategorier; processdokumentation och produktdokumentation (Sommerville, 2011).

Processdokumentation behandlar alla dokument som skapas och behandlar projektet under processens gång. Denna typ av dokumentation är till för att stödja deltagarna i projektet, samt ett effektivt sätt att kunna styra och leda ett projekt. Vanligt är att processdokumentation inte fyller någon funktion efter projektets slut, då dessa dokument inte längre är aktuella och ej uppdaterade (Sommerville, 2011). Sammanfattat innehåller processdokumentation följande typer av dokument (Sommerville, 2011):

- Planer och scheman - för att ha en planering att utgå efter.
- Rapporteringar - hur resurser har används under en viss process.
- Standarder - hur processer ska gå till.
- Arbetsfiler - sparade idéer, tankar och identifierade problem i dokument, som skapas under projektets gång.
- E-mail, wikis, etc - sparade konversationer mellan projektdeltagare

Produktdokumentation behandlar dokumentation angående produkten, vilket är, som nämnts ovan, en viktig del av leveransen till kunden. Denna dokumentation är i sin tur uppdelad i två delar; användardokumentation och systemdokumentation (Sommerville, 2010). Användardokumentation kan beskrivas som hjälpinformation, likt manualer till de som ska använda produkten. Det finns variationer beroende på till vem dokumentet riktar sig, installationsguider till systemadministratörer eller guider hur användaren ska gå tillväga vid errormeddelanden (Sommerville, 2011).

Systemdokumentation är den andra kategorin av produktdokumentation, och är vad denna studie kommer att fokusera på. Denna typ av dokumentation innefattar beskrivningar och viktig information för att förstå systemet, både i helhet men även på en mer detaljerad nivå. Målet med systemdokumentationen är att stödja organisationen med att underhålla sitt system, samt göra det lättare för utvecklare att vidareutveckla systemet (Sommerville, 2011). Systemdokumentation kan innehålla följande sorters dokument (Sommerville 2011):

- Beskrivning av systemets arkitektur.
- Beskrivning av arkitekturen för varje program inom ett system.
- Förklaring av hur varje komponent inom systemet fungerar.
- Kodkommentarer.
- Dokument angående kvalitetssäkring.
- Beskrivningar av hur systemet bör underhållas och vad som ska göras vid vissa kända problem.

Inom systemdokumentationen finns inline- och extern dokumentering. Inline-dokumentering är det som ses som koddokumentering, vilket kan ske på olika sätt. Grunden är att dokumentationen ligger i källkoden. Det går att skriva längre kommentarer och förklaringar i källkoden, medan ibland kan det räcka med en "självdokumenterande" kod. Det sistnämnda betyder exempelvis att ett metodnamn är tillräckligt tydligt för att ge en förståelse över vad den utför (Ambler, 2012).

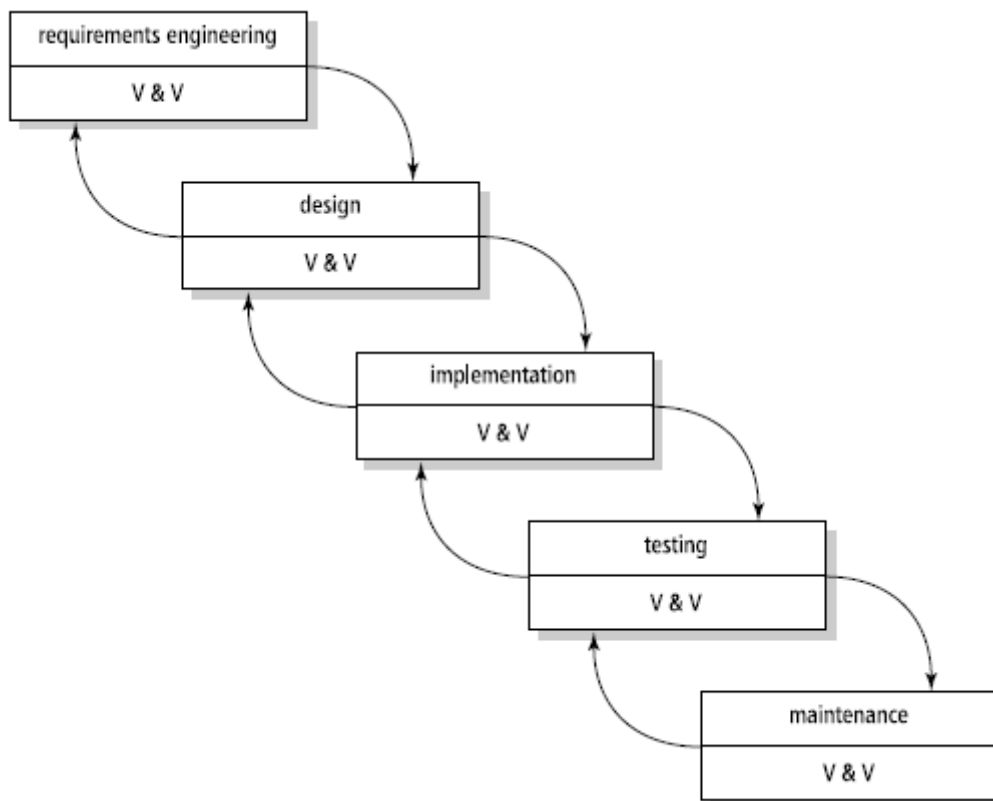
Den externa dokumenteringen är riktad till en bredare målgrupp än bara utvecklare. Alla i projektet har varken tillgång till eller förståelse för att läsa vad som står i källkoden (Ambler, 2012). Extern systemdokumentering innebär alltså dokumentfiler separat sparade, och kan innehålla text och modeller, som förklarar och beskriver systemet (Rüping, 2003).

## 2.2 Utvecklingsmetoders synsätt på dokumentation

### 2.2.1 Traditionell utveckling

Vattenfallsmodellen är en av de tidigaste utvecklingsmetoderna, och ses som en grund till andra mer effektiva modeller. Denna modell illustreras nedan i figur 2.1 och den består av ett antal separata steg eller faser, som ligger sekventiellt. När ett steg är helt färdigt och klarat en slutgiltig genomgång, kan projektet gå vidare in till nästa fas. Denna modell är kritiserad bland annat för att vara oflexibel och det är kostsamt att behöva gå tillbaka till en tidigare fas (Görling, 2009; McConnell, 1996).

I denna utvecklingsmetod finns det som kan kallas det traditionella synsättet på dokumentation. Vattenfallsmodellen är, enligt McConnell, S. (1996), "dokumentationsdriven". Att skapa dokumentation är själva huvudprodukten i varje fas, och dessa dokument med sin information blir grunden som nästa fas ska bygga vidare på. Dokumentation under projektets gång kan också jämföras med den första dokumentationen för att ge indikatorer hur projektet går (McConell, S., 1996).

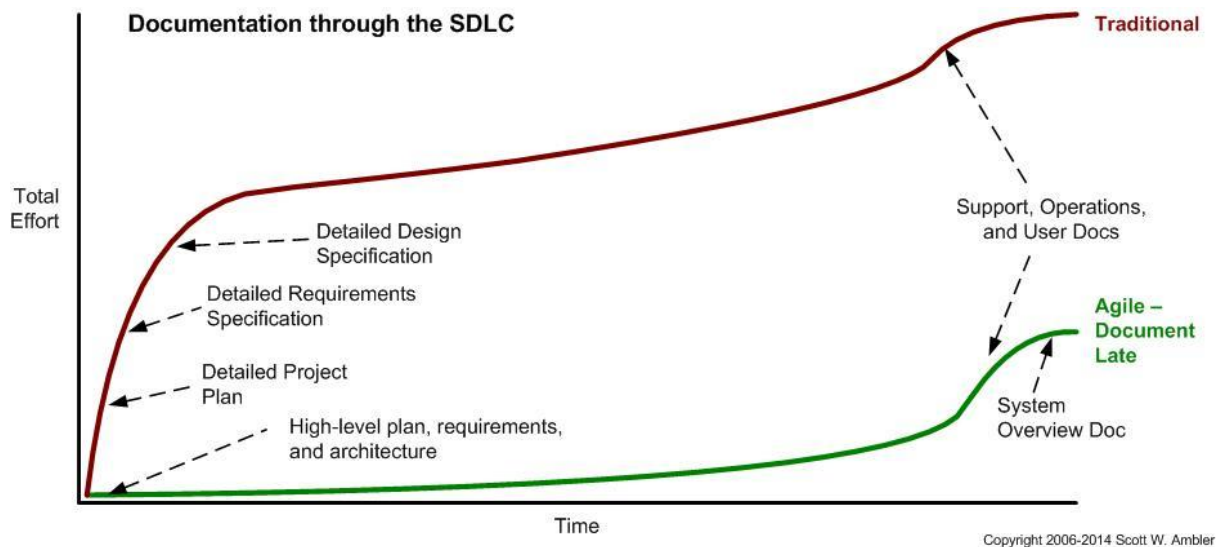


Figur 2.1 – Waterfall Model (van Vliet, 2007)

### 2.2.2 Agil dokumentation

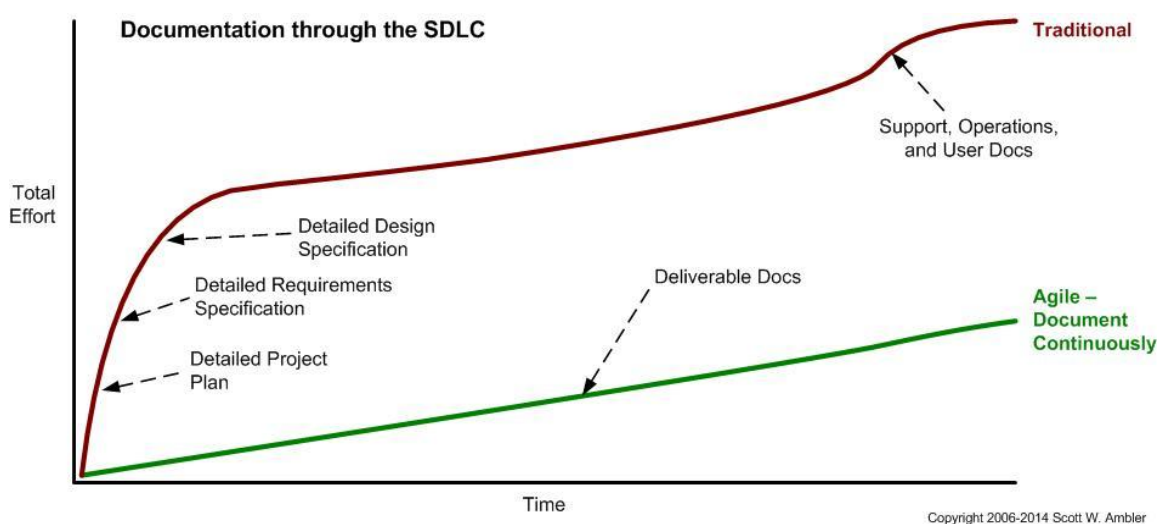
Agil dokumentation är ett synsätt som fokuserar på att dokumentation ska ta så lite ansträngning som möjligt att skriva. Detta illustreras i figur 2.2 och 2.3 där två tillvägagångssätt för att skriva agil dokumentation jämförs med den traditionella vattenfallsmodellen. En av de stora skillnaderna är att agil dokumentation ska vara precis bra nog för att vara förståelig. Allt som skrivs efter det stadiet är ett slöseri med tid och resurser. Det vill säga att fördelen med att skriva dokumentation måste vara större än kostnaden att skriva och underhålla det (Ambler, 2012). En annan förespråkare av agil dokumentation, menar att även om en organisation skulle förlora den person som har all kunskap, hade inte massvis med dokumentation räddat situationen (Highsmith, 2002). Skrivs det för mycket dokumentation tenderar utvecklare att mistro den dokumentationen eftersom den ofta är osynkroniserad med koden i mjukvaran. Agil dokumentation fokuserar på att organisationer frågar sig själva innan de dokumenterar ifall dokumentationen verkligen behövs, eller om det endast är något som de bestämt sig för att ha (Ambler, 2012).

När dokumentation skrivs agilt presenterar Ambler (2012) två tillvägagångssätt. Det första är att skriva dokumentationen så sent som möjligt i livscykeln för mjukvaruutveckling (SDLC) (se figur 2.2). Att skriva dokumentation i slutet beror på att det inte ska läggas någon tid på att skriva dokumentation om osäkra aspekter (Ambler, 2012). Aspekter som är osäkra kan senare tas bort, vilket gör den tid och de resurser som lades på att skriva dokumentation om den aspekten till ett slöseri. Just systemdokumentation är det som helst ska skrivas allra sist under livscykeln.



Figur 2.2 – Agile document Late SDLC (Ambler, 2012).

Det andra tillvägagångssättet är att skriva dokumentation kontinuerligt under livscykelns gång, se figur 2.3. Det här är tänkt att vara applicerbart på iterativa projekt. Mer specifikt iterativa projekt där det är tänkt att mjukvaran, och därmed all tillhörande dokumentation, ska vara redo att skickas till kund vid varje genomförd iteration. Det som är utmaningen med det här tillvägagångssättet är att dokumentationen som produceras måste vara synkroniserad med koden i mjukvaran som utvecklas. För att lyckas med det krävs det en mer disciplinerad inställning hos de som skriver dokumentation än vad som krävs i det första tillvägagångssättet (Ambler, 2012).



Figur 2.3 – Agile Document Continuously SDLC, Ambler (2012)

Att uppdatera dokumentation enligt den agila metoden fokuserar också på att det ska vara så lite ansträngning som möjligt. Ambler (2012) rekommenderar att endast uppdatera dokumentation när det verkligen behövs. En organisation har mängder med dokument och alla dokument är inte uppdaterade, men alla dokument behöver inte heller uppdateras för att en organisation ska fortsätta att fungera och producera mjukvara och underhåll till en kund.

## 2.3 Systemdokumentation

För att uppnå kvalitativ dokumentation krävs en tydlig dokumentstruktur som efterföljs, samt dokumentationsprocess-standarder. Ledningen bör förespråka och stötta utvecklarna kring dokumentation. Dokumentstruktur i den externa dokumentationen betyder att samtliga dokument har samma struktur, samt att strukturen underlättar sökning av information och användbarhet (Sommerville, 2011). Följande lista innehåller några exempel på riktlinjer för god struktur (Sommerville, 2011):

- Alla dokument ska ha en framsida med följande information:
  - Projekttillhörighet
  - Dokumentnamn och typ av dokument
  - Skapelsedatum
  - Mottagare
  - Sekretessnivå
- Längre dokument uppdelat i:
  - Kapitel
  - Sektioner
  - Subsektioner
- Detaljerad information av refererande objekt i en lista.

Dokumentationsprocess-standard handlar om att ha ett definierat sätt hur skapandet av dokument ska gå till och att det efterföljs. Genom en bra process ökar chansen för god och effektiv dokumentation (Sommerville, 2011) och denna process bör starta så tidigt som möjligt. Storleken av ett projekt påverkar även hur viktigt det är med dokumentation, ju större projekt desto viktigare (van Vliet, 2007). Dokumentationsprocessen är en iterativ process, som genomgår flera kontroller innan den uppnår den kvalité som eftersträvas. Ett dokument skapas, där relevant information först läggs in, sedan förbättras i form av att formuleringar och presentationen görs tydligare, och till sist görs redo för att publiceras (Sommerville, 2011).

### 2.3.1 *Inline & Extern dokumentation*

Inline-dokumentation är riktad till utvecklarna och bör därför vara mer detaljerad och specifik, vilket möjliggör att ha mindre detaljer i den övergripande dokumentationen och ett mindre dokument. Det är bättre att fokusera på kvalitet framför kvantitet, och dokument med tusentals sidor ökar dessutom risken för fel och försvårar underhållsarbete med dokumentationen (Ambler, 2012).

Men för att inline-dokumentation ska kunna fylla sin funktion, räcker det inte med automatiserade lösningar eller endast självdokumenterande kod. Automatiserade dokumentationsverktyg kan bidra med diagram, index och modeller, men gör det inte tillräckligt bra och missar en viktig aspekt; varför. Dessa verktyg får inte med anledningen till varför en bit av systemet ser ut och fungerar på ett visst sätt, något som är viktigt information för underhåll (Raskin, 2005). Bra inline-dokumentation bör ha passande metodnamn, med kommentarer som förklarar varför utvecklaren har gjort på ett visst sätt. Att utföra dessa kommentarer direkt vid skapandet av kod är mycket lättare än att försöka titta på kod i efterhand och skapa dokumentation (Raskin, 2005).

Extern dokumentation bör struktureras så att olika dokument, eller delar i ett dokument är självständiga, för att underlätta uppdateringsarbetet. Om inte, skapas risk för redundant information och en ändring skapar behov av ändringar i andra dokument, vilket inte alltid görs och dokumentationen blir då inte uppdaterad. För att stödja detta finns det mjukvaror som hjälper en person att skapa kopplingar mellan dokument, utan att behöva skriva samma information i flera dokument (Sommerville, 2011).

### 2.3.2 Dokumentationsverktyg och hjälpmedel

För att hantering och skapandet av dokumentation ska bli lättare för utvecklare finns det olika verktyg att använda. Detta blir viktigare ju större projektet blir, då det rör sig om mer dokumentation att hantera. Verktyg och hjälpmedel kan variera. För den externa dokumentationen finns dokumentationshanteringssystem som kan hjälpa de involverade i ett projekt att hantera och få en överblick av relevanta dokument, och har funktioner som möjliggör relationer mellan dokument (Sommerville, 2011).

Wikipedia-funktion är ett annat verktyg för en organisation att hantera sin externa dokumentation, vilket är vanligt där organisationen arbetar agilt. Varje, så kallad wikisida, avser ett specifikt ämne (Ambler, 2014). Wikipedia är en internet-encyklopedia som tillåter de inblandade att publicera information på wikisidor. Fördelarna är att informationen är samlad på samma plats och lättillgänglig. Informationen bör även vara koncis och därför kan det behövas regler över hur publicering av wikisidor ska gå till för att få en bra organisations-Wiki. Frånvaro av regler kan skapa en svårhanterlig Wiki (Wallace, 2008).

Inom programmeringsspråket Java finns det ett kommenteringsverktyg som heter Javadoc. Genom att markera kodkommentaren på ett specifikt sätt, möjliggörs det att textstycket inom markeringarna kan extraherats till webbsidor. Dock bör dessa kommentarer inte bli för långa, då det kan bli omständigt att titta i koden för utvecklarna (Kramer, 1999).

### 2.3.3 Användning av systemdokumentation

Användning av systemdokumentation varierar, beroende på vilken roll personen har i projektet, samt vilken metod de jobbar utefter (Grubb & Taking, 2003). Inline-dokumentationen används av de som arbetar direkt i koden, medan designarkitekter oftare använder de externa dokumenten, angående överblicksbeskrivningar av strukturen och hur komponenter har kopplingar till varandra. Systemdokumentationen behövs för att kunna underhålla ett system, speciellt dokumentation av hela systemarkitekturen (Grubb & Taking, 2003).

Dokumentation kan fortfarande vara till hjälp även om den inte är uppdaterad, beroende på nivå och typ av dokumentation. Ju närmre koden, desto mer korrekt måste dokumentationen vara (van Vliet, 2007). Med andra ord är det viktigare att inline-dokumentation är uppdaterad, än extern dokumentation.

I Nicolas Anquetils m.fl. studie *A Study of the Documentation Essential to Software Maintenance* (2005) vill författarna få svar på frågan vilken typ av dokumentation som är mest användbar för de som underhåller en mjukvara. I sin studie genomförde de två olika enkätundersökningar. I den första enkäten bad de intervjupersoner att uppskatta hur viktiga olika dokumentationstyper är medan i den andra fick intervjupersonerna beskriva vad de faktiskt använ-

der för att underhålla ett system. Intervjupersonerna var 76 stycken personer med varierad erfarenhet och yrkestillhörighet, 26% inom ledning, 63% analytiker, 7% programmerare och 4% konsulter. Dock är detta deras tillhörighet, många analytiker i undersökningen utför programmering (Anquetil, m.fl., 2005).

I första undersökningen framkom källkod och kommentarer i koden, som de två viktigaste dokumentationstyperna, följt av logisk- och fysisk datamodell, klassdiagram och testplan. Dokumentation som ger en allmän överblick av systemet, liksom visionsdokument upplevdes inte som lika viktigt (Anquetil m.fl., 2005). I undersökningen om vad som faktiskt används hamnar fortfarande inline-dokumentationen, det vill säga källkod och kommentarer i koden, högst; första och tredje plats. Den logiska datamodellen tappar sin placering till testplan, kravspecifikationer och icke-funktionell prototyp, och i överensstämmelse med deras uppfattning att dokument som ger en allmän överblick inte är viktiga, används de inte heller i samma omfattning som inline-dokumentation (Anquetil m.fl., 2005).

Studien av Anquetil m.fl. (2005) visar att en stor del av de olika typer av dokumentationstyper som finns anses viktiga, om inte väldigt viktiga för att kunna underhålla ett system. Studien visar även att inline-dokumentation används mest, men tar också upp att vissa typer endast behöver användas ett fåtal gånger under ett mjukvaruprojekt för att ge tillräckligt med information (Anquetil, m.fl., 2005).

I en annan studie, *Production and Maintenance of System Documentation: What, Why, When and How Tools should Support the Practice* (2002) genomför Sulaiman, Bashah Idris & Sahibuddin en undersökning om dokumentationens relation till mjukvaruutveckling och underhåll. Även i denna undersökning svarade 76% att systemdokumentation är användbart för underhåll. När systemdokumentationen inte kan stödja utvecklare beror det på att den är ej uppdaterad och opålitlig.

Lethbridge m.fl. publicerade resultatet från tre utförda undersökningar i sin publikation *How Software Engineers Use Documentation: The State of the Practice* (2003). I en av undersökningarna studerade de hur en mjukvaruutvecklare arbetade i 14 veckor inom ett telecomeföretag, samt nio andra utvecklare var för sig, medan de arbetade i en timme. I Lethbridge m.fl. undersökning (2003) ansåg minst hälften av de intervjuade att dokumentation är effektivt eller, extremt effektivt vid fyra olika uppgifter kring systemanvändande. Uppgiften som fick flest röster var lärandet av ett mjukvarusystem, följt av testning av ett mjukvarusystem. Därefter kom att arbeta med ett nytt mjukvarusystem, samt lösa problem när andra utvecklare inte är tillgängliga för att besvara frågor. Men trots dessa svar visar det sig att fler personer spenderar sin tid att läsa av kod, istället för att läsa olika dokument (Lethbridge m.fl., 2003).

### 2.3.4 Problem kring systemdokumentation

Inom ett projekt prioriterar utvecklarna att göra sin produkt redo för kund framför att lägga tid på dokumentation, speciellt om det är ont om tid. Samtidigt kan situationer uppkomma där kunderna inte tar upp systemdokumentation i sina krav, då de inte är medvetna om hur viktigt det kan vara. Detta leder i sin tur till att systemdokumentation inte är något som projektledarna tar upp till sitt utvecklingsteam som hög prioritet, utan förväntar sig att utvecklarna gör det vid sidan av sina övriga uppgifter (Sulaiman m.fl., 2002).



Utvecklarens negativa inställning till dokumentation nämns också som en bidragande faktor till att dokumentation inte prioriteras. Enligt Sulaiman m.fl.(2002) anser utvecklare att dokumentation är en tråkig och omständigt uppgift, speciellt att hålla dokumentationen uppdaterad. I en annan studie, *Architectural Design and Documentation: Waste in Agile Development* av Prause & Durdik (2012) utfördes 37 stycken intervjuer i samband med ESEC, the European Software Engineering Conference. Intervjupersonerna var mjukvaruutvecklingsexperter, och hälften av de intervjuade experterna svarade att utvecklare inte bryr sig om dokumentation, vilket kan ha ett samband med att de upplever att dokumentation inte ger lika mycket tillbaka, som uppväger den tid och energi det tar att skapa. Experterna i studien höll även med i att utvecklare inte har fullständig kunskap hur de ska utföra effektiv dokumentation, samt att tidsbristen än en gång är en bidragande faktor (Pause & Durdik, 2012).

En mindre andel experter, runt 19%, påstår att det kan bli en intressekonflikt. Att prioritera god dokumentation försvårar att uppnå andra uppsatta mål, och resultatet blir att dokumentation bortprioriteras (Prause & Durdik, 2012), vilket stämmer överens med resultat från Sulaimans m.fl. studie (2002). Andra orsaker som bidrar till att dokumentationen blir utdaterad eller ej gjord kan bero på att ledningen inte är medveten om hur viktig dokumentation kan vara, samt kontinuerliga ändringar i projektet (Prause & Durdik, 2012). Det är lättare för projektledare att se kostnaderna som uppstår då dokumentation ska skapas, än de kostnader som kan uppstå då dokumentation inte görs, och därför kan dokumentation ses som en kostsam aktivitet. Kostnader som uppstår då dokumentation inte är genomförd, hamnar i andra aktiviteter som tid att hitta buggar, få förståelse för systemet och förseningar (Parnas, 2011).

Ett av de stora problemen kring dokumentationen är att den sällan är uppdaterad och aktuell (Anquetil, m.fl., 2005; Lethbridge m.fl., 2003). I sin studie från 2003, tyder Lethbridges m.fl. resultat på att speciellt extern systemdokumentation tenderar till att ej bli uppdaterad, och när den väl blir uppdaterad är det inte simultant med de praktiska ändringarna i exempelvis kod, utan efter flera veckor. Undantaget är om uppdatering av dokumentation är en del av hela processen (Lethbridge m.fl., 2003). Dock visar det sig att icke uppdaterad dokumentation kan fortfarande vara till hjälp för utvecklare. Enligt Lethbridge m.fl. (2003) tyder resultaten på att det finns en relation mellan hur korrekt och aktuell dokumentationen måste vara och typen av dokumentation, vilket stämmer överens med van Vliet (2007). Ju närmre koden utvecklaren kommer, desto mer korrekt och uppdaterad måste informationen vara, medan icke uppdaterade dokumentation i en högre nivå kan fortfarande ge en utvecklare en användbar överblick (Lethbridge m.fl., 2003).

### 2.3.5 Konsekvenser av dessa problem

Konsekvenserna av dessa problem kan ses som en negativ spiral. När utvecklare inte prioriterar dokumentationen, skapas en situation där den fort bli föråldrad. Om informationen inom dokumentationen inte stämmer överens, kommer den inte kunna stödja utvecklare och används inte lika mycket, samt bortprioriteras i framtida projekt. Och i längden kan detta skada kvalitén på produkten (Parnas, 2011).

En stor konsekvens av bristfällig dokumentation är tidsslöseri, vilket i projekt kan bli kostsamt. Som tidigare nämnt kan det ta längre tid att hitta buggar och få förståelse för systemet. Ett till exempel är om en nyanställd person ska lära sig ett system, så kommer denne ta mycket tid utav den personal som har kunskap om systemet för att få en förståelse. Om det finns

dokumentation kan personen vända sig till den, och de längre anställda kan lägga mer tid på sina faktiska uppgifter (Parnas, 2011).

Inspektion och utvärdering av systemet kan försvåras om dokumentationen inte stämmer, vilket kan resultera i att fel i koden inte upptäcks. Om systemversioner skulle där det finns buster i själva systemet lanseras till kunder, kan det bli kostsamt för organisationen och kunden (Parnas, 2011).

### 2.3.6 *Software Crisis och dokumentation*

På 1960-talet uppmärksammades det inom IT-industrin att ett flertal mjukvaruutvecklingsprojekt resulterade i likartade problem. Exempel på sådana problem är att projekten gick ordentligt över budget, att mjukvaran levererades sent, att produkten inte uppfyllde alla krav som projektet haft, samt att det inte fanns tillräckligt med dokumentation (Rai, Madan, Anand, 2014). Detta fenomen kom senare att kallas "Software Crisis" och fick NATO att på sent 60-tal kalla till en konferens där de introducerade termen "Software Engineering". Software Engineering ska ses som ett stöd som IT-projekt kan använda sig av för att strukturera upp projektet kring, detta för att undvika att de problem som projekten tidigare haft.

Ett problem inom Software Crisis är som sagt bristfällig dokumentation, vilket ledde till att den mjukvara som många projekt resulterade i var svår och dyr att underhålla. Trenden inom mjukvaruutveckling är att en mjukvara blir mer och mer invecklad och komplex, samt att ett samhälle blir mer och mer beroende av datoriserade lösningar och apparater. Detta kombinerat med bristfällig dokumentation gör att fenomenet Software Crisis förmodligen bara kommer att förvärras (Rai m.fl., 2014).

## 2.4 Sammanfattning av litteratur

Litteraturgenomgången har identifierat relevanta aspekter till forskningsfrågan, samt tidigare studier avseende användning av systemdokumentation. Kvaliteten av dokumentation påverkas av flera faktorer. Det krävs standarder och tydliga direktiv hur skapandet ska gå till och hur innehållet bör se ut, samt anpassat gentemot den person som är tänkt att använda informationen (Sommerville, 2011; Ambler, 2012).

Inom olika utvecklingsmetoder existerar det en skillnad i synsättet på dokumentation. Traditionell utveckling, vattenfallsmetoden, förespråkar mycket dokumentation, medan inom agil utveckling bör dokumentation ta så lite resurser som möjligt (Ambler, 2012). Det agila synsättet förespråkar att dokumentationen i ett projekt ska skrivas i slutet av projektet eller, i iterativa projekt, vid varje avslutad iteration. Anledningen till detta är att det inte ska läggas resurser på att skriva dokumentation som i slutändan av ett projekt ändå inte kommer att användas. När det kommer till att uppdatera dokumentation förespråkar det agila synsättet att göra det bara när det verkligen behövs. Att hålla all dokumentation inom en organisation uppdaterad är ett enormt slöseri med tid och resurser (Ambler, 2012).

I användning av systemdokumentation anses inline-dokumentation, det vill säga källkod och kommentarer i koden, som den viktigaste och den som användes mest, vid utveckling och underhåll av system (Anquetil m.fl., 2005). Enligt Raskin (2005) räcker det inte med automatiserade lösningar eller självdokumenterande kod angående inline-dokumentation. Kommen-

tarer bör ske av utvecklarerna simultant med kodskrivandet, där vad koden gör, samt varför exempelvis metoden ser ut som den gör, kort måste beskrivas (Raskin, 2005). Inlinedokumentation har större behov att vara uppdaterad och korrekt, än överblickande extern dokumentation (Lethbridge m.fl., 2003).

Extern dokumentation kan fortfarande vara användbar även om den inte är fullständigt uppdaterad då den fortfarande kan ge en viktig överblick (Lethbridge m.fl., 2003). Den externa dokumentationen bör innehålla mer övergripande information, samt fokusera på kvalitet framför kvantitet för att göra dokumenten lättare att underhålla. Den bör även struktureras smart och minska redundant information mellan dokumenten (Sommerville 2011; Ambler, 2012).

Litteraturgenomgången har även identifierat ett antal centrala problem som finns inom systemdokumentation. Problemen grundas i en negativ inställning hos utvecklare, intressekonflikter då det finns tidsbrist, låg medvetenhet hos ledning och kunder, samt att dokumentation kan vara massivt och komplext. Det skapas även en negativ spiral, där utvecklare tappar motivationen att göra eller underhålla systemdokumentation, för att den inte används, vilket i sin tur beror på att den inte är uppdaterad (Prause & Durdik, 2012; Sulaiman m.fl., 2002; Parnas, 2011).

Konsekvenser av bristfällig systemdokumentation kan bli kvalitetsbrister hos produkten, samt tidsslöseri i flera processer. Om det inte finns lättillgänglig information måste en person ta reda på informationen på andra sätt, fråga andra och ta tid från deras arbete, alternativt leta aktivt själv, exempelvis leta i koden (Parnas, 2011). Ett exempel på kvalitetsbrister är det fenomen som kom att kallas för "Software Crisis" på 1960-talet. Bristfällig dokumentation var en av anledningarna till att mjukvaruutvecklingsprojekt var svåra att underhålla och är ett problem som bör tas på största allvar. Detta speciellt eftersom trenden inom mjukvaruutveckling är att en mjukvara blir mer och mer komplex, vilket kombinerat med fortsatt bristfällig dokumentation skulle kunna leda till ännu värre underhållsproblem i framtiden.

Bra dokumentation ska dock inte ses som en kritisk framgångsfaktor i ett projekt. Ett projekt kan klara sig utan dokumentation, men kan även få problematik som skulle kunnat undvikas. Enligt Ambler (2012) existerar det ej någon stark relation mellan omfattande dokumentation och projektframgång. Dokumentation bör endast ses som ett stöd till utvecklare och kan spara tid inom projektet, alltså inte en kritisk framgångsfaktor (Parnas, 2011; Ambler, 2012).

## 3 Metod

*Detta kapitel motiverar den metod som till insamling och analys av det empiriska materialet. Uppsatsen använde sig av Jacobsens bok "Vad, hur och varför? - Om metodval i företags-ekonomi och andra samhällsvetenskapliga ämnen" (2002) som stöd till motiveringen av metoden.*

### 3.1 Kvalitativ undersökningsmetod

Den här uppsatsen fokuserade på att analysera ett få organisationer och deras arbete med systemdokumentation. Detta var för att bättre optimera den tidsperiod som var tillgänglig för studien, för att analysera djupare hur en organisation arbetar med systemdokumentation och därmed bättre förstå hur de hanterar de identifierade problemen med systemdokumentation. Eftersom arbetet med dokumentation utförs av utvecklarna på en organisation (Sommerville, 2011) fokuserade studien till stor del på utvecklarnas arbetssätt rörande dokumentation. Detta gjorde att en kvalitativ undersökning var det bättre alternativet jämfört med en kvantitativ undersökning, eftersom det kunde generera mer utrymme för de som intervjuades att uttrycka sig, vilket i sin tur bidrog med en mer klagörande bild av hur en organisation faktiskt arbetar med dokumentation.

### 3.2 Val av organisation och intervjupersoner

Uppsatsen hade som avsikt att undersöka hur en organisation hanterar de problem med systemdokumentation som uppsatsens litteraturgenomgång hade identifierat. Därför krävdes det minst en organisation som arbetar med mjukvaruutveckling, eftersom systemdokumentation är en del av en mjukvaruutvecklings livscykel. Kontakt togs med mjukvaruutvecklande företag i Skåne-regionen av logistiska skäl, varav två stycken ställde upp. Ett företag i Lund och ett med kontor i Malmö, som båda arbetar med mjukvaruutveckling. Företagen hittades genom personliga kontakter.

De personer vars arbetssätt som uppsatsen ämnade undersöka är de anställda i en organisation som arbetar med att utveckla mjukvaran. Fokus var på de personer som arbetar med att utveckla mjukvaran eftersom det är de som ska, enligt Sommerilles (2011) definition av en mjukvara, se till att dokumentationen blir gjord. Genom att intervjua utvecklarna i en organisation kunde uppsatsen få ut bäst möjliga information kring hur en organisation faktiskt arbetar med systemdokumentation och hur de hanterar litteraturgenomgångens identifierade problem.

### 3.3 Intervjuer

Vid möjlighet utfördes intervjuerna via fysiska möten eftersom denna typ av intervju bidrar till att lättare få en personlig kontakt kontra opersonliga medium så som via telefon (Jacobsen, 2002). Detta tillämpades i den mån det gick. Det är också uppsatsens uppfattning att en fysisk intervju leder till att den som intervjuas kommer att ha lättare att utveckla sina svar eftersom han eller hon inte begränsas av att uttrycka sig i skrift eller under påverkan av en uppkoppling via telefon. Den personliga kontakten leder också till ett mer avslappnat intervjutillfälle, vilket kan vara av vikt ifall den som intervjuas sluter sig själv på grund av att fel och/eller för utmanande frågor ställs (Jacobsen, 2002). Jacobsen (2002) beskriver vidare att intervjupersonen bör intervjuas i en miljö som är familjär för denne. Därför genomfördes intervjuerna på den plats som föreslagits av intervjupersonen. Personerna intervjuades en i taget eftersom det var individens personliga åsikt som var av intresse för att ge en bredd.

#### 3.3.1 Intervjuguide

Den här uppsatsen använde sig av en kvalitativ undersökningsmetod, vilket innebär att den information som har samlats in ska vara djupgående och utvecklade. Jacobsen (2002) diskuterar vikten av förstrukturering i form av begränsningar eller någon annan typ av styrning vid insamlandet av sådan information. Detta eftersom vid ett samtal som saknar strukturering och istället är helt öppet, existerar risken att viktiga områden som intervjupersonen själv missar att nämna faller bort (Jacobsen, 2002). Risken att information som samlats in i ett för ostrukturerat samtal är också att det samlas in för mycket information som gör analysarbetet mer komplicerat och informationen svår att bearbeta och att jämföra (Jacobsen, 2002). Det är däremot också viktigt att intervjuguiden inte är för strukturerad eftersom detta kan leda till att intervjupersonens möjligheter att utveckla sitt svar begränsas för mycket och på så sätt skadar den kvalitativa undersökningsmetoden (Jacobsen, 2002).

Med det i beaktande hade intervjuguiden utformats med öppna frågor för att ge intervjupersonen möjligheten att belysa det som denne ansåg var viktigt kring arbetet med problem inom systemdokumentation. Intervjuguiden användes också för att hålla intervjun inom relevanta områden.

Jacobsen (2002) berättar också om att för komplexa och invecklade frågor vid starten av en intervju kan leda till att intervjupersonen sätts under onödig press, vilket i sin tur kan leda till att hela samtalet låser sig. I början av intervjuerna som genomfördes för den här uppsatsen ställdes därför enklare, mer allmänna frågor, vilket skapade en avslappnad atmosfär där intervjupersonen kunde känna sig mer villig att svara utförligt på frågorna.

### 3.4 Analys av intervjuer

För att analysera de data som intervjuerna gav, använde den här studien sig av Jacobsens (2002) metod för att analysera kvalitativ data. Metoden består av tre steg, först *beskrivning*, sen *systematisering och kategorisering* och tillslut *kombination*.

Det första steget *beskrivning* innefattar en renskrivning av de rådata som de genomförda intervjuerna resulterat i. Detta innebär att transkribera de bandinspelade intervjuerna för att enk-

lare kunna hitta de delar av en intervju som är relevanta för studiens syfte (Jacobsen, 2002). Detta också för att kartlägga ifall åsikterna bland utvecklarna, gentemot hur deras organisation hanterar dokumentationsproblemen, går isär eller är eniga. Transkriberingarna skickades sen ut till de som intervjuades för att säkerställa att transkriberingen är helt korrekt.

Det andra steget *systematisering och kategorisering* handlar om att strukturera upp transkriberingarna för att jämföra och dela in de insamlade data i kategorier och underkategorier utifrån identifierade nyckelord och meningsbärande enheter i texten (Jacobsen, 2002).

Tillslut det sista steget *kombination* där de övergripande tabellerna hjälpte att kombinera vad intervjupersonerna har sagt för att se likheter och skillnader i inställning, arbetssätt och hantering av de identifierade problemen. Direkta citat har också använts för att visa på dessa likheter och skillnader.

### 3.4.1 Huvudkategorier

Med resultatet som grund identifierades tre områden, vilka ansågs relevanta för att besvara forskningsfrågan. Dessa tre huvudkategorier skapades efter att resultatet delats in efter nyckelord, som ansågs viktiga baserat på den gjorda litteraturgenomgången, i underkategorier som tillslut skapade huvudkategorierna. Tabeller användes för att kunna utföra en sammanställning av huvudkategorierna och dess underkategorier, för att ge en överblick över samtliga intervjupersoners svar.

Den första kategorin ville se hur individen förhåller sig till systemdokumentation, då det har visat sig ofta kunna föreligga en negativ inställning hos utvecklare och kallades därför *Individens inställning till dokumentation*. Inom denna kategori användes nyckelord som inställning, användning och informationsbehov för att hitta relevant information. Området ansågs relevant för att ge en bild av hur viktigt ämne systemdokumentation är för individen.

Nästa kategori, *Organisationsriktlinjer och hantering av systemdokumentation*, handlade mer om hur organisationen och den avdelning som intervjupersonen arbetade på hade för inverkan på systemdokumentationen. Faktorer av arbetsgruppen och arbetssätt framför personlig inställning. Nyckelord som denna kategori är byggd på är storlek, riktlinjer och ledningsstöd.

I skapandet av den tredje kategorin, *Hantering av problem*, nämndes olika problem som hade identifierats och intervjupersonen nämnde sin åsikt kring dem och om de existerade, och i så fall om de arbetade något för att motverka dem. Nyckelord här var problem, hantering och verktyg.

Dessa tre kategorier skapar tillsammans en helhet gentemot hur en organisation hanterar sin systemdokumentation i relation till kända problem med systemdokumentation.

## 3.5 Undersökningskvalitet

### 3.5.1 Validitet

Jacobsen (2002) presenterar två typer av validitet, intern validitet och extern validitet. Han definierar intern validitet som en intersubjektiv förståelse för ett visst fenomen. Det innebär att ifall flera personer är överens om en beskrivning av ett fenomen är det mer sannolikt att den beskrivningen är riktig. Detta resonemang ansågs rimligt och användes därför i analysarbetet av empirin, för att plocka fram väsentliga poänger. För att resultaten i den här uppsatsen ska anses vara sanningsenliga återkopplades de som intervjuats med sin intervju transkribering. Detta gav dem möjligheten att uppmärksamma felaktig information, men också att ta bort information som de inte vill ska delas alls. Intervjupersonerna fick den här möjligheten för att höja uppsatsens interna validitet.

Den andra typen av validitet, den externa, handlar om i vilken grad som uppsatsens resultat kan generaliseras (Jacobsen, 2002). Den kvalitativa undersökningsmetoden är i sin natur inte menad till att resultera i en slutsats kring ett fenomenens omfång eller frekvens. Den är mer ofta menad till att förstå och fördjupa ett begrepp eller fenomen. Styrkan är med andra ord att utveckla insamlade data till mer generella teorier (Jacobsen, 2002). Med detta sagt hade uppsatsen svårt att resultera i bestämda slutsatser eftersom den använde sig av ett mindre urval av organisationer. Däremot gick det att urskilja likartade mönster i hur de två undersökta organisationerna resonerar, men också skillnader, som kan vara värda att ta i beaktning för framtida forskning.

### 3.5.2 Reliabilitet

Jacobsen (2002) diskuterar kring vikten av att kritiskt granska reliabiliteten i studiens resultat, eftersom den som intervjuas utsätts för diverse stimuli och signaler. Detta är något som Jacobsen (2002) beskriver som *intervjuareffekt* och det innebär mer detaljerat att ett samtal anpassas i både stil och innehåll av de som deltar i samtalet. I en intervju påverkas intervjupersonen av exempelvis intervjuarens kläder, kroppsspråk och tal. En intervjuare som framstår som aggressiv och framfusig kan få totalt annorlunda svar jämfört med en intervjuare som framstår som oengagerad (Jacobsen, 2002). Dessa faktorer motverkades genom att intervjuarna klädde sig i så neutralt som möjligt använde en lugn samtalston, och försökte ha en så naturlig dialog som möjligt.

Jacobsen (2002) diskuterar en andra effekt, kontexteffekten, som kan påverka informationen som samlas in. Han beskriver två olika kontexteffekter, artificiell kontra naturlig och planerat kontra överraskande. I frågan om den första kontexteffekten, använde den här uppsatsen sig av en naturlig kontext istället för en artificiell. Intervjupersonen fick själv välja tid och plats, vilket har ledde till att personen antingen blivit intervjuad på sin arbetsplats eller i sitt hem. Skälet till detta är att intervjupersonen ska vara så bekväm som möjligt, vilket är fördelen med en naturlig kontext (Jacobsen, 2002). I frågan om den andra kontexteffekten, använde den här uppsatsen sig av en planerad kontext istället för en överraskande. Det innebär att intervjupersonen fick ta del av vilket ämne som intervjun skulle hålla sig inom och fick ta del av intervjuguiden med frågorna innan intervjutillfället. Intervjupersonerna kunde då förbereda sig innan intervjun.

Jacobsen (2002) diskuterar slutligen en tredje och sista sak som kan påverka uppsatsens tillförlitlighet, vilket är slarv vid nedteckning och analys av data. Vid ett intervjutillfälle kan inte en intervjuare få bättre information än den information som denne lyckas att registrera (Jacobsen, 2002). Detta innebär att all information som ges under ett intervjutillfälle kanske inte noteras, vilket skulle leda till ofullständig information som kan vara missvisande för de slutsatser som dras i uppsatsen. Detta motverkades genom att intervjuerna spelades in och sedan transkriberades. Intervjupersonerna frågades innan intervju startades om det var okej att intervjun spelades in. I samtliga intervjuer var det okej med inspelning och det behövdes inga anteckningar, som kunde ha distraherat intervjun.

### 3.6 Etik

Jacobsen (2002) diskuterar kring *informerat samtycke*, *rätt till privatliv*, samt *krav på att bli korrekt återgiven* som etiska aspekter i en undersökning, vilka studien använde sig utav. Dessa är tre aspekter hade intervjuarna i åtanke för att motverka saker som kan hota uppsatsens tillförlitlighet.

Den första aspekten, *informerat samtycke*, handlar om att intervjupersonen har gått med på att bli intervjuad (Jacobsen, 2002). Det kan vara på det viset att den som intervjuas känner sig tvingad till att ställa upp på grund av påtryckningar från kollegor, vilket ökar risken för att den som intervjuas inte utvecklar sina svar. Intervjupersonerna kontaktades direkt och gick självmant med på att ställa upp på en intervju, istället för att kontakta deras chef som ger dem direktiv att ställa upp på en intervju. I denna första kontakt fick de information vad undersökningen handlade om och skulle användas till och kunde därefter avgöra om de ville eller ej. Det är nämligen viktigt att intervjupersonen har fått full information gällande syftet med undersökningen, så att personen känner att den har tillräckligt med kunskap om ämnet och praktisk erfarenhet för att kunna delta. Jacobsen (2002) hävdar att en intervjuperson som deltar av fri vilja samt känner att den kan bidra ökar undersökningens kvalité.

Vidare diskuterar Jacobsen (2002) om intervjupersonens rätt till privatliv där han bland annat tar upp vikten av att det inte ska gå att identifiera en individ baserat på de svar och data som redovisas i uppsatsen. Detta är mer troligt att ske i kvalitativa undersökningar, eftersom det finns färre intervjupersoner. I undersökningen ställdes inga frågor angående personliga data så som ålder eller kön.

Sist diskuterar Jacobsen (2002) om den tredje aspekten, *krav på att bli korrekt återgiven*, som handlar om att den empiri som samlas in via intervjuer ska bli transkriberad. Vidare säger han att den inte ska tas ur sitt sammanhang eller på något sätt förvrängas (Jacobsen, 2002). Transkriberingarna skickades till intervjupersonerna för att de ska bli godkända och kunna ses som riktiga.



## 4 Resultat

*Detta kapitel beskriver studiens undersökningsobjekt och presenterar sedan resultatet av den empiriska undersökningen, genom att lyfta fram poänger och placera dessa inom de tre huvudkategorier som studien valt att behandla.*

### 4.1 Studiens undersökningsobjekt

Nedan beskrivs de två organisationerna och deras tillhörande intervjupersoner som har intervjuats för den här uppsatsen.

#### 4.1.1 Företag A

Företag A grundades på 1980-talet och växer än idag och har numera inte bara kunder i Sverige, men också inom norra Europa. Mjukvaran företaget erbjuder idag är en IT-lösning som ska bidra till en ekonomisk optimering av energiproduktionen för energiproducerande företag. De anställdas roller varierar från HR, till energitekniker och mjukvaruutvecklare. Kärnan i företaget är mjukvaran och majoriteten av de anställda arbetar således med den på ett sätt eller annat. De erbjuder också webbhosting, personlig service kopplat till mjukvaran, utbildningspaket och konsultering till sina kunder.

##### *Intervjuperson A1*

Denna person arbetar som utvecklare och har varit på företaget i över tio år. I början var det främst utvecklingsärenden, till exempel projektledare för installationer. Sen blev personen utvecklingsledare, men sen har personen gått tillbaka till att vara projektledare istället. Personen har varit med från starten av företagets mjukvaruprodukt.

##### *Intervjuperson A2*

Denna person samordnar för utveckling och informerar ledningen om olika projekt. Personen utvecklar också själv och gör servicearbete kring problem som energiteknikerna inte kan lösa. A2 har arbetat med utveckling i sex år.

#### 4.1.2 Företag B

Företag B är ett amerikanskt bolag som grundades i början av 1900-talet och har en internationell verksamhet, som omsätter miljarder amerikanska dollar. Bolaget har funnits i Sverige i över 80 år, där runt 4000 personer är anställda. Företag B arbetar med att utveckla hårdvaror, samt mjukvarulösningar till den offentliga sektorn, samt till både små och stora företag inom den privata sektorn. Intervjupersonerna från Företag B arbetar på mindre avdelningar.

### *Intervjuperson B1*

Denna har den officiella rollen Tech-Sales och har snart arbetat i två år på Företag B. Det mesta i B1:s arbete kretsar kring utveckling, där mycket handlar om utveckling kring bolagets moln-plattform. Arbetet handlar om att utveckla produkter till kunder, men kreation av olika sorts demos, där målet är att se om prototypen kan lösa ett visst problem, eller finna andra användningsområden.

### *Intervjuperson B2*

Denna person har en utbildning inom utveckling och jobbar inom IT-projekt, dock inte inom mjukvaruutveckling. Personen jobbar på en mindre avdelning inom Företag B och levererar olika typer av dokument som produkt. Där av togs bara vissa delar från den här personen med i beaktning i resultatet, då viss data var relevant.

## **4.2 Individens inställning till dokumentation**

De utvecklare som intervjuades hade skilda åsikter om dokumentation och dess vikt i deras arbete med utveckling. Majoriteten var överens om att det var *“ett nödvändigt ont”*, men det fanns också åsikten att intern dokumentation är överskattat, *“om man skriver bra kod som man kan läsa, så är inte behovet av intern dokumentation så stor alls egentligen”*. Deras personliga åsikter har inte ändrats något märkbart under den tiden som de har varit utvecklare, men de utvecklarna som arbetat under en längre tid ser mer nyttan av dokumentation eftersom vissa detaljer i koden kan vara svåra att förstå. Intervjuperson A2 berättar till exempel om att det är en tydlig skillnad på projekt där dokumentationen är bra och där den är sämre. A2 menar samtidigt på att personen aldrig varit delaktig i ett projekt där det har funnits ett bra utvecklat system för dokumentation.

### *4.2.1 Användning*

Användningen av systemdokumentation varierar hos intervjupersonerna. B1 säger att de inte använder, klassisk systemdokumentation, men använder sig regelbundet av commit-meddelanden och kommentarer, vilket bör ses som en sorts systemdokumentation. Användning blir även lägre då de förespråkar muntlig kommunikation mellan kollegor, samt att varje projekt skapar de något nytt, och behöver inte få förståelse kring något som redan existerar och är skrivet. Likadant är det för A1, som har varit med och skapat systemet från grunden och känner inte något behov av systemdokumentation, samt om de tär något A1 inte förstår vänder sig personen till koden. Vid de få tillfällen A1 har använt sig av systemdokumentation har det varit specifikationer, alternativt tanken bakom en ny feature, information som de sparar i ett webbaserat projektstyrningsprogram vid namn Redmine.

I samma organisation som A1 har A2 en annan uppfattning om systemdokumentation och använder andra delar. Informationen som A1 upplever sig ha användning av är exempel detaljerad information angående affärslogik. Men A1 känner att användningen av den dokumentation som finns minskar då det existerar problem med att den är utdaterad och det inte går att lita på den. Däremot ser A2 fördelen i längden med att ha information nedskrivet.

I Tabell 4.1 har cellerna blivit fyllda med de viktiga poänger som varje intervjuperson har delat med sig av. Tabellen används för att kunna sammanfatta och ge en överblick över vad intervjupersonernas personliga åsikter är till systemdokumentation, samt hur de använder systemdokumentation. Två faktorer som ansågs relevanta då individerna utgör organisationen. Inställning ansågs relevant eftersom exempelvis en tidigare studie, Prause & Durdik (2012), påstår att det finns en negativ inställning hos utvecklare. Användningen av systemdokumentation anses också relevant eftersom det då framgår om det är något som en organisation har behov av.

**Tabell 4.1 – Resultat i undersökningsmodell angående individens inställning.**

	A1	A2	B1	B2
<b>Inställning till system-dokumentation</b>	Intern dokumentation är inte nödvändig	Viktigt att ha, men svårt att hitta rätt nivå	Att det är överskattat	Nödvändigt ont. Vill ha, men inte göra
<b>Användning av system-dokumentation</b>	Använder gamla specifikationer	I arbetet med andra företags produkter för att undvika gräva i kod.	Använder inte, läser kod eller vad personen kallar "commit-meddelanden"	Använder för detaljerna personen inte kan räkna ut själv

### 4.3 Organisationsriktlinjer och hantering av dokumentation

Följande huvudkategori har delats upp i tre mindre områden som sedan sammanfattas i tabell.

#### 4.3.1 Storlek

I intervjuerna med A1, A2 och B1 sågs storleken på en organisation, inklusive avdelningsstorlek som en bidragande faktor till hur dokumentationen går till. B2 menar att det snarare beror på avståndet till den som dokumentet är ämnat till. Att till nära kollegor kan dokumentationen vara mer simpel och lättsam, jämfört om det är till en extern målgrupp.

B1 förklarar att i personens arbetslag arbetar det sex personer väldigt nära varandra, i vissa fall med parprogrammering där de roterar. Detta bidrar till att samtliga i arbetslaget har omfattande kunskap om systemet, samtidigt som den mindre storleken på arbetslaget förenklar muntlig kommunikation, då de arbetar bredvid varandra, vilket minskar behovet av dokumentation, menade B1. Dock gäller detta den avdelning B1 arbetar på. B1 poängterar att inom hela Företag B existerar "rigorösa dokumentationsprocesser" och att "det mesta finns någonstans inom Företag B" (B1, 28).

Både A1 och A2 säger att storleken påverkar hur viktigt det är med nedskrivna dokumentation. När företag A endast bestod av fyra personer blev det mer muntlig kommunikation, och nu när organisationen har vuxit känner A1 och A2 hur det har blivit allt viktigare med att ha systemdokumentation. Tidigare var den enda dokumentationen A1 och A2 gjorde, dokumen-

tation till kund. Med mer dokumenterat minskar risken att bli avbruten i sitt arbete då någon har en fråga. En person kan läsa sig till svaret eller en grundläggande information, och om det sedan fortfarande finns behov av information, då kan man vända sig till en person.

#### 4.3.2 Arbetsmetod & Riktlinjer

Mellan de två företagen skilde sig svaren om hur organisationen arbetar, samt vilka krav som ställs på dem. B1 förklarade att i sitt garage-team, arbetar de inte med så många typer utav systemdokumentation, eller dokumentation överhuvudtaget. "Vi jobbar lite ortodox" (B1, 28) och jobbar inte som många andra avdelningar på Företag B. Ofta gör de en produkt från scratch och behöver således ingen information från tidigare utvecklare. Den enda typen av dokumentation de använder sig av är kommentarer i sitt versionshanteringsystem, och det finns en standard hur man bör formulera sig i dessa, men utöver de finns inga krav eller riktlinjer från ledning. Då dessa kommentarer värdesätts, är det något som blir gjort. Poängen med dessa kommentarer eller "commit-meddelanden"(B1, 75) som de också kallas är att förklara varför, tanken bakom och inte vad. Att skriva vad anses vara onödigt då det är lättare att se det i koden.

Undersökningen fann att den avdelning som B1 arbetar på hos Företag B, väljer att ersätta viss dokumentation, som ska hjälpa kunden eller slutanvändare med underhåll, samt med att låta dem vara med i utvecklingen. Dels för att låta dem ge kontinuerlig feedback, men genom att ha med slutanvändarens utvecklare kan de få den kompetens de behöver, vilket minskar behovet av massa dokumentation som ska levereras med produkt.

Företag A arbetar mer utefter en vattenfallsmodell i sin utveckling. Varje projekt avslutas med att det dokumenteras, något som är med i varje planering av projekt. Dock kan det bli svårt vid större projekt, då risken ökar för att missar i dokumentationen sker. Ofta blir någon ansvarig för att skriva dokumentation, och sedan skickas ett dokument vidare till en person som får granska dokumentet för att ge feedback och se om det behöver kompletteras.

Ledningen ställer en del krav på att dokumentation ska göras, dock ligger fokus på att skriva dokumentation som hjälper kund, och inte direkt på systemdokumentation som skulle hjälpa de inom organisationen med utvecklingen. Inom de krav om dokumentation som ska hjälpa internt inom organisationen, ligger fokus på processdokumentation, framför produktdokumentation.

Ett annat verktyg som utvecklarna hos Företag A har tillgång till är Redmine, som har en wiki. Den använder de för att stödja sitt arbete och minska behovet att behöva fråga kollegor samma fråga flera gånger. Men då det inte existerar tydliga riktlinjer finns ingen standard över hur dokument ser ut utan dokumentationsförfattaren får frihet i sin formulering, samt beskrivningar i Redmine kan bli felaktiga. Det enda direktivet ledningen har givit de anställda är att dokumentation bör ske, men inte hur, utöver att det ska vara på engelska. De arbetar inte heller regelbundet med att skriva kodkommentarer, trots tillgång att till sådana verktyg finns.

#### 4.3.3 Stöd från ledning

Endast en av intervjupersonerna, A2 nämner hur ledningen ger stöd till att förbättra systemdokumentationen, samt dokumentation överlag. Hos Företag A har ledningen godkänt ett förbättringsprojekt av dokumentation.

Ingen av de intervjuade har efterfrågat några verktyg från ledningen för att förbättra sin dokumentation, vilket intervjuperson A1 anser kan leda till att systemdokumentation glöms bort av ledningen. Personen resonerar att ledningen fokuserar på att få ut värde till kunden och om inte de som utvecklar själva produkten själva belyser problemen kan det bli så att det förbises av ledningen. Personen tror inte att ledningen inte anser att problemen är viktiga. Detta understryks i Företag A där ledningen som sagt har godkänt ett förbättringsprojekt av dokumentation.

I Tabell 4.2 är samtliga celler fyllda med sammanfattande ord av vad varje intervjuperson uttryckte kring frågorna. Tabellen används för att sammanfatta vad intervjupersonerna sade kring frågor ur ett organisatoriskt perspektiv. Storleken på organisationen anses relevant eftersom Van Vliet (2007) påstår att behovet av dokumentation påverkas av projektstorlek. Riktlinjer behöver besvaras för att se om systemdokumentation har uppmärksammats av ledningen, vilket sedan leder in på nästa fråga: Ifall utvecklarna får stöd av ledningen.

**Tabell 4.2 – Resultat i undersökningsmodellen för en organisations riktlinjer.**

	A1	A2	B1	B2
<b>Hur mycket organisationens storlek påverkar hanteringen av system-dokumentation?</b>	Det är viktigare att ha systemdokumentation ju större organisationen blir	Ju fler på organisation, desto viktigare blir det att ha systemdokumentation	Arbetar nära sina kollegor, till exempel via parprogrammering, vilket personen anser gör behovet av system-dokumentation obefintligt	Anser att det påverkas mer av avståndet mellan den som skriver och den som ska läsa det
<b>Har de några riktlinjer de arbetar utifrån?</b>	Finns inga, mer beskrivet mellan utvecklare	Finns inte i nuläget, men tanken på skaffa ett format och ställe där all dokumentation kan samlas finns	Finns standarder som de rättar sig efter, men inte vid klassisk dokumentation	Finns mallar hos företag de arbetar hos, som följs lite efter behag
<b>Stöd från ledningen i hanteringen av problemen</b>	Anser att det glöms bort till förmån för att producera värde till kunden	Får resurser från ledningen så de kan satsa på det, triggat av att kunderna påpekade det	Frågan ställdes inte	Prioriteras inte, de säljer inte dokumentationen

## 4.4 Hantering av problem

Följande huvudkategori har delats upp i fyra delar, de första tre delarna är de problem som uppsatsen har identifierat och den sista delen behandlar hur företagen hanterar dessa problem. Avslutningsvis sammanfattas viktiga poänger i undersökningsmodellen.

#### 4.4.1 *Negativ inställning*

Samtliga intervjupersoner har den uppfattningen att det finns en negativ inställning till systemdokumentation bland utvecklare, “det är negativt belagt att göra sådant som inte är att skriva kod.” (A2, 52). Utvecklare vill arbeta med att skriva kod, inte skriva dokument. I en av intervjuerna framfördes liknelsen “du är en kock, du gillar att laga mat, men sedan så måste du diska efteråt.”(B2, 42). Den negativa inställningen fanns även hos intervjuperson B1 “Jag skulle nog säga överlag att jag tycker det är överskattat” (B1, 8).

Studien hittar ingen tendens till några försök att förändra den inställningen, varken från ledning eller intervjupersonerna. Hos B1s avdelning verkar inte den negativa inställningen påverka deras arbete. Däremot hos Företag A finns det risk för att den låga motivationen och negativa inställningen bidrar till sämre kvalitet. A2 berättar att om någon föreslår förbättringar i en dokumentation, finns det inget som tvingar andra personer att genomföra dessa, och att man som utvecklare skulle inte självmant gå in och redigera någons dokumentation för att man är missnöjd.

A2 nämner förslag på vad som hade kunnat öka motivationen och menar att problemet idag är att det är inte kul att skriva något som man inte vet om någon eller vem, kommer att använda, något som A1 håller med om: “man vet om att det man skrivit nu kommer förmodligen ingen att läsa”(A1, 54). Om det var möjligt att kunna få mer feedback på sin dokumentation hade varit bra, menar A2. Antingen i form av att få kommentarer från andra användare eller någon sorts gilla-knapp. Det hade ökat motivationen då en utvecklare faktiskt märker av om någon har använt och haft nytta av det som har skrivits.

#### 4.4.2 *Intressekonflikt & Prioriteringsfråga*

De två intervjupersonerna på Företag A tycker kanske inte att det existerar någon direkt intressekonflikt, då dokumentationen är en planerad del av sina projekt. Däremot tydliggör A2 att om de hamnar i ett scenario där nästa projekt måste påbörjas, går det före att skriva dokumentation, men det händer sällan. B1 menar också att göra klart en produkt går före god dokumentation och att det är en prioriteringsfråga, fast personen menar också att “ifall vi hade mer resurser så hade jag inte valt att lägga dem på dokumentation”(B1, 67).

Prioriteringsnivån för att ha systemdokumentation är låg hos Företag A. Ur ledningens perspektiv får utvecklarna tid till dokumentering inom projekten, men det handlar mer om dokumentation som kunden har nytta av. Det som ska ut till kunden är det som skapar värde, och då blir det inte att de prioriterar systemdokumentationen. B2 har samma uppfattning att dokumentation som ska till kund värdesätts högre och görs mer noggrant än dokumentation som endast ska användas av kollegor.

#### 4.4.3 *Utdaterad dokumentation*

Svaren skiljer sig åt kring problem med utdaterad eller brist på dokumentation. B2 har inte någon direkt erfarenhet av det, då deras commit-meddelande uppdateras kontinuerligt, samt de gånger någonting varit otydligt har B2 kunnat vända sig till kollega tack vare att de arbetar i en liten grupp.

Både A1 och A2 håller med om att utdaterad dokumentation är ett problem som existerar inom organisationen, men att det aldrig har lett till någon allvarlig konsekvens. Men åsikten mellan A1 och A2 går isär, då A1 menar att man ändå inte använder dokumentationen, medan A2 ser tydliga problem med brist av uppdaterad dokumentation. Då dokumentationen inte uppdateras kontinuerligt är det inte säkert att det går att lita på den, vilket minskar anledning till att använda den. Den direkta konsekvensen blir att det tar mer tid, då en person måste se om den faktiskt stämmer eller behöva gå in i koden och leta självmant, menar A2.

Problemet inom Företag A med utdaterad dokumentation eller att den inte finns är så stort att det inte hade räckt att alla inom organisationen hade arbetat med det i en vecka och gjort ny dokumentation. Processen med att kontinuerligt förbättra dokumentationen har brister, då fördelarna med det tycks vara för låga och det är ingen tydlig ansvarsfördelning.

En tydlig poäng är att intervjuperson A1 har varit med från starten av företaget och alltid haft den kunskap om systemet som behövs. "Det är jag som är dokumentationen här"(A1, 76). Idag har kunskapen spritt sig, och behovet av systemdokumentation kanske inte är lika högt, men A1 menar att om han hade försvunnit för år sedan hade det blivit ett stort problem för organisationen.

#### 4.4.4 Hantering av problem

Resultatet tyder på att samtliga är medvetna och håller med om att dessa problem existerar inom sin organisation, men det ligger lågt på prioriteringshögen att försöka åtgärda. Inom B1s avdelning används en typ av systemdokumentation, commit-meddelanden och kommentarer i deras versionshanterings-system. Då dessa värdesätts och används av de involverade utvecklarna blir den typen av dokumentation gjord kontinuerligt. Samtidigt prioriterar de att samtliga ska ha fullständig kunskap av det system de jobbar med, för att reducera behov av övrig systemdokumentation. Kortfattat ökar de samtligas kunskaps nivå och prioriterar kommunikation framför dokument och nedskrivna beskrivningar och modeller.

Inom organisationen hos Företag A syns tydligare konsekvenser av de problem som finns. På grund av negativ inställning och bristande motivation blir inte dokumentation uppdaterad och i vissa fall inte gjord överhuvudtaget. Dokumentation som prioriteras är den som kunden har nytta av, inte systemdokumentationen för internt bruk. Vid problem kan utvecklare kolla kod eller fråga A1, men ju större organisationen blir, desto större börjar behovet bli av gedigen systemdokumentation.

En annan anledning till att det inte har skett någon förändring hos Företag A är att "Det glöms bort helt enkelt" (A1, 68). Ledningen tänker inte på systemdokumentation, så länge varje projekt anses vara lyckat. Nu har ett projekt satts igång för att identifiera ett verktyg som kan stödja dokumentationsprocessen, med stöd från ledningen. Men fokus här ligger på att förbättra de produktdokument som ska ges till kund. Dock har det funnits vissa verktyg redan för att skapa systemdokumentation, men som inte har använts fullt ut, menar A1.

I Tabell 4.3 är samtliga celler fyllda med sammanfattande ord av vad varje intervjuperson uttryckte kring frågorna. Tabellen användes för att ge en överblick över hur hanteringen av olika problem kopplade till systemdokumentation. Valet gjordes att ta med vad som ansågs vara de mest relevanta problemen och intervjupersonernas svar angående hur de står sig till problemen och hur problemen hanteras.

Tabell 4.3 – Resultat i undersökningsmodell för hantering av kända problem.

	A1	A2	B1	B2
<b>Negativ inställning</b>	Känner igen och håller med	Håller med om att det är negativ be- lagt att göra så- dant som inte är att skriva kod	Håller med och menar på att personens kolle- gor också håller med	Håller med, jämför med att en kock vill laga mat, men ogär- na vill diska efter sig
<b>Hantering</b>	Ingen alls, led- ningen glömmet bort att problemet finns	Finns inget motiva- tionssätt, ifall nå- gon inte vill skriva dokumentation så blir den inte gjord	Jobbar inte med att hantera detta alls	Frågan ställdes inte
<b>Intressekonflikt &amp; prioriterings- fråga</b>	Håller inte med	Ingen egentlig intressekonflikt, men gör det när allt annat är gjort och om inget an- nat trycker på så finns det inget problem	Håller med, be- rättar att det prioriteras lågt vid intressekon- flikt	Håller med om, tid kommer läg- gas på det som ger värde för leveransen
<b>Hantering</b>	Tid sätts inom varje projekt	Gör det i slutet eftersom det mås- te göras	Ingen, skulle det finnas mer resur- ser skulle perso- nen ändå inte lägga det på dokumentation	Försöker alltid att avsätta tid för skriva dokumen- tation
<b>Utdaterad do- kumentation</b>	Anser att det är det största pro- blemet	Det är problemet som gör allt mind- re värt; går det att lita på dokumenta- tion du läser?	Vet inte, kommer inte i kontakt med den typen av system- do- kumentation	Håller med, svårt att gissa vad folk kommer undra om i fram- tiden
<b>Hantering</b>	Ibland blir de uppmanade till att fixa utdaterad do- kumentation, men det är inte alltid att det görs och det finns inga riktiga direktiv	I nuläget inte alls	Ersätts med vad personen kallar "commit- medde- landen"	Försöker att avsätta tid för att uppdatera do- kumentation



## 4.5 Sammanfattning

Samtliga intervjupersoner verkar vara överens om att systemdokumentation inte är det som de vill spendera sin tid på, då de sällan har behov av att använda det. Däremot gick åsikterna isär då en tyckte det var onödigt och en annan kunde se fördelarna med systemdokumentation. Nämnvärt dock är att personen som använde systemdokumentation regelbundet i form av commit-meddelande och kommentarer, var den som ansåg att systemdokumentation var över-skattat.

Storleken på ens avdelning verkar spela stor roll i hur viktigt man tycker det är med systemdokumentation. Samtliga arbetar på mindre avdelningar, och nära kollegor ökar graden av muntlig kommunikation framför att skriva ned den, samtidigt som de försöker sprida kunskapen mellan kollegor för att minska behovet av dokumentation. I ett av fallen låter man de externa utvecklarna vara med i utvecklingen för att låta dessa lära sig systemet från början istället för att skapa extra dokumentation och skicka med vid leverans till kund eller slutanvändare.

Utdaterad systemdokumentation visade sig vara ett problem hos Företag A, trots att de får tid inom sina projekt. Inom projekt ligger fokus på dokumentation som kunden har nytta av, istället för dokumentation som kan ge stöd åt utvecklare. Då A1 och A2, sällan använder sin systemdokumentation blir den sällan uppdaterad, vilket blir ännu en orsak till varför de inte använder den. Medan hos intervjuperson B1, uppdateras deras typ av systemdokumentation kontinuerligt, då den värdesätts och används.

## 5 Diskussion

*I detta kapitel diskuteras resultatet från den empiriska undersökning, samt ställs i förhållande med vad som har sagt inom litteraturgenomgången.*

### 5.1 Individens inställning till dokumentation

Enligt Sulaiman m.fl. (2002) existerar en negativ inställning hos utvecklare avseende dokumentation. Att skriva dokumentation är tråkigt och omständigt, och det är inte något som utvecklare bryr sig om (Prause & Durdik, 2012), en uppfattning som intervjupersonerna i denna undersökning delade. Sedan var åsikterna delade i hur viktigt det egentligen är med systemdokumentation.

En av intervjupersonerna, B1, anser att systemdokumentation är överskattat och menar att skriver man tillräckligt bra kod existerar inget behov av systemdokumentation. Raskin (2005) menar dock att det inte räcker, då bra kod inte är tillräcklig, då den inte besvarar frågan “varför?”. B1s avdelning använder sig av kommentarer i sitt versionshanteringsprogram, där de besvarar just frågan “varför?”. De använder och värdesätter dessa kommentarer och därför bör kommentaren om systemdokumentation “jag skulle nog säga överlag att jag tycker det är överskattat”(B1, 8) syfta på övrig extern systemdokumentation då en viss typ av systemdokumentation bevisligen används regelbundet. Detta ligger även i linje med Amblers (2012) åsikt att dokumentation ska omfatta endast det som behövs, används och att dokumentationen ska vara minimal, precis tillräcklig.

Trots att inställningen att systemdokumentation är onödig existerar hos utvecklarna enligt resultatet i den här uppsatsen, ser nästan alla de intervjuade nyttan av att ha det i lärandet av ett mjukvarusystem. Utvecklarna är med andra ord beredda att erkänna att de ser fördelarna med bra systemdokumentation samtidigt som de ser det som onödigt att skriva den. Detta kan förklaras med ett citat från en av de intervjuade “Det är ju som att du är en kock, du gillar att laga mat, men sedan så måste man diska efteråt. Det är ju ingen som vill det” (B2, 42). I den givna metaforen ser nog dock kocken nyttan med att diska efteråt, så att redskapen går att använda igen nästa gång kocken lagar samma maträtt. Precis på samma sätt ser nog en utvecklare nyttan med att skriva systemdokumentation så att den går att använda igen som ett inlärningsredskap i ett mjukvarusystem för nya utvecklare.

Van Vliet (2007) menar att dokumentation kan vara till hjälp för utvecklare även fast den inte är uppdaterad, men i den här uppsatsens resultat uttrycks det att icke uppdaterad dokumentation används mycket mindre på grund av att den inte går att lita fullt på. Att utdaterad dokumentation inte används för att den anses vara opålitlig stöds också av Sulaimans m.fl (2002) undersökning gällande dokumentationens relation till mjukvaruutveckling och underhåll.

### 5.1.1 *Användning av Inline-dokument*

Raskin (2005) påstår att det inte räcker att skriva bra kod, utan utvecklare måste någonstans kommentera varför strukturen ser ut som den gör och beskriva tanken bakom varje beslut. Även om B1 uttryckte att om de skriver bra kod behöver de ingen inline-dokumentation, så tyder resultatet ändå på att B1 delar Raskins uppfattning. Detta genom att i sina commit-meddelande skriver de på B1s avdelning tanken bakom och beskriver inte vad de har gjort. Dock tycks inte kodkommentarer inte vara aktuellt för Företag A, då det inte var något de nämnde eller lade vikt på.

## 5.2 Organisationsriktlinjer och hantering av dokumentation

Resultatet från samtliga intervjuer tyder på att det som ger direkt värde prioriteras, med andra ord går produkten och dokumentation till kunden före dokumentation för den interna organisationen och stöd till underhåll. Poängvärt här att notera är att intervjuperson B1 arbetar med mindre demos och prototyper, vilket gör det lättare att ha kunskapen om det systemet som det arbetas med.

### 5.2.1 *Hantering varierar efter storlek*

Det framgår av resultatet att hur en organisation hanterar systemdokumentation och problem kring det beror på deras storlek, eller närmare bestämt storleken på arbetslaget med utvecklare som arbetar med projektet. Van Vliet (2007) menar att ju större projekt, desto viktigare med dokumentation, vilket organisationen där A1 och A2 arbetar på börjar märka av, då deras system och organisation börjar bli större. När arbetslagen blir större blir det svårare att inte ha systemdokumentationen nedskriven. Att endast berätta allt muntligt, som i resultatet har visat sig vara substitutet, blir svårare eftersom mängden avbrott där en utvecklare måste pausa och förklara för en annan utvecklare ökar. Detta stöds av A2 som berättar att detta är det främsta problemet med att hantera systemdokumentation muntligt. Ett sätt att hantera problemet med att bli avbruten och behöva förklara går att finna i B1:s arbetssätt. B1 arbetar nära sina kollegor, via parprogrammering, med syftet att alla utvecklare i arbetslaget ska kunna alla delar av koden, vilket i sin tur leder till att behovet av andra utvecklares kunskap om systemet minskar. Att arbeta på B1:s vis fungerar förmodligen endast i mindre projekt, då mängden kod för alla utvecklare att förstå inte är så stor. Skulle B1 arbeta med större projekt kanske mer systemdokumentation hade varit aktuell, och personen nämner också att det finns systemdokumentation inom andra avdelningar på Företag B.

### 5.2.2 *Brist på riktlinjer*

Inom litteraturgenomgången tar Sommerville (2011) upp vikten av att ha standarder och riktlinjer i sin systemdokumentation för att uppnå kvalitativ systemdokumentation. Detta konfirmeras i den här uppsatsens resultat. Där framgår det att de två företagen har få eller inga riktlinjer rörande systemdokumentation. Detta har, till exempel, i Företag A lett till att dokumenten saknar struktur och att de ser olika ut beroende på vem som skriver, vilket gör det svårt att hitta rätt information för den utvecklare som letar. Saknas riktlinjer faller ansvaret på utvecklarna själva att skapa riktlinjer, något som har hänt i Företag A och som stöds av Sulaimans m.fl. (2002) undersökning där det visade sig att projektledarna förväntar sig att systemdoku-

mentation är något som utvecklarna får sköta själva vid sidan av. Dock eftersom dessa mer är överenskommelser mellan utvecklarna så följs de inte alltid fullt ut, vilket har lett till en situation i Företag A där utvecklarna upplever att system-dokumentationen är av dålig kvalitet gällande till exempel struktur. I Företag B har de istället ersatt klassisk systemdokumentation med commit-meddelanden, såsom beskrivs i resultatdelen. De har inga riktlinjer kring hur dessa commit-meddelanden ska skrivas och vad de ska innehålla, men trots detta upplever de commit-meddelanden som en kvalitativ ersättare till klassisk systemdokumentation. Den här uppsatsen hävdar att skälet till detta är arbetssättet hos utvecklarna på den avdelning som B1 arbetar på i Företag B. Genom parprogrammering får alla utvecklarna en god förståelse för koden och strukturen, vilket minskar behovet av klassisk kvalitativ systemdokumentation.

Det främsta skälet till brist på standarder och riktlinjer är att det orsakas av en negativ inställning till att skriva och hantera systemdokumentation hos utvecklare. I uppsatsens resultat går det att se att ifall det inte finns något önskemål från utvecklare om standarder och riktlinjer så skapas inte heller några standarder eller riktlinjer. Alla de intervjuade utvecklarna ansåg att systemdokumentation var något de var negativt inställda till och alla de intervjuade berättade att det fanns få eller inga standarder och riktlinjer. Detta beror med största sannolikhet på att systemdokumentation glöms bort av ledningen eftersom deras fokus är att producera värde till kunden, något som systemdokumentation inte anses göra baserat på de intervjuer som gjorts. Därför skapar de inte några riktlinjer och eftersom utvecklare är negativt inställda mot att skriva och hantera systemdokumentation så påminner de inte heller ledningen. Detta resonemang understöds av A1 och B2 som hävdar att ledningen och företaget inte säljer sin interna dokumentation och därför glöms den bort. Resultatet blir en brist på riktlinjer och standarder som i sin tur leder till nästan ingen hantering av systemdokumentation, som i sin tur leder till systemdokumentation av dålig kvalitet, vilket är fallet i Företag A.

Brist på riktlinjer eller som J. Wallace (2008) beskriver det, regulationer, är också ett problem i arbetet med dokumentationsverktyg som en wikipedia. Utan regulationer är inte wikipedian så givande som den kan vara och den blir svårhanterlig. Detta går att se bevis på hos Företag A som använder sig av en wikipedia via deras Redmine. A1 bekräftade att wikin kan vara svårhanterlig då det förklarades att beskrivningar i Redmine kan vara motsägelsefulla som en följd av att författaren har fria tyglar. Detta leder till att verktygen som används för att hantera systemdokumentation inte utnyttjas till sin fulla potential och kvaliteten på systemdokumentationen blir därav lidande. Det verkar dock inte finnas något konkret önskemål om bättre riktlinjer hos Företag A för att förbättra informationen som finns på wikin. Det finns inte heller något önskemål från varken utvecklarna på Företag A eller B om nya verktyg som kan hjälpa dem med att skapa och hantera systemdokumentation. Skälet till detta tros vara den negativa inställningen som diskuterats ovan.

### 5.2.3 Utvecklingsprocessens påverkan

Uppsatsen har funnit att dokumentation är det som skrivs sist i ett mjukvaruutvecklingsprojekt, vilket kan bero på att det prioriteras så pass lågt gentemot andra mål, vilket är fallet i Företag A. Eftersom dokumentationen görs sist så riskerar den att bli slarvigt gjord eller inte gjord alls. Van Vliet (2007) hävdar att för att få så kvalitativ systemdokumentation som möjligt bör den skrivas kontinuerligt under projektets gång. Det kan dock vara svårt att införa i ett företags utvecklingsprocess på grund av prioriteringsfrågan, samt den negativa inställningen som finns hos utvecklare. Det finns ett annat synsätt som Ambler (2012) presenterar. Han

hävdar att det räcker med att skriva det i slutet för att få den tillräckligt bra, samt att det sparar på resurser så som tid.

Den avdelning där B1 arbetar på, har en mer agil utvecklingsmetod och agilt synsätt på dokumentationen. De har begränsat sin systemdokumentation till det de känner behov av, vilket stämmer med vad Ambler (2012) påstår. Genom att främja verbal kommunikation, närhet och endast utföra den dokumentation de tycker ger nytta blir dokumentationen gjord. Företag A har mer av det traditionella synsättet att det ska vara gedigen systemdokumentation och det bör finnas även om de inte är säkra till vem eller varför. Denna situation verkar ha dragit ner deras motivation, samt att det tycks vara en för omständlig process.

#### 5.2.4 *Kunder får ledningen att agera*

Sommerville (2011) hävdar att för att uppnå systemdokumentation med kvalitet krävs det också att ledningen är med och stöttar utvecklarna i deras arbete med systemdokumentation. Ifall de får stöd från ledningen är något som de intervjuade hade svårt att svara på eftersom de inte själva upplever systemdokumentation som en nödvändighet och därför inte efterfrågar stöd så som diskuteras ovan. A2 nämner dock att ledningen på Företag A har påbörjat ett projekt där ett nytt system för hantering av organisationens dokumentation ska implementeras. Detta projekt startades dock på grund av efterfrågningar från deras kunder gällande bättre hjälpdokumentation och inte som ett resultat av dålig kvalitet på den interna dokumentationen.

### 5.3 Hantering av problem

I dagsläget visade de intervjuade företagen ingen konkret hantering av de identifierade problemen. De hanterar systemdokumentation på olika sätt och därav blev problemen tydligare hos Företag A.

#### 5.3.1 *Prioriterings- & Intressekonflikt*

Parnas (2011) nämner den negativa spiral, där systemdokumentation bortprioriteras, för att den inte används, då den är utdaterad, vilket i sin tur beror på att den bortprioriteras. Detta scenario identifierades i praktiken hos Företag A, vilket har lett till en svår situation som har resulterats i att så mycket information saknas så det hade tagit mer än en vecka för hela organisationen att göra en uppdatering.

Prioriteringsfrågan är ständigt aktuell. Bland de intervjuade ansågs det, som sagt, att det fanns viktigare uppgifter att göra än att sätta sig och uppdatera dokument eller skriva systemdokumentation. Att göra klart en produkt eller utföra andra uppgifter som gynnar kunden mer går före systemdokumentation. Detta beror på vad ledningen anser är viktigast, vilket är det som ger direkt värde. Dock kan det faktum att koden kommer alltid att existera, dra ner motivationen och vara en anledning till att prioritera systemdokumentation högre. Som samtliga intervjupersoner svarade, så går det alltid att titta i koden om man behöver få förståelse, sedan att det kan ta lite längre tid gör inte så mycket för utvecklarna.

Intervjupersonerna från Företag A får som nämnt tidigare utsatt tid till dokumentering inom sina projekt som ett sätt att hantera prioriteringsfrågan och på så vis få dokumentationen gjord. Detta verkar dock inte vara tillräckligt. Antingen är tiden för kort för att göra all dokumentation som borde utföras, eller existerar det brister i verktyg och de dokumentationsprocesser som Företag A arbetar utefter. Företag A arbetar efter vattenfallsmodellen som vissa (Görling, S., 2009; McConnell, S., 1996) hävdar är en oflexibel process när det kommer till att skriva dokumentation, vilket kan vara förklaringen. Skulle de arbeta mer agilt och endast skriva den dokumentation som är nödvändig, så skulle prioriteringsfrågan kanske inte vara lika mycket av ett problem.

Ledningen hade inte gett några direktiv mer än att det ska vara på engelska. Ej prioriterad systemdokumentation hos ledningen stämmer överens med vad Sulaiman m.fl.(2002) beskrev. Men nämnvärt är hur A2, nämner att det kommer ske en förändring, där det ska implementeras ett system som kommer att stödja all dokumentation och underlätta processen för utvecklarna.

### 5.3.2 Utdaterad dokumentation

Dokumentation som inte har blivit uppdaterad och är inte längre aktuell var endast ett problem hos företag A. Då de inom företag A inte har uppdaterat systemdokumentationen kontinuerligt, har problem blivit så stort att det inte hade räckt med en hel arbetsvecka för att få den helt uppdaterad. En situation som liknar det Lethbridge m.fl.(2003) lyfte fram som en bidragande faktor till att dokumentation inte blir uppdaterad.

Båda intervjupersonerna lyfter fram att den utdaterade systemdokumentationen är ett stort problem, som gör att de inte använder den. Men trots detta tyder inget på att organisationen gör något för stunden för att hantera detta problem, mer än en planering att ta in något sorts verktyg till dokumentation. Frågan är om de borde avgränsa sin systemdokumentation, likt det B1s avdelning har gjort. Ambler (2012) påstår att det är onödiga resurser att uppdatera sådan dokumentation som ska skapas bara för att, utan att en organisation bör fokusera på den dokumentationstyp som kommer skapa värde.

### 5.3.3 Negativ inställning

Utöver att företag A har tänkt sätta upp ett system som kommer att hjälpa organisationen, verkar inte någon av intervjupersonerna göra något åt problemen. Resultatet visar att samtliga var medvetna om olika problem som existerar inom ämnet, och att dessa problem existerar i olika fall bland intervjupersonerna, men trots denna medvetenhet är det inget man arbetar för att nå en förändring. Det går att spekulera i vad detta beror på, och av vad resultaten i denna undersökning tyder på bör det bero på att det glöms bort, vilket A1 nämnde, men även det faktum att det är ingenting som utvecklarna är intresserade av. Om en person inte har något intresse i ämnet, låter det logiskt att det inte finns någon motivation att investera tid och energi att förändra faktorer inom det ämnet. Det faktum att ingen av de intervjuade kunde minnas ett exempel om det gått illa i ett projekt på grund av för dålig eller obefintlig systemdokumentation, kombinerat med att den här uppsatsen inte har funnit någon studie som visar på det motsatta, tyder på att problemet med systemdokumentation inte anses vara värt att hantera.

Frågan är om systemdokumentation inte tas på mer allvar för att risken för att något ska hända är minimal, så att det anses vara onödigt. Bland dessa mindre avdelningar och organisationer blir kunskapen om systemet mer personbunden och mot det Ambler (2012) menar med sin agila dokumentation. Men A1 nämner det att om han skulle ha försvunnit för något år sedan hade det kunnat gå illa för Företag A, då A1 erhåller all viktig information, men ändå gjorde de inget åt saken. Här går det att spekulera i om organisationen var medvetna om de risker som existerade, eller om tankesättet var likt det Highsmith(2002) att konsekvensen hade inte gått att rädda med hjälp av dokumentation.

## 5.4 Sammanfattning

Vid jämförelse av det som kapitel 2 tog upp och undersökningens resultat, går det att identifiera likheter. Den negativa inställningen som Sulaiman m.fl. (2002) hävdar existera bland utvecklare framgår i den här undersökningen också. Systemdokumentation verkar inte vara något som utvecklare vill spendera tid på, eller något som ledningen prioriterar. Det som ger direkt värde tenderar att gå före.

Storleken på ett projekt och en organisation verkar påverka hur viktigt det är med dokumentation, vilket stämmer överens med vad van Vliet (2007) påstår. Hos B1 arbetar de nära och ett få antal personer, men intervjupersonerna hos företag A sade att det börjar bli viktigare nu när deras mjukvara har vuxit. Även om det har blivit viktigare för dem hos företag A så verkar det inte göra någon påverkan då ledningen inte har tagit upp det, samt utvecklarna har inte självmant. Sommerville (2011) hävdar att mjukvaruutvecklarna behöver stöd från ledning för att få kvalitativ systemdokumentation, något som inte förekom. Ledningen verkar prioritera dokumentation som skapar värde för kunderna, inte värde för organisationen.

Vid en jämförelse mellan organisationerna verkar B1s avdelning utföra och finna mer värde i den systemdokumentation de begränsar sig till, vilket fungerar med deras utvecklingsmetod. Företags As traditionella synsätt på dokumentation verkar ha placerat dem i en situation där de ska men inte gör sin systemdokumentation, vilket har gjort att de har hamnat i den negativa spiralen som Parnas (2011) nämner.

## 6 Slutsats

Forskningsfrågan som ställdes vid introduktionen av uppsatsen löd: *Hur ser en IT-organisations hantering och inställning ut beträffande systemdokumentation i relation till problem med systemdokumentation?*

Studien undersökte endast två organisationer och kan då inte fastställa att så här ser det ut i samtliga organisationer. Däremot pekar den empiriska undersökningen på att systemdokumentation är lågt prioriterat och hanteras på olika sätt, vilket kan bero på varierande faktorer. En organisation verkar prioritera det som ger direkt värde högre än att minska risker och problem som kan uppkomma med brist av systemdokumentation. Undersökningen visar att fokus ligger på att färdigställa en produkt eller uppdatering av mjukvara, samt göra dokumentation som slutanvändaren har ett behov av. Detta innebär att hanteringen av systemdokumentation blir lidande och i vissa fall nästintill obefintlig.

Inställningen bland intervjupersonerna är negativ och det är ingen arbetsuppgift som de vill spendera sin tid på. Behovet av systemdokumentation verkar inte vara särskilt högt, då det fokuseras på att sprida relevant kunskap genom verbal kommunikation och praktik, än att skapa dokument som en person kan vända sig till. Detta har lett till att uppsatsens identifierade problem med systemdokumentation inte ses som allvarliga problem och därför finns det inte heller några riktlinjer eller verktyg inom organisationen som kan hjälpa utvecklarna i hanteringen av systemdokumentation. För att ledningen ska uppmärksamma dokumentation som något som producerar värde så är det på grund av att deras kunder har uttryckt önskemål om detta.

Hur en organisation hanterar systemdokumentation påverkas också av vilken utvecklingsmetod och arbetsätt som avdelningen eller organisationen arbetar efter. Hos den avdelning inom företag B, där intervjuperson B1 arbetar på, har de ett annat perspektiv på systemdokumentation jämfört med företag A. B1s avdelning hanterar och utför regelbundet den systemdokumentation som de har valt att begränsa sig till och finner god användning av den. Företag A har en annan utvecklingsmetod och har tanken att det bör utföras gedigen systemdokumentation, men dokumentation med större kundnytta prioriteras. Frågan blir om de inte använder systemdokumentation lika mycket för de inte har en kort och koncis systemdokumentation tillgänglig, ingen effektiv dokumentationsprocess eller om de inte känner något behov.

Resultaten tyder på att organisationer prioriterar sprida kompetens framför att skapa systemdokumentation. Hos avdelningen där B1 arbetar på genom att rotera, vilket ger samtliga personer omfattande kompetens och reducerar risken att en nyckelperson försvinner med samtlig kompetens. Däremot hos företag A, upplevde de en situation där det var en nyckelperson, men nu har kunskapen spridit sig bland övriga inom organisationen. Hälften av intervjupersonerna uttryckte ett lågt behov av systemdokumentation då de upplever att de redan har fullständig kunskap och kompetens om mjukvaran, och att det inte finns någon dokumentation som hade kunnat rädda organisationen om något hade hänt dem. Och eventuellt stämmer det Highsmiths (2002) påstått, om att dokumentation ändå inte hade kunnat rädda en organisation vid en sådan situation där personen med all kunskap försvinner.



Efter en genomgång av litteratur och resultatet från intervjupersonerna tycks systemdokumentation vara som bilbältet och koden en krockkudde vid en liknelse med bilkörning. Folk använder bilbälte kontinuerligt, även om de inte har planerat att krocka. Att inte ha systemdokumentation är som att köra bil utan bilbälte. Vid en krock så blir skadan större än om en person skulle ha bältet på sig. Men det tycks vara en för omständlig process att ta på sig bilbältet. Dock finns krockkudden i form av kod, som dämpar lite, men det kan fortfarande gå illa. Än har ingen av intervjupersonerna krockat, och förhoppningsvis kommer de aldrig göra det heller, men en risk finns alltid.

## Bilaga 1 – Intervjuguide

- Kan du kort beskriva företaget, din avdelning och din roll här?
- Hur länge har du arbetat här?
- **(Vi berättar om vår definition av systemdokumentation)**
- Har du en åsikt systemdokumentation alternativt, vad är din bild? Onödig/viktig?
- Har din syn förändrats under din tid som utvecklare?
- Har den förändrats ifall du bytt roll och/eller företag?
- Vad för information är viktigast för dig när du utvecklar?
- Vilken information använder du när du utvecklar?
- 
- Finns det organisatoriska riktlinjer över systemdokumentation? Följs dessa riktlinjer?
  - Vems ansvar är det att se till så att riktlinjerna följs?
  - Om inte, är det inom avdelning bara?
  - Om inte, hur gör ni för att få reda på relevant information?
- Hur ser er systemdokumentationsprocess ut?
- Tror du att storleken på er organisation påverkar användning av systemdokumentation och informationsutbyte?
- Sker informationsutbyte muntligt istället för att sparas i dokument?
- Är systemdokumentation något som förväntas, eller planeras det in som en aktivitet inom projekt och vardagligt arbete?
- Hur fungerar er systemdokumentation?
  - Positivt?
  - Var ser det sämre ut?
    - Vad tycker ni är förklaringen till att det ser sämre ut?
- Vad för information anser du viktigast vid underhåll av system?
- Hur kan systemdokumentation hjälpa era kunder?
- Hur högt prioriterat är systemdokumentationen i ett projekt?
- Får ni stöd från ledning med ert arbete med systemdokumentation?
- Har ni de verktyg som behövs? Efterfrågar ni verktyg? Om inte, varför? Får ni de verktyg som ni efterfrågar?
- Kända problem inom dokumentation är;
  - Negativ inställning hos utvecklare
  - Ger inte tillbaka lika mycket som tid det tar
  - Intressekonflikt med andra mål.
  - Tidsbrist
  - Resursbrist
  - Låg prioriteringsnivå.
  - Ej uppdaterad, används ej.
  - Känner ni igen dessa? Håller med? Andra problem?
- Jobbar ni för att motverka dessa? Om ja, hur? Om inte, varför inte?
- Har ni upplevt problem på grund av brist av systemdokumentation?
- Har det uppkommit konsekvenser på grund av ingen systemdokumentation?
- Har du något övrigt som du vill ta upp?

## Bilaga 2 – Intervjuperson A1

A1 - Intervjuperson

S1 – Marcus

S2 - Fredrik

1. **S1: Vi tänkte att du kunde börja med att beskriva din roll här på företag A?**
2. A1: Ja, jag jobbar som utvecklare och har varit här på företaget i 16 år. Och i början var det väl främst utvecklings ärenden, som projektledare för installationer och sådant, så det är väl hela bitarna. Sen blev jag utvecklingsledare, sen har jag gått tillbaka till att vara projektledare istället. Sen har jag varit med ända från början av när produkten började byggas.
3. **S2: Vad har du för systemdokumentation?**
4. A1: Min personliga åsikt är väl att det är bra för att beskriva det som är komplext. Det vill säga för andra som inte vet hur det fungerar eller om det är något i systemet som är särskilt svårt att förstå, då är det bra. Om det är mer intern dokumentation så tycker jag inte att den är nödvändig, det är i alla fall min åsikt.
5. **S1: Har du alltid haft den här åsikten eller känner du att den har förändrats under din tid som utvecklare?**
6. A1: Ja, hm.. Vi har väl inte haft så mycket dokumentation innan, så att den har inte dykt upp, så att det är svårt att säga. Det känns nog som den är relativt oförändrad.
7. **S2: Vad för typ av dokumentation brukar du använda om du använder det?**
8. A1: Ja, oftast är det en form av färdiga dokumenterade, gamla specifikationer..
9. **S2: Hur ofta är det som du använder dig av systemdokumentation?**
10. A1: Det är inte ofta alls, det händer knappt en gång om året.
11. **S2 - Vad för typ av dokumentation brukar ni använda mest? Eller göra mest?**
12. A1: Alltså, nu har det blivit lite mer såhär att när du gör en ny feature så skriver du en liten snabb dokumentation om vad det är tänkt att, alltså vad tanken bakom om hur det ska fungera och så ligger det i vårt RedMine-system. Så att det mest så vi dokumenterar idag, kraven dokumenterar vi inte så där jättenoga, men vi borde kanske vara lite bättre på det. Men det är mest att vi inte gör det.
13. **S2: Finns det några riktlinjer inom organisationen?**
14. A1: Nej, det finns inte. Det finns när vi gör en ny feature, alltså om vi kallar det själva specbiten, så finns det till exempel det här dokumentet och det här dokumentet och vi ska göra en typ av manual också, en beskrivning. Men sen själva formatet och sådär är väl inte... Alltså det är ganska hårt beskrivet mellan utvecklare, men sen själva slutdokumentationen, där har vi inga rik-

tiga riktlinjer, utan det är faktiskt att man definierar och skriver ner lite vad man själv tycker, så alla dokument ser annorlunda ut. Även i RedMine så är beskrivningen där motsägelsefull. Så det finns inga riktiga regler

15. **S2:** **Så det är att ledningen har sagt att ni ska dokumentera men inte hur?**
16. A1: Ja. Generellt är väl att vi ska skriva allt på engelska.
17. **S2:** **När ni dokumenterar, hur ser processen ut?**
18. A1: Dokumentation är väl det sista steget egentligen när vi jobbar i ett projekt. Så när väl själva utvecklingen är färdig så blir det väl att skriva dokumentation. Innan har vi gjort en spec där vi skriver lite vad som ska hända och ibland kan man återanvända den texten, men för det mesta gör vi ett helt nytt dokument.
19. **S2:** **Får ni utsatt tid för att skriva dokumentationen?**
20. A1: Det finns med i direktiven för utveckling så att där har vi lagt av tid för att skriva dokumentation.
21. **S2:** **Hur tror du att storleken av er organisation påverkar skapandet och användningen av systemdokumentation?**
22. A1: Jag tror att det har blivit lite viktigare i och med att vi har blivit större, när det har kommit in en ny så har de haft lättare att läsa dokumentationen än att man själv ska behöva berätta det för dem. Förut var det mest att man pratade med varandra i och med att vi var ett litet företag på fyra personer så var det mest diskussioner.
23. **S1:** **Så när du började så var det mer diskussioner som var informationsutbytet istället för i skrift?**
24. A1: Ja, i stort sett. Det hände att vi skrev, eftersom vi var tvungna att ge en beskrivning till kunden också, men då handlade det mer om den typen av dokumentation, det handlade inte alls om intern dokumentation.
25. **S1:** **Tyckte du personligen att det var svårt att komma in och arbeta i systemet, eftersom det inte fanns något riktigt dokumenterat internt i skrift?**
26. A1: Nej, för när jag kom så började vi om från scratch. Vi byggde om det gamla systemet, så på det sättet så var jag ju som inblandad i varje bit, men områdes spec, så att jag behövde aldrig den dokumentationen. Så för mig var det inte ett problem.
27. **S2:** **Hur tycker du att systemdokumentationen här på företaget fungerar? Är du nöjd eller uppstår det problem?**
28. A1: Nej, alltså problemet är väl att man inte riktigt vet var man ska leta, tycker jag. För det mesta kollar man ju på RedMine, men om där det fanns ingenting så får man gå till någon (ohörbart) och kolla om där finns någonting. Det finns alltså ingenting som säger allting ligger här. Här ska vi leta. Annars är det väl inget problem som jag ser det, men nu är jag ju inte slutanvändaren heller så jag vet inte om de andra säger om det.
29. **S2:** **Kunderna, är de inne något i själva produkten och arbetar, eller får de produkten och underhåll av er?**

30. A1:       Precis, med produkten för de en genomgång i hur den fungerar eller om det är ny teknik som kräver undervisning av den punkten. Vi har gjort en mall och eventuellt har vi uppdaterat den också, men det har vi inte gjort de sista åren, så där är vi lite slarviga. Det kan jag väl säga är ett stort problem också, det här med att vi vill generera hjälpen på en massa olika språk. Detta kräver en massa extra arbete, det ska översättas och det ändrar man på ett ställe så ska man ändra på alla andra ställen.
31. **S1:       Vid ett underhåll av ett system, vilken information är viktigast för dig att det finns "easy access to", eller att det finns överhuvudtaget?**
32. A1:       Ja, jag vet inte. Jag använder inte dokumentation sådär direkt egentligen, utan oftast är det att jag kollar i koden för att se hur det hänger ihop. Men, visst, jag skulle säga att det är bra att hålla en typ av beskrivning med bilder som visar upp hur ett system hänger ihop, det kan vara klassdiagram.
33. **S1:       Tycker du att det fungerar bra att läsa i koden för att se hur det hänger ihop?**
34. A1:       Ja, jag tycker det.
35. **S1:       När ni utför ett projekt för en ny feature, hur högt prioriterat är systemdokumentation? Ses det som ett huvudmål eller mer som ett önskemål?**
36. A1:       Det är mer är att det ska göras, men det finns väl egentligen inte någon riktig tanke bakom det. Så att det har inte högsta prioritet, det har väldigt låg prioritet skulle jag säga. Men alltså det ska ju göras, men själva innehållet är mest att man skriver något.
37. **S1:       Men ni känner ändå att dokumentationen blir gjord?**
38. A1:       Ja, det blir den. Men, vi är slarviga på göra dokumentation för slutkunden. Även om vi gör några manualer, som också är till kunden, så är de kanske lite mer anpassade för utvecklare. Det är inga professionella författare som har skrivit det, som är bra på att skriva dokumentation. Det blir några slarvfel och det kan vara bilder som inte stämmer eller det är gammal text, text som har försvunnit och sådär. Där är vi ju lite slarviga kan man säga. Sen vill vi ju fokusera mer på det, själva tanken bakom som vill strukturera upp det.
39. **S1:       Det leder lite in på nästa fråga, har ni de verktygen som behövs för att kunna skriva den dokumentation ni önskar?**
40. A1:       Nej, jag tycker inte det. Vi har ju dels RedMine, vi har ju även verktyg för att dokumentera koden, inbyggt i visual studio, men vi har inte lagt ner tid riktigt på att säga det här verktyget ska du använda på det här sättet. Så vi har lite verktyg, men framför allt tycker jag att den som skriver dokumentet ska veta hur man skriver till slutanvändare.
41. **S1:       Känner ni att det från ledningshåll finns något krav på dokumentation eller känner ni att det är något som ni själva har satt krav på att det ska finnas?**
42. A1:       Nej, det kommer en del från ledningen. Så det är ett krav där att det ska finnas dokumentation just för att man skulle kunna ge den till kunden. Och det är mest för att de ska vara nöjda och känna sig säkra när de ska arbeta i systemet.
43. **S1:       Och om man pratar om dokumentation som är från utvecklare till utvecklare, finns det något krav på det från ledningshåll?**

44. A1: Ja, alltså vi har lite dokumentation på arbetsprocesser och sådär, men inte så mycket produktokumentation. Utan mer process, till exempel hur vi ska tillverka. Mer den typen av dokumentation har vi upplagda.
45. **S1: Har ni någon gång efterfrågat verktyg från ledningen för att kunna skriva bättre dokumentation?**
46. A1: Nej, det har vi inte, eftersom dokumentation är inte det som vi är intresserade av att skriva.
47. **S1: Okej, nu ska vi köra sista delen av intervjun. Vi har identifierat några problem med systemdokumentation, så vi kommer att säga dem, en efter en, och så kan du säga om du känner igen och håller med om dem?**
48. A1: Okej.
49. **S1: Det första är att det finns en negativ inställning till systemdokumentation hos utvecklare, att de gärna inte vill skriva det?**
50. A1: Ja, det kan jag hålla med om.
51. **S1: Och du känner igen det här också?**
52. A1: Javisst, och det är inte bara min personliga åsikt, utan det är också andra.
53. **S1: Och detta kan då bero på att det inte ger tillbaka lika mycket för den tid som det tar?**
54. A1: Ja, det skulle jag säga. Det stämmer. Och eftersom man vet om att det som man skrivit nu kommer förmodligen ingen att läsa.
55. **S1 - Att skriva dokumentation har en intressekonflikt med andra mål i ett utvecklingsprojekt?**
56. A1: Nej, det tycker jag inte direkt. I våra projekt finns det utsatt att det ska dokumenteras också så att det säkert blir en del av projekten.
57. **S1: Att det inte blir gjort på grund av tids- eller resursbrist. Det finns helt enkelt ingen tid att skriva det eftersom man måste vidare till nästa projekt eller mål?**
58. A1: Nej, det tycker jag inte heller.
59. **S1: Att det har en låg prioriteringsnivå?**
60. A1: Njaej, ni vet.
61. **S1: Att dokumentationen ej är uppdaterad eller används ej?**
62. A1: Ja, det är väl egentligen det största problemet om man säger, internt kommer det inte att användas för det kommer man inte göra något åt, men att den inte uppdateras för att det inte finns ett system för att uppdatera, det är väl problemet. Det är det som gör också att någon inte tycker att det är lönt att skriva egentligen.

63. **S2: Gör ni någonting för att motverka de här problemen?**
64. A1: Nej, det är väl om någon kommer ihåg att det finns dokumentation skrivet, när man ska ändra något, så kanske man blir manad till att fixa till det, men det är inte alla som gör det och det finns inga uppsatta direktiv som man måste rätta sig efter.
65. **S2: Försöker ledningen att ändra på er inställning?**
66. A1: Haha, nej.
67. **S1: Varför tror du att det är så, att det inte försöker ändras på?**
68. A1: Jag tror att man glömmer bort att problemen finns. Oftast att man vill sälja något nytt till kunden så att det är resultatet som är viktigare alltså det som vi har producerat, än vad för dokumentation som finns. Det man säljer ska ut så fort som möjligt till kunden och då tänker man inte så mycket på att det ska vara snyggt dokumenterat, utan det är enklare att producera mycket värde snabbt. Det glöms bort helt enkelt, så det är inte såhär att ledningen tänker att det inte är så viktigt, utan det är bara det att de glömmer bort att det finns.
69. **S1: I en brist av systemdokumentation, alltså att det inte är uppdaterat eller att du inte hittar rätt, har det lett till problem här på organisationen någon gång?**
70. A1: Nej, faktiskt inte. Det är väl mer det att det ser lite slarvigt ut, att det är gamla beskrivningar, men det har inte lett till något problem. Man använder inte dokumentationen så mycket egentligen.
71. **S1: Och du kan inte se några möjliga problem eller konsekvenser av att det är på det här sättet?**
72. A1: Nej, därför om funktionalitet ändrar sig, så är det väl mest det att det tillkommer något nytt. Så det är inte sådär att det är stora ändringar som gör att den gamla beskrivningen inte passar alls, utan det är mer att det kanske saknas någon detalj i beskrivningen.
73. **S1: Då har vi bara sista frågan kvar och den är ifall du har något övrigt som du vill ta upp, belysa eller tillägga?**
74. A1: Hmm... Nej, jag har inget.
75. **S2: Du har inget problem som du tycker att vi har missat?**
76. A1: Nej, inte vad jag känner till i alla fall. Jag vet inte vad slutkunden, vad dem tycker. Om de tycker det finns tillräckligt med dokumentation och hjälp och sådär, men oftast så ringer om det som de vill få förklarat. Det är jag som är dokumentationen här.
77. **S2: Vad skulle hända om du försvann från företaget?**
78. A1: Ja, nu har ju kunskapen hunnit sprida sig, men om vi går tillbaka något år så hade det kanske varit ett större problem. Men just nu så är det väl ingen fara på det sättet och det som man då verkligen behöver göra är att gräva i är finns det någonstans skrivet i dokumenten och då får man leta lite.
79. **S1 & S2: Okej, tack så mycket!**

## Bilaga 3 – Intervjuperson A2

A2 - Intervjuperson

S1 - Marcus

S2 - Fredrik

1. **S1: Då kan vi börja med att du kan beskriva din roll här på företaget?**
2. A2: Ja, jag samordnar för utveckling och jag informerar ledningen om olika projekt och jag utvecklar också själv och gör servicearbete kring problem som energiteknikerna inte kan lösa.
3. **S1: Hur länge har du arbetat med den här rollen?**
4. A2: I två och halvt, nästan tre år. Innan dess var jag utvecklare.
5. **S1: Hur länge har du arbetat som utvecklare sammanlagt?**
6. A2: I fem till sex år skulle jag säga.
7. **S1: Om vi då fokuserar på systemdokumentation, vad är din personliga åsikt kring det? Tycker du att det är viktigt eller att det kanske är onödigt? Hur ser du på det?**
8. A2: Nej, jag tycker att det är viktigt att ha, men att det är svårt att hitta rätt nivå och rätt ställe och hur mycket det ska vara.
9. **S2: Hur använder du systemdokumentation? Till exempel i situationer där du stött på problem i utvecklingen?**
10. A2: I fall jag utgår från andras dokumentation för andra produkter som vi använder när man utvecklar så behöver man dokumentation där, mest i det sammanhanget. Sen i vår egna dokumentation försöker man använda RedMine, man kan inte veta alla detaljer hur det fungerar, så innan jag går in i koden, jag försöker undvika det genom dokumentation, annars så måste jag gräva i detalj i koden och det tar ganska så lång tid.
11. **S1: Du sa att dokumentation är något som är bra att ha, men svårt att ha rätt nivå. Är det ett synsätt som du har haft under dina fem år som utvecklare eller har den ändrats under de här fem åren?**



12. A2: Nej, det skulle jag inte säga. Jag har aldrig varit i en situation där det fanns en bra dokumenterat utvecklat system, men det är skillnad mellan projekt som är bra dokumenterade och projekt som inte är det, där märker man att det blir skillnad. Jag tror att i början var det inte så viktigt, eller har jag själv inte lagt så mycket fokus på att dokumentera för att man själv blir inte drabbad på samma sätt om man inte gör det, nu när man jobbat lite med det så ser man mer värdet i detta, att göra så.

**13. S2: Vad för information brukar vara viktigast för dig, då du behöver komma vidare eller förstå koden?**

14. A2: Det beror lite på vilken nivå det är, om jag måste veta något som rör affärslogik då måste jag ha väldigt detaljerat, men om det till exempel bara är lite grundläggande om vilken knapp jag ska trycka på, då måste jag ha dokument som mer är inriktat på slutanvändare.

**15. S2: Vilken systemdokumentation är viktigast för dig, vad uppskattar du mest?**

16. A2: Är det mitt vardagliga arbete så är det mest arbetsdokumentation, mer detaljerad information, inte bara en screen shot och så, utan mer i detalj än övergripande.

**17. S2: Finns det några riktlinjer här inom organisationen över hur systemdokumentationen ska gå till?**

18. A2: Nja, inte i nuläget, men tanken är väl att sätta upp ett system som har ett visst format som man skulle hålla sig till och så att informationen är samlad på samma ställe.

**19. S1: Hur länge har det varit att ni inte har haft ett sådant system?**

20. A2: Sen vi började arbeta, vi har alltid haft dokument, men inget system. Det som vi har nu, som huvuddokument, var redan inte så bra när jag började här.

**21. S1: Och hur länge har den här idén om att det ska bli bättre funnits?**

22. A2: Hmm... Ja, hur länge är det. Kanske... Ja, de flesta visste väl om det, att det inte är så bra, men det har blivit mer och mer tydligt eftersom vi har lagt flera funktionalitet inne i programmet som inte finns alls i den gamla delen, så det blir mer och mer verklighet. Att det inte var så bra redan från början. Viljan att ändra detta kom förra året skulle jag säga, de flesta andra utanför som vi snackat med, eller att ledningen har bestämt att vi ska starta med det.

**23. S2: Hur ser processen ut vid systemdokumentationen?**

24. A2: De säger att vi ska skapa user guides efter varje projektlämning, som innefattar de features som har förändrats eller uppdaterats eller skapats.

**25. S2: Är det något som ni får tid för i varje projekt?**

26. A2: Ja, det är det. Man måste skapa user guides, annars klienten kommer undra hur man använder programvaran, hur fungerar detta, man har inget dokument att kolla i, så istället för att hjälpa varenda person som har frågor så är det bättre att de har ett dokument att kolla i.

**27. S1: Vems ansvar är det att de här user guides blir gjorda? Är det utvecklarna eller kollas det upp sen?**

28. A2: Jag skulle säga att det är projektledarens ansvar att se till att alla dokument som skulle skaffas skapas. För att om det inte finns några dokument, då skulle det hamna på huvudet hos projektledaren som har ansvar för att det blir gjort. Men det finns ingen kontroll av att specifika dokument skapas, men man märker när det är något som saknas.

**29. S1: Tror du att storleken på er organisation har haft en påverkan på hur ni sköter dokumentationsprocessen så som den ser ut idag?**

30. A2: Ja, det tror jag. Särskilt tror jag att när det är färre anställda så märker man att ju fler det blir desto viktigare blir det, men kan bara inte hålla all dokumentation i huvudet. Och att hela tiden hoppas på att andra kollegor tar sig tid för att svara på frågor som hur fungerar detta, kan du förklara.

**31. S2: Är det mindre muntligt informationsutbyte?**

32. A2: Ja, det har jag redan märkt. Alltså innan jag började var det hela tiden att man pratade med personen som man satt intill, alltså det blir mer dynamiskt och kaotiskt när man gör det på det sättet. Och när man dokumenterar, kan man istället säga, här är ett dokument som du kan läsa och har du frågor utöver detta så kan du komma till mig. Det blir mer organiserat.

**33. S1: Har du sett någon skillnad i effektivitet i hur ett projekt går framåt, om man jämför att man går och pratar med varandra med att man har det i skrift så att man kan referera till ett dokument när någon kommer och frågar om hjälp?**

34. A2: Det beror på om man menar processen för den som frågar en person frågor eller den som måste svara på frågor. Det är mer effektivt är den som ställer en annan person frågor för den kan gå vidare, men för den andra är det helt oeffektivt. Jag tror att generellt sett att om man har möjlighet att få information på en annan väg än bara att prata med någon, är det effektivare för alla. För då kan den personen i alla fall först och bara ställa vissa frågor, annars måste jag göra väldigt mycket om jag ska svara på alla dina frågor.

**35. S1: När dokumenterar ni under ett projekts gång?**

36. A2: Ja, jag skulle säga om det är ett stort projekt försöker man då att dela upp den med milstolpar eller att vi gör vissa grejer klart först eller att produkten är klar och sen kommer det andra. Och för varje sådan projektdel skulle jag säga att där dokumenterar man. När vi gör inte så stora projekt då kommer väl dokumentationen in där i slutändan med allt beskrivet om hur det ser ut, för annars måste man hela tiden uppdatera och det är mer naturligt att man gör det i slutändan.
37. S1: **Är det ett medvetet val eller är det som du säger mer naturligt, att det är bara så det blir?**
38. A2: Ja, jag tror det är resultatet av att man inte hela tiden måste uppdatera dokumentationen istället för att skriva ner det på en gång. Man får ett bättre flöde genom hela dokumentet, alltså direkt från att vi skapar det i nuläget. Annars så har jag små kapitlen ibland och sen, som sagt, kanske har uppdaterats lite grand så att det finns överlappningar.
39. S1: **Att göra det på det sättet, vad är fördelarna och nackdelarna som du har märkt av genom att skriva det på slutet?**
40. A2: Det är alltid risk att det inte blir gjort, eller inte så bra gjort där på slutet, för då börjar vi med något nytt. Och sen så tycker folket att det bara ska vara klart snabbt, sen måste man också ge feedback till ett större dokument och då kanske man missar grejer. Om jag skulle dela det i smådelar, kunde jag dokumentera eller ge feedback mycket mer detaljerat än vad jag gör när jag jobbar i det stora hela dokumentet på slutet. Det är en nackdel. Vi har också mer en vattenfallsmodell när vi utvecklar, inte agilt, att man dokumenterar den för sig och sen är man klar med den. Det är mer så att man gör allt och sen dokumenterar man och sen dokumenterar man.
41. S1: **När ni har gett ut en produkt till en kund och sen ska underhålla systemet under den tiden som kontraktet gäller på, vad för systemdokumentation tycker du är viktigast när du ska underhålla ett system, till exempel felsöka och hitta buggar?**
42. A2: Det beror på vilket område man arbetar på om det är en väldigt speciell (ohörbart), då vill jag först läsa lite mer allmän dokumentation, eller hur fungerar det i detalj, om jag ställer in värdet av detta så förväntar jag mig detta resultat och så sätter man upp det så man kan först kolla om konfigurationen hos kunden är rätt, men det utgår jag från att mina kollegor kollar upp sådant innan, men de måste också ha sådan information. Och själv så har jag ingen kod-dokumentation på det sättet, det gäller bara att kolla i koden. Vi har inget dokument som är användbart, sen finns det Java doc som genererar koddokument för största delen av programvaran.
43. S1: **Hur högt tycker du att prioriteringsgraden till att skriva dokumentationen är i ett projekt? Prioriteras det högt eller lågt?**
44. A2: Alltså inte jättehögt. Jag tror första mål är att det fungerar och sen att följa vissa riktlinjer mot guide lines och så, jag tror det är viktigare. Och sen kanske även test case är huvudmål att de finns, att dokumentation är lägst. Dokumentation kan vara fel, men om funktionaliteten

är fel så är det allvarligare så allt som har med produktkvalitet är viktigare än koddokumentation skulle jag säga.

- 45. S1: För att skriva dokumentation, känner du att ni har de verktyg som behövs eller finns det något som ni har efterfrågat?**
46. A2: Efterfrågat, nja. Detta eftersom ingen har kunskap om ett bättre fungerande dokumentationssystem eller process, det har alltid varit så. Med word-dokument och den dåliga e-hjälpen, så vi har ingen bättre fungerande lösningen i nuläget, men efter vi har haft en projektgrupp, eller er, har vi nu den "toolen" som vi försöker använda för framtiden.
- 47. S2: Får ni stöd av ledningen? Om de hjälper er i någon form med att skapa dokumentation?**
48. A2: De ger resurser för att göra det och tillåter att vi satsar på det. Hela projektet var triggat från ledningen, de har pratat med kunder och de sa att det är viktigt att man får mer stöd av systemet så att man kan jobba mer självständigt än att hela tiden ringa.
- 49. S1: Okej, då kommer vi in här på sista biten. Och då är det så att vi har identifierat genom en litteraturgenomgång några problem som finns inom systemdokumentation. Så vi tänkte gå igenom dem en efter en och sen så får du säga ifall du känner igen och håller med om dem eller inte.**
50. A2: Mm.
- 51. S1: Det första är då att vi har hittat att det finns en negativ inställning hos utvecklare till att skriva dokumentation, kanske speciellt dokumentation som är till för andra utvecklare eller för sig själv.**
52. A2: Ja, jag håller med om att det är negativt belagt att göra sådant som inte är att skriva kod. Det är inte populärt alls, jag tror väl att det är mer populärt hos utvecklare som aldrig ser en kund eller man har inte den kontakten att man känner mindre att det krävs och att det är någon som uppskattar det överhuvudtaget. Sen jag tror att det är sämre att skriva dokumentation för slutanvändare för där måste man verkligen förklara allting, om jag har en arbetsdokumentation så kanske jag skriver kodkommentarer inne i koden, då kan jag se att det är mer detaljerad dokumentation.
- 53. S1: Att skriva dokumentation det ger inte tillbaka lika mycket för den tiden som det tar att göra det.**
54. A2: Mm, jag tror det lätt är så när man inte vet för vilken målgrupp som man skriver och om man inte känner någon som använder dokumentationen och tittar på den, eller ingen kollegor som säger "det är bra så". Sen för, det är klart, det är när jag skriver själv, då måste jag inte läsa vad det är på det sätt, för då vet jag bäst. Men det kanske ger någon annan någonting. Så

på något sätt, skulle vara bra att få iallafall feedback för dem grejer som använder. “Hej, det var bra”, som “I like”-button på dokumentation, skulle vara bra.

**55. S1: Att det ofta har en intressekonflikt med andra mål i ett projekt, som det då skulle ta tid ifrån.**

56. A2: Ja, alltså, om det är på slutfasen skulle jag säga då är nästan allt typ klart och gjort, då är det ingen, alltså på ett sätt är det ingen utvecklare.. (Otolkningsbart). Det är mest att när allt är klart, då gör jag detta och det måste jag, så.. Om, det finns resurser och tid för detta och inget, nästa som trycker på så, så finns det inget problem där egentligen.

**57. S1: Nästa är just det att det då finns tids- och resursbrist, att det helt enkelt inte finns tid eller de resurser att skriva den dokumentationen som krävs. Är det någonting du känner igen och håller med?**

58. A2: Det.. Hm.. Alltså för oss kanske det.. Alltså andra ställen kan jag tänka mig att det är mycket mer press på utveckling och verkligen bli klart och att man inte får mer tid än.. (Otolkningsbart). Men här om vi inte blir klara, så måste vi skriva dokumentation, vi måste göra det så vi gör det också för att folk som vill titta behöva det direkt. Så du ser här inte så riktigt, att.. Att det inte händer eller mest så att man skriver för kanske för mycket som sagt, eller inte riktigt till målgruppen.

**59. S1: Att det ofta har en låg prioriteringsnivå och på grund utav det kan bli lite slarvigt gjort, eller ibland inte gjort alls.**

60. A2: Ja alltså, kontroll av dokument man gör, hos oss iallafall är inte inne i flödet, eller gjort för rätt person. För att, att läsa dokumentation är inte heller så spännande om man riktigt inte vill eller har användning för det.. (Otolkningsbart) Men vi försöker iallafall att den som använder toolet eller den funktion läser själv dokumentation och säger “förstår jag?” eller “detta måste läggas till” eller någonting. Försöker vi iallafall att göra.

**61. S1: Att dokumentation ej är uppdaterad, och att det är ett stort problem.**

62. A2: Ja, det mesta. Alltså, det, egentligen den delen som försämrar all tidsinsatser eller gör allt mindre värt är att den inte uppdateras på samma sätt. För varje gång du läser någonting då vet du; “kan du lita på det?”, “stämmer det så?”. Om jag har, till exempel på IT har vissa dokument för att konfigurera import-paket, om man skulle lita hela tiden på den, så blir det inte bra, för ibland är vissa parametrar stämmer inte, finns inte längre, inte uppdaterad. Så, an kan gör det som riktlinjer, men man kan inte lita riktigt på det. Det, som sagt, största problem i långtidsanvändning och så hur högt dokumentation är varderat, hur aktuellt och tillitigt.

**63. S1: Hur jobbar ni för att motverka till exempel det, att dokumentation inte är uppdaterat?**

64. A2: I nuläget inte alls skulle jag säga. För att dokument skaffas och ligger där.. (Otolkningsbart) Dokumentation, dokumentation blir uppdaterad igen. Något nytt skaffas, men det gamla finns fortfarande och ja..

**65. S1: Varför tror du att det är så?**

66. A2: Ja, för att man inte har haft tid för att fundera på processen och ändra den och kanske, ja, insätter sig att sätta upp en ny tool, tänka sig något nytt, prata om alla möjligheter, att ork. För det är så övergripande att om det inte, dem hinner skapa dem, dem andra användare, det finns ingen riktig person som, vad säger man, som täcker hela kedjan om det inte vals en och bestäms en som måste göra det, kommer in och tar det direkt. När du har ett system som bara används inom utveckling så är det klart att intresserar att själva.. (otolkningsbart)servern till exempel.. Och det tar jag gärna hand om för att den hjälper oss i vårt dagliga arbete, men en dokumentation, sätta upp ett dokumentationssystem, har jag ingen direkt benefit av.. kanske lite grann, men ja, det finns andra grejer man kan göra istället.

**67. S2: Uppkommer det problem på grund av ej uppdaterad information?**

68. A2: Ja, eller att dokumentation inte finns till exempel. Jaja, klart. Att, då måste jag.. annars måste jag gå in i koden och leta upp. Men problemet alltså, att det hela är så mycket så att det inte kan bara säga "Okej, men då dokumenterar vi allt en vecka och sen är det klart". Det kommer inte täcka allt och sen blir det ändå på slutet så att vissa specifika grejer måste sen gräva igen för att få, lägga till information i dokumentationen så..

**69. S1: Du sade att du höll med lite om att den här negativa inställningen hos utvecklare kan finnas. Jobbar ni här för att motverka den?**

70. A2: Ja, alltså, det, vi har in, vi skaffar inte dokumentation varje vecka, eller inte ens varje månad skulle jag vilja säga. Det är bara om vi har en utvecklingsprodukt som är i slutfasen. Där skaffas dokumentation och den personen som får, får den task att göra detta, om det inte är bra, kommer den feedback på den, alltså "du måste lägga till" då, men det finns.. Om en person vill verkligen inte vill göra detta eller inget som händer på grund, alltså det är klart, det måste finnas ett dokument, men man kan göra dokument på olika nivåer. Så, även om jag säger två gånger till "så kan du inte göra" så ser det fortfarande så ut, alltså, vad kan man göra? Det är bara att sätta sig in självt då är det lättare att låta det bli som det är. Så, något riktigt motivationssätt har vi inte, men jag kunde tänka mig att som sagt, det är om man skriver vissa artiklar och kunde få feedback, som "Det hjälpte mig här" eller.. (otolkningsbart).. Kanske är för avancerat. Jag tänkte till exempel på Microsofts arbetardokumentation när det finns.. (otolkningsbar).. eller man kan lägga in kommentarer inne i själva site:n som "hey, good example", "nej, det saknas något".. (otolkningsbart).

**71. S2: Är, du sade att en konsekvens blir att man måste lägga ner mer tid för att hitta det man söker, när dokumentation inte finns. Är den konsekvensen så liten att man inte prioriterar att förbättra hela dokumentationen?**

72. A2: Ja, jag tror att man kommer fortfarande rätt långt med att bara fråga folk och genom åren som man har lärt sig också lite grann, ja. Jag tror att det är, det funkar så lite ... (Otolkningsbart) Det är inte så att någon jobbar en hel dag och får inte bli störd... ..Om det inte är väldigt, vad säger man, statisk information. Vi har till exempel alltid en fråga som kommer upp, hur långt sparar det vissa typer data i vår databas, och då måste alltid Monica kolla upp vad det var, vilken.. (Otolkningsbart)..men det har vi lagt, alltså viss information har vi lagt mer i alltså, i Wiki.
73. S1: **Då har vi bara en sista fråga och det är om du har något övrigt som du känner att du vill tillägga nu efter att ha hört frågorna, eller om det är någonting du har glömt att ta upp eller du vill ta upp?**
74. A2: Nej, jag tror, som sagt alltså största bristen är att den inte är uppdaterad, och.. (Otolkningsbart)..kan den inte är uppdaterad kan man inte lita på den och sedan är man mindre beredd att sätta tid för den, bara för att den försvinner och ligger här och.. Ja alltså, jag vet vi har 20 worddokument som inte blir använda, då är det så roligt att skriva worddokument. Ja, om man har ett bättre system som fångar sådana problem med.. (otolkningsbart).

## Bilaga 4 – Intervjuperson B1

B1 - Intervjuperson

S1 - Marcus

S2 - Fredrik

1. **S1: Vill du börja med att beskriva företaget, vad du gör där och vad är din roll är på företaget?**
2. B1: Jag jobbar på Företag B. Det är ett jättestort globalt IT-företag, med jag tror vi är väl 400 000 personer, knappt. Så vi gör väldigt olika saker, men det mesta kretsar kring datorer och IT på något sätt. Och jag gör lite olika grejer. Min officiella roll är Tech-Sales, för vår moln-plattform som heter Bluemix, men sedan gör jag mycket annat. Mycket av mitt jobb handlar ju om, kretsar kring utveckling på något sätt. Antingen utveckla produkter för kunder eller i demo-syfte, eller till och med har jag varit med en del i produktutvecklingen av själva plattformen. Den delen är väldigt begränsad. Eller ja, utveckling överlag.
3. **S1: Ja, okej. Det är mycket utveckling. Hur länge har du arbetat på företag B?**
4. B1: I, ska vi se.. Det börjar närma sig Augusti för-förra året. Så två år i Augusti.
5. **S1: Två år? Yes. Har du bytt.. roll någon gång under dem här två åren eller har du jobbat med utveckling ända från början?**
6. B1: Ja, jag gör lite olika saker men nej officiellt har jag inte bytt roll nej.
7. **S1: Nej, okej. Det som vi har lite fokus på är ju just dokumentation dårå, och du säger att, eller om vi kan börja på det sättet, vad är din personliga åsikt kring dokumentation och systemdokumentation i sig, som dokumentation i kod och all sådan dokumentation som är till för utvecklare?**
8. B1: Hm.. Ja, jag skulle nog säga överlag att jag tycker det är överskattat.
9. **S1: Att det är överskattat?**
10. B1: Ja, för att om man ska förlita sig väldigt mycket på dokumentation, så behöver man inte skriva bra kod längre, känns det lite som att uppfattningen är. Och jag tycker att om man skriver bra kod som man kan läsa, så är inte behovet av intern dokumentation så stor alls egentligen. Alltså, i det, jag vet inte om jag går utanför frågan nu, men i det utvecklingsarbete jag har varit involverad i, så har det inte funnits någon dokumentation alls i princip. Men det har liksom funkat bra ändå, i alla fall den biten. Det har inte varit något problem, för ja, så länge man liksom skriver okej kod, som är välstrukturerad och alla har sina ansvarsområden så brukar det inte vara ett problem.
11. **S1: Är det en syn du har alltid haft gällande dokumentation eller något du har har anpassat efter dina två år på företag B, eller hur har din syn förändrats?**



12. B1: Hmm.. Ja, nu ska vi se. Bra fråga. Jag tror att det iallafall är en starkare åsikt nu efter att ha jobbat med utveckling. Annars under skoltiden så vet jag inte om jag tyckte så mycket om det överhuvudtaget.
13. S2: **Hur gör du själv om du skulle behöva information? Säg att du ser ett stycke kod eller nåt, som du inte förstår eller något sånt. Hur tar du reda på den information du behöver eller om du bara försöker förstå arkitekturen kring hela systemet eller applikationen du arbetar med?**
14. B1: Okej, så om jag skulle behöva utveckla någonting som förlitar sig på annan kod som jag behöver förstå hur den fungerar? Är det frågan?
15. S2: **Förlåt, kan du repetera?**
16. B1: Alltså om jag ska utveckla någonting som förlitar sig på annan kod, och jag behöver förstå hur den andra koden fungerar? Hur bär jag mig åt då?
17. S2: **Ja, exakt. Antingen det eller om du ska fortsätta på något som redan är skapat satt säga.**
18. B1: Ja, jag börjar med att läsa koden. Och ifall det är alldeles för oförståeligt, så går jag igenom, för ofta finns det någon versionhanteringssystem, typ Git. Och det finns ju egentligen kan man säga, en inbakad dokumentation i det. Man skriver meddelande varje gång man commit:ar saker, och genom det så kan man egentligen peka på ett stycke kod och kolla när det, den här koden checkades in till versionshanteringsystemet, vad var den kommentaren. Så det är väl steg två, och steg tre, ifall jag inte fattar det då, då tar jag väl kontakt med den som har skrivit det, ifall det går.
19. S1: **Är det någonting som förekommer ofta? Att du inte förstår?**
20. B1: Nej. Så sättet vi, alltså som jag har skrivit, den mesta koden, den mesta är ju själv bara jag egentligen, när jag har gjort demos, så då uppstår det ju inte. Den näst vanligaste är att jag har jobbat tätt med andra kollegor, så vi tillsammans jobbar på det liksom, och från scratch då, så då förlitar vi oss inte på någonting någon annan har gjort. Och då har vi ju väldigt god kommunikation så där uppstår det ju inte. Det är väl sällsynt egentligen. Det har väl hänt någongång, men då ja, de tre stegen.
21. S1: **När du precis började, och var ny på jobbet, hamnade du in i något sådant projekt som dem hade jobbat på under ett tag, eller har du alltid börjat från scratch? Jag tänker, hade du önskat någon typ av dokumentation när du började? Lite åt det hållet. Jag tänkte för att du säger att det är väldigt tight kommunikation mellan er i projektgruppen, var det svårt att komma in i den?**
22. B1: Nej. Så när jag började, var det enbart själv. Så då gjorde jag enbart demos med våran produkt. Och sen så var det i princip, så började jag jobba tätt med en grupp efter augusti, så det börjar närma sig ett år vi har jobbat tillsammans. Och då jobbar vi väldigt tätt och gör saker från grunden. Och sen så är det bara lite nu under våren som jag har jobbat i själva labbet, som är själva produktutvecklingen av själva plattformen, och det är nog enda gången jag har behövt förlita mig på kod, som någon annan har skrivit.
23. S1: Ska vi gå över på organisation nu?
24. S2: Ja, det kan vi göra.

- 25. S1: Då tänker vi, finns det några riktlinjer kring dokumentation som ni har när ni jobbar med utveckling och inom projekt och sådär?**
26. B1: Hmm.. Det beror lite på hur ni menar med dokumentation. Det är väl lite riktlinjer kring hur vi skriver våra versionshanteringskommentarer, som när vi checkar in saker till vårt repository, så finns det en standard hur man ska formulera sig, så att ja, det är lite standardiserat och lättläst. Men nej, inte på såhär klassisk dokumentation överhuvudtaget. Vi jobbar inte med dokumentation alls egentligen i dem typen av projekt.
- 27. S2: Är det inom din avdelning, eller är det något som är, det är svårt att svara på, men är det något som är inom hela företaget B?**
28. B1: Definetivt inte inom hela företaget B. Jag kan garantera att det finns väldigt rigorösa dokumentationsprocesser, någonstans inom företaget B. Men det mesta finns någonstans inom företaget B. Men vi jobbar lite ortodox. Det heter något såhär, delorganisation som heter Garage, Bluemix Garage är det som jag har jobbat i. Och dem, eller vi, som det team jag är en del av, jobbar annorlunda. och I den processen vi använder, så är det ingen dokumentation. Men det är nog snarare ett undantag än en regel. Är min gissning, jag har inte så bra koll som sagt.
- 29. S1: Du sade, ni jobbar inte med så mycket dokumentation alls?**
30. B1: Inte i Bluemix Garage nej.
- 31. S1: Okej, så ni har ingen dokumentationsprocess, liksom något såhär att ni dokumenterar i slutet av projektet eller under projektets gång eller någonting sånt, utan det görs inte alls?**
32. B1: Alltså, det beror ju på. Vi skriver ju kommentarer vi checkar in kod. Och det kan man ju se som viss typ av dokumentation, för det är en beskrivning på koden, så "arguebly" är det dokumentation, och i så fall så är det någonting, en process som sker löpande under hela projektet. Inte innan och inte efter, utan kontinuerligt. (hör ej)
- 33. S2: Hur stor är den avdelning du jobbar på? För att frågan är egentligen hur tror du att storleken på en organisation påverkar själva användningen av systemdokumentation? Men hur stor, hur många personer jobbar runtomkring dig?**
34. B1: Hm... Ja.. Då ska vi se, för det är lite såhär; "vad är en avdelning?" försöker jag tänka i förhållande till det. Då kan vi se. I liksom Garage-teamet är vi sex personer kan man säga.
- 35. S2: Yes. Blir det mycket muntligt informationsutbyte då också? Alltså, ni jobbar nära varandra?**
36. B1: Ja, extremt nära. Vi jobbar till och med med parprogrammering. Så vi liksom sitter två stycken framför samma dator och sen så roterar man varje dag. Så tanken är att liksom, att alla ska kunna alla delar av koden och då blir behovet av dokumentation obefintlig i princip.
- 37. S1: Har ni några problem med den här dokumentationsprocessen när ni gör det löpande, någonting sådant? Är det något du önskar fanns?**
38. B1: Vad sade du? Någonting jag önskar?

- 39. S1: Nu ska vi se, när ni, du sade att ni skriver lite kod och sådär, eller kommentarer till kod och löpande under projektets gång. Och du, förut sade du att du tyckte att det är onödigt att skriva mycket dokumentation och så. Men finns det något problem som du har märkt eller som du tänker på, har tänk på med sättet ni arbetar på?**
40. B1: Ur ett dokumentationsperspektiv antar jag att du menar då.
41. S1: Ja, precis.
42. B1: Nej, nej. Nej det kan jag inte påstå.
- 43. S1: Är det här en uppfattning som du känner delas av dem i garage-teamet?**
44. B1: Ja, det skulle jag absolut tro. Eller ja, jag är ganska övertygad att vi känner samma sak på den punkten.
- 45. S2: Vad är det för sorts, du behöver inte gå in på detalj så, men produkterna ni skapar, är det produkter som sen levereras till någon sorts kund, eller är det internt inom organisationen eller?**
46. B1: Det är till kunder, fast slutprodukterna av det vi gör är en så kallad minimum-.. och nu pratar jag då om garage-delarna av det jag gör, så är det en såkallad mininum-viable product, och den är inte tänkt att sättas i produktion men den är tänkt att vara tillräckligt utvecklad för att man ska kunna testa om det ger värde. Så det har mycket att, om en kund har en idé eller ett problem så bygger vi någonting som fungerar tillräckligt bra för att kunna testa om det löser problemet. Sen är tanken att om det funkar, då tas utvecklingen vidare, men då är det av någon annan part. Antingen kunden själva, eller vem som helst. Men vi tar det bara från scratch till en produkt man kan testa.
- 47. S2: Okej, förstår. För jag tänkte är det kunder sen då som skulle underhålla ett system, eller är det ni som hjälper kunder att underhålla systemet?**
48. B1: Det är inte vi i garaget. Men det kan vara en annan del av företaget B, typ såhär den stora konsultorganisationen, eller det kan vara kunden själv. Det är vilket som, det beror på, från case till case. Iallafall någon annan än vi.
- 49. S2: Då är det vara svårt för dig att svara på hur systemdokumentation kan hjälpa slutanvändaren eller kunden?**
50. B1: Ja, det är, har jag svårt. Men tanken är ju att under vårt, kanske en viktig poäng att tillägga, att när, under själva projektarbete i garaget, när vi har utvecklat den först minimum-viable product:en, så är det viktigt att kunden är med i det utvecklingsarbetet, för att kunna ge kontinuerlig feedback hela tiden. Och, optimalt är att dem har en eller flera utvecklare som del av det projektet också. Så att dem sen, när de går vidare, kan ta med sig kompetensen om, dels hur vi jobbar men också, ja, liksom kunskapen av det. Hur produkten fungerar och ta den vidare, så det är det som ersätter dokumentationen dåå.
51. S2: Det var intressant.
52. S1: Ska vi gå vidare till nästa delen?
53. S2: Ja det kan vi göra.

54. S2: **Klassiska problem, inom dokumentation, är exempelvis negativ inställning hos utvecklare, du säger själv att du inte är ett fan av dokumentation. Andra är klassiska åsikter är att det inte ger tillbaka lika mkt som tid det tar, tidsbrist, resursbrist, det är någonting som inte prioriteras hos ledning. Håller du med om dessa? Ser du andra problem hos dokumentation, systemdokumentation?**
55. B1: Hm.. Vad sade du? Resursbrist? och att det inte är värt det, tidsmässigt?
56. S1: Jag kan rabbla dem, så säger du bara om du känner igen det eller inte, vid varje gång.
57. B1: Hos mig själv då, antar jag?
58. S1: Det ger inte tillbaka lika mycket som tid som det tar?
59. B1: Hm.. Det håller jag med om.
60. S1: Det kan vara en intressekonflikt med andra mål i ett projekt?
61. B1: Hm.. Jag vet inte. Jag vet inte vad de betyder riktigt.
62. S2: Det är viktigare att göra klart en produkt än att göra god dokumentation.
63. B1: Jaha, ja, absolut.
64. S1: Vid ett projektslut finns det inte tid att göra dokumentation?
65. B1: Ja, eller alltså det är väl en prioriteringsfråga, är det inte? Jag varderar inte dokumentation så högt, så svaret är nog ja på det mesta av det där. Men nej, det finns inte tid då, för vi prioriterar annat.
66. S1: Känner du att det kanske är en resursbrist också? Eller hur man ska uttrycka det här, det kanske du har svårt att svara på.
67. B1: Jag hade, ifall vi hade mer resurser så hade jag inte valt att lägga dem på dokumentation. Ifall det besvarar frågan.
68. S2: **Har du upplevt att den dokumentation du har använt, att den inte är uppdaterad?**
69. B1: Jag har ju knappt sett någon så...Jag vet inte.
70. S2: Den används inte?
71. B1: Jag tror att jag aldrig har konsumerat någon sådan typ av dokumentation.
72. S2: **Ja, jobbar du någonting för att, det är ju svårt, ni har ju ett annat synsätt på dokumentation, men jobbar ni på något sätt för att motverka typ negativ inställning hos utvecklare, eller är det bara den allmänna uppfattningen? Att det är som det är?**
73. B1: Vi jobbar inte för att motverka negativ uppfattning nej.
74. S1: **Nej, och ni jobbar inte för att dokumentation ska vara en större del utav ert projekt heller?**

75. B1: Vi värdesätter ju, ja så kallade commit-meddelanden. För dem går vi tillbaka till liksom och kollar vad menade dem runt, när vi gjorde det här. I min uppfattning är ju det en dokumentationsersättare, eller ett alternativ till klassisk dokumentation. Och den, värdesätter vi. Så det beror ju lite, ja, på vad man menar med dokumentation, men. Och det absolut. Det försöker vi hålla oss liksom, försöker skriva bra commit-meddelande och förklarande och förklara. När vi skriver dem så försöker vi svara på frågan varför, och inte så mycket vad. Så när man, liksom checkar in kod, så varför gjorde jag det här, inte vad jag har gjort, för det kan man läsa i koden är tanken. Utan mer varför.
76. S2: **Har det någon gång blivit konsekvenser på grund av att någonting har blivit fel i den sorts kommentarer, eller att man har slarvat med koden, att den inte är självtalande eller vad man ska säga?**
77. B1: Hm.. Ja.. Det har det säkert. Inte särskilt liksom allvarliga konsekvenser, för det är typ bara att sträcka sig en meter bort och fråga vad någon menade när den gjorde något för att få ett svar. Men till den grad, jo, så har det säkert blivit.
78. S2: Yes, det var nog allt faktiskt. Som sagt, det blir lite svårt när ni jobbar på det sättet, men det är fortfarande viktig information.
79. B1: Ja, jag misstänkte lite det
80. S2: Det är det nya synsättet på det, eller nya.

## Bilaga 5 – Intervjuperson B2

B2 - intervjuperson

S1 - Marcus

S2- Fredrik

1. **S1: Om du kan då igen, beskriva din roll på företaget? Vad är det du gör?**
2. B2: Jo, i det här stora IT-projektet, så finns det horder av utvecklare, testare och labbmännskor. Och för att få någon form av struktur över det här, så försöker man använda verktyg där man fyller i vad man har jobbat med, enligt diverse olika processer, så vi använder agil process (oklart) folk får fylla i sina tasks och stories och grejer, hur många timmar dem har. Så jag jobbar med att stödja folk i det verktyget. I någon form av tech.support mer eller mindre. Och i och med detta så producerar vi ju en del dokumentation och utbildningar, och om möten och sånt hur folk ska tas med det här. Och då, som vanligt förstås, så gör vi inte så mycket dokumentation som det hade önskas att det skulle finnas. Och ja, en av de stora anledningarna är väl att dokumentation inte ger den omedelbara nyttan som man ser som något annat ger, utan det är någonting som man måste hantera i framtiden och liksom, värde idag är mer värt än värde i framtiden.
3. **S1: Hur länge har du arbetat med den rollen?**
4. B2: Jag har varit där sen den femte oktober tror jag, så ett halvår.
5. **S1: Har du jobbat med någon annan roll förut?**
6. B2: Nej, jag tog examen i september, nej i somras, så började jag jobba i september. Så det är mitt första uppdrag.
7. **S1: Hur skulle du beskriva din egen personliga åsikt kring dokumentation och kanske lite, ja, dokumentation?**
8. B2: Jag tror att jag har samma bild som de flesta andra, att det är något nödvändigt ont. Det är något som alla är överens om att vi behöver det men ingen vill göra det. Ja, ungefär så.
9. **S2: Har du alltid haft den synen eller har den förändrats under din tid då du har arbetat med dokumentation?**
10. B2: Jag vet inte. Alltså man har ju alltid stött på dålig dokumentation och blir alltid lika arg varje gång när man vill ha någonting, men det påverkar nog inte min egna drivkraft att göra det samtidigt liksom. Dels är det ju tråkigt och dels har man inte det trycket på sig från ens uppdragsgivare ofta, att göra det, för att det finns andra saker man kan göra som syns mer, och det är nog också en viktig poäng i att det inte blir gjort, för att det inte finns liksom en omedelbar vinst i det.

- 11. S2: Vad för typ av information är viktigast för dig, när du, om du läser någon sorts dokumentation eller behöver information som finns sparad någonstans?**
12. B2: Alltså det beror ju förstås på vad det är för något man tittar på, men oftast är det ju den här alltså detaljerna i någonting. Ifall jag ska använda någon form av program eller något flöde, så vill jag ju hitta dem sakerna, alltså dem här detaljerna som jag inte kan räkna ut själv, kanske hur ska man koppla det här. Medan såhär saker som syfte, och dem här, du vet, luddiga grejerna kanske är bättre, i min personliga åsikt då, att lämna åt sidan eller att separera dem. För det ska ju vara en instruktionsbok för hur man gör grejerna.
- 13. S1: Om vi hoppar över till del två, av tre i den här intervjun, så ska vi prata lite mer om organisationen. Och finns det några riktlinjer kring dokumentation, som ni arbetar med eller som ni följer?**
14. B2: Hm.. Nej?
- 15. S1: Typ hur man ska skriva det, hur man uppdatera det, hur man ska använd det?**
16. B2: Vi producerar ju en del dokument som ska liksom användas av användare, som i princip är kunderna av den tjänsten vi tillhandahåller. Och då finns ju lite mallar och sådant från företaget vi arbetar hos, alltså som templates, loggar och lite sådant man har att göra med. Och det följs ju, lite efter behag. Det används ju men inte alltid att man, det är inte kanske heller alltid så viktigt tycker jag att man gör det i den formen utan det är ofta med sunt förnuft egentligen som det görs. Jag tror, iallafall min erfarenhet så har det varit ganska så liberalt med hur man formulerar dokumentation och så, utan det är ofta den processen att någon skriver det, sen får någon annan läsa det, och se om det här känns vettigt och sen försöker man väl ofta få någon som faktiskt är expert på området. För att en svårighet kan ju vara om man är väldigt bra på något, så ska man förklara det så förstår man inte vad som är svårt, för att man är bra på det. (oklart)
- 17. S2: Är dem riktlinjerna och så vidare, är det väldigt lokalt för den avdelningen du är på? För det är ett ganska stort företag du är på tänker jag.**
18. B2: Jag skulle säga, att så som vi jobbar är nog lokalt, eller som vi känner. Vi är ändå en så pass liten avdelning, vi är typ tio personer ungefär, så vi kan liksom jobba som vi känner mer, mer med en sunt förnuft-basis. Medan om man skulle leverera till en större organisation, om du levererar komponenter, så hade du säkerligen fått vara mer strikt i din dokumentation.
- 19. S1: Skulle ni säga att det är ni, personligen som jobbar med det, att de tar ert ansvar att dem här mallarna följs då? Eller är det någon som har något typ av dokumentationsansvar på avdelningen?**
20. B2: Det är definitivt vårt eget ansvar. Alltså man kommer ju få skit liksom ifall du skriver dem här dokumenten dåligt. Men sedan också, ett av problemen är att om man hade skrivit det dåligt så finns det en stor risk att det inte upptäcks förrän någon ska använda dokumentation, vilket kan vara flera månader ner liksom. Man är alltid frestad egentligen att havsa ihop det, och lägga det på högen, och gå vidare med allt annat jobb som man måste göra. Och det är nog också en bidragande orsak till att dokumentationen blir lidande.
- 21. S2: Vad har ni för process i dokumentationen? Är det liksom inlagt i schemat? Du säger att det är eget ansvar? Ska det ske under hela projektets gång, eller är det något som ni går igenom i slutet?**

22. B2: Nu jobbar inte jag med att leverera en mjukvaruprodukt på det viset, utan jag jobbar närmre ett linjeorganisation-aktigt sätt, så att det blir inte dem leveranserna på det sättet. Utan då blir det ju mer, många av leveranserna vi gör är dokumentation i sig självt, så då blir det ju dokument.
23. S2: **Tror du att storleken på organisationen, vi har ju varit inne på det lite grann, att storleken på den avdelningen eller organisationen man är på, om det påverkar hur systemdokumentation, dokumentation, hur informationsutbytet sker?**
24. B2: Jag tror att det påverkas snarare av avståndet på något vis till den som ska läsa dokumentationen. Ifall jag producerar någonting som ska läsas av mina kollegor, så kan jag vara ganska lugn alltså med, jag kan göra lite som jag känner för. Men ifall jag ska producera någon form av dokumentation och det här ska stämpas med ett företagsnamn och sedan läsas för våra kunder, då kommer dem liksom vara mycket striktare med det. Och det spelar nog ingen roll hur stort ditt team är, eller din organisation, utan det handlar mer om hur långt det är, hur publikt det är på något sätt.
25. S1: **Skulle ni säga att när ni jobbar då mer inom avdelningen, dokumentation som ska knaske läsas av andra inom avdelningen, att ni gör det mer muntligt istället, än att det sparas ner någonstans i skrift?**
26. B2: Hm.. Vi försöker göra det, att dokumentera det i skrift. Och sen så finns det alltid, eller där jag jobbar finns det mycket konsulter, så det finns alltid en risk att du tappar kompetens som försvinner iväg, så då försöker man göra ganska detaljerade dokumentation på vissa grejer som är mer kritiska. Men det är också mycket som du säger att det hålls muntligt för att det går snabbare.
27. S1: **För då tänker jag lite den här muntliga, eller dels i skrift också. När du var ny, kände du att det var svårt att komma in i det, dokumentationsprocessen, att läsa, att förstå, att hitta information som du behöver?**
28. B2: Ja, jo, det skulle jag nog ändå säga, tror jag. För att dokumentation är ju också något som måste hållas levande, för att saker förändras ju, och då måste man gå in och uppdatera dokumentationen, vilket också är långt nere på mångas to-do-list. Så att vanligt sätt att göra på är att när du hittar någon som är ny, eller någon kommer in som ny, och så läser dem dokumentationen, så säger dem att det här var ju skit svårt att hitta, eller det fanns ingen dokumentation, så brukar man då sätta den nya personen att göra dokumentationen. För att om du precis har lärt dig det, så vet du precis vad du tyckte var svårt och vad som var konstigt. Och det är ett bra tillfälle, så det är en anledning. Och så den andra anledningen är att du är längst ner i hierkin, och då får du göra det tråkiga jobbet, så då får du göra dokumentationen liksom.
- Jag känner att jag svara på andra frågor ibland.
29. S2: Ja, men det gör inget. Det är jättebra. Det är bara utveckla så mycket som du känner att du vill.
30. S1: **Men det här sättet du beskrev precis, ser du några problem med det arbetssättet? Har du stött på några problem då, som du kan utveckla lite på?**
31. B2: Det här med att låta de nya göra dokumentationen?
32. S1: Ja, till exempel det eller att det är lite lågt prioriterat. Att det kanske inte är 100% uppdaterat, och att det är svårt att hitta.



33. B2: Alltså det är klart det är ett problem, men det vägs ju också mot vad det kostar liksom. För ofta någon alltså såhär, och det är också hur mycket utbyte du har med andra människor liksom. För om det är något som produceras innan det läggs upp så kan man ju ofta göra det lite slarvigt och sedan kan man prata med varandra, när man behöver det. Så att jag tror att det är klart att man hade velat ha en perfekt dokumentation, men det finns ju väldigt många saker man hade velat vill ha, så det blir ju en konstant avvägning i det. Jag tror att, jag vet inte ifall hur många människor som faktiskt hade prioriterat upp det, att lägga mer energi på det. Och det också, är en funktion av vem som ska läsa detta. Hur viktigt det är att dem har en funktion liksom.
- 34. S1: Tycker du att ni har dem verktyg som behövs för att den här dokumentationen och hantera dokumentationen? Har ni någon gång efterfrågat nya verktyg från ledningen? Har de stöttat i dokumentationsfrågan eller hur ser det ut?**
35. B2: Verktyg för din dokumentation är egentligen att du ha en text-editor, och det är ganska lugnt. Det brukar man ha. Och någon form av versionshanteringssystem är bra att ha, för du vill hantera dina dokument. Och eftersom det är i en IT-miljö, så finns ju dem sakerna. Och där tror jag att det behöver inte vara specifikt för just dokumentation, så det är inte ett så stort problem skulle jag säga.
- 36. S2: Vi har varit inne på det litegrann om ledningen, hur mycket pratar dem om dokumentation? Om att det ska utföras, eller är det något de bara tar för givet då?**
37. B2: Det beror på också. I det sättet jag jobbar är ju dokumentationen produkten. Så då blir det att jag bara ska göra det, och jag kan tänka mig att om man jobbar med, nu spekulerar jag, men att om man jobbar med mer leveransprojekt, typ kod, och så ligger du efter, vilket du gör för det mesta av gånger, så kommer dokumentationen prioriteras ner, för att det är någonting som inte är så lika viktigt, för det inte leverar samma omedelbara värde. Så jag skulle gissa att man uppifrån inte sätter samma värde på det, särskilt om man är lite synisk och ser det som att det är mer såhär, att det som kommer uppifrån är affärsvärdet dem vill ha, och affärsvärdet är antagligen inte lika starkt i dokumentationen, som produkten i sig. Du säljer liksom inte dokumentationen.
38. S1: Dags för sista delen?
39. S2: Ja, det kan vi köra.
40. S1: Sista delen då är dem här problemen som vi har identifierat, som finns lite kring dokumentation. Så vi tänkte gå igenom dem, en för en, och sen så kan du se om du känner igen dem, och om du håller med ett problem.
- 41. S1: Den första, som kanske blir lite svår då, men att det finns en negativ inställning hos utvecklare till att skriva och göra dokumentation.**
42. B2: Det tror jag definitivt att det finns. Jag tror att folk är utvecklare för att de gillar att utveckla, dem gillar känslan av att bygga logiska saker, och dokumentation är någonting helt annat. Är ju inte det som man liksom.. Det är som ju som att du är en kock, du gillar att laga mat, men sedan så måste man diska efteråt. Det är ju ingen som vill göra det. Så det tror jag definitivt är ett problem, eller det leder ju till att det blir mindre och sämre dokumentation.
- 43. S1: Att det ger inte tillbaka lika mycket tid som det tar. För den tiden det tar att skriva så får man inte samma värde. Lite som det du var inne på förut.**

44. B2: Jag tror att det blir, kortsiktigt så ger det betydligt mindre värde, och dokumentation är alltid en långsiktig investering. Det är till nästa person som ska ta tag i det här. Det kommer inte synas lika mycket på dina personliga mål, utan det blir mer.. Och det är också någonting, alltså det blir lite internt spel, vad gynnar mig och vad gynnar gruppen.
- 45. S1: Att det har en intressekonflikt med att arbeta, uppdatera och skriva dokumentation jämfört med andra mål, som ett projekt till exempel kan ha då.**
46. B2: Ja, definitivt. och det var ju det som vi kom in på att du får ju väga om jag ska lägga tiden på det eller på något annat som är viktigt för min leverans.
- 47. S1: Att man under ett projekts gång inte har tid helt enkelt, till att skriva det, att det finns ingen tid att göra det på.**
48. B2: Ja, det tror jag faktiskt. Och nu är ju mycket av dem här grejerna är ju, alltså i min erfarenhet att jobba med, det är andrahands erfarenheter ofta, vilket ni får ta höjd lite för. Men det tror jag definitivt, att det finns tidsbrist.
- 49. S1: Nästa är ungefär lite samma, att är bara det att det finns resursbrist istället. Att man har inte resurserna att arbete, lägga på dokumentation.**
50. B2: Ja, eller alltså, i IT-branschen är tid och resurser samma sak. Att antal timmar är allt. Det handlar inte om datorer eller..
- 51. S1: Att tid och resurser är samma sak, ja okej. Och sen så har vi då den här att det är låg prioriteringsnivå, men det har vi också sagt. Och att det finns mycket icke-uppdaterad dokumentation, eller att det dokumentation inte ens används överhuvudtaget.**
52. B2: Ja, precis. Och det kan ju också vara svårt för att man gissar på något vis vad folk kommer undra om i framtiden och det är inte säkert att de kommer undra om det, och det är där det liksom, när man väl hittar information som är helt utdaterad, så är det ett tecken på att ingen har tittat i den här på väldigt länge. Så det kan man ju också ta som en hint att fundera på värdet i det. Men ja, det är definitivt ett problem att det inte uppdaterats. Men jag tror att i alla dem här analyserna, där man diskuterar dem här problemen, så får man alltid väga dem mot kostnaden till exempel, och vad det ger. Och det är också svårigheten hitta vad som kommer vara bra.
- 53. S2: Har du några andra problem som du känner existerar inom dokumentationsvärlden..**
- 54. S1: ..Som inte vi tog upp nu?**
55. B2: Alltså det kan ju finnas problem med vem man riktar sig till, vem är liksom slutkunden för din dokumentation. Om du jobbar mot ingenjörer eller mer mot tekniska människor, så kanske du gör din dokumentation med en viss mutation, att du ritar såhär UML-diagram och grejer, men det kanske inte fungerar så bra mot andra människor. Det kan också vara, jag har stött på sådana problem att när jag presenterar dokumentation för folk som har känt såhär; "nejnej, det här kan inte jag"..(Otolkningsbart).. så det är också någonting att ta in i beaktning. Om du missförstår dokumentation är den värdelöst, och andra sidan ifall den för dummar informationen, om den förvrängs är den värdelös.
- 56. S1: Jobbar ni med att motverka dem här problemen?**

57. B2: Ja, vi försöker ju alltså sidosätta tid för det. Vi rapporterar alla timmar vi gör, i liksom, idag har jag jobbat med det här, i så många timmar. Och där så fyller man ju bara i jag har lagt så här mycket på.. (Otolkningsbart)..så tar man sig den tiden och försöker göra det. Och sedan så, ja, det hamnar ju ofta, eller ibland på, liksom, längre ner på ens lista.
- 58. S1: Förväntas det ett något visst timental på en vecka, som man då ska lägga på dokumentation?**
59. B2: Nej, nej, det skulle jag nog inte säga. Nej, det tror jag inte.
- 60. S2: Har du upplevt problem på grund av bristande dokumentation? Alltså, har du hamnat i situationer där du märker att något inte är uppdaterat, någon information fattas?**
61. B2: Ja, alltså det har ju ofta varit typ, många gånger har jag sprungit in i något sådant här; “gå till den här sidan så hitta hjälp hur du ska göra.. (otolkningsbart)” Och sådana här grejer som händer ju, och privat när man vill jobba, bygga något tekniskt liksom, så undrar man hur den här betar sig, så finns det ingen förklaring liksom, så blir man ju liksom mindre benägen att använda den produkten, för att man blir arg. Och om det är något form av företag som utvecklar en produkt, den kan vill vara hur bra som helst. Folk vill förstå hur du använder den, så spelar det inte så stor roll.
- 62. S1: Jag tänkte, att det ligger lite längre ner på ens prioriteringslista också, jobbas det något med att få upp den högre? Förespråkas det på något sätt från någon, eller?**
63. B2: Alltså, det beror ju på, ibland får man ju direkt tilldelande att nu har vi gjort det här, sammanställ ett dokument. Och det blir ju en leverans i sig att göra det. Det blir ju mer problem om man jobbar på någonting, och vill bli klart med det och det blir liksom en sidosyssla, så kan det hamna längre ner liksom.
- 64. S1: Har du något du vill tilläga? Som du känner att vi har missat, eller inte tagit upp?**
65. B2: Nej, tror inte det. Jag tror jag har täckt in.. mina upplevelser.
66. S1: Då får vi ju tacka för din tid.



## Referenser

- Ambler, S., 2012. *Agile/Lean Documentation: Strategies for Agile Software Development*. Tillgänglig på: <http://www.agilemodeling.com/essays/agileDocumentation.htm>. [2016-03-29]
- Ambler, S., 2014. *Best Practices for Agile/Lean Documentation*. Tillgänglig på: <http://www.agilemodeling.com/essays/agileDocumentationBestPractices.htm>. [2016-03-29]
- Anquetil, N., Cozzetti B. de Souza, S., M. de Oliviera, K., 2005. *A Study of the Documentation Essential to Software Maintenance*. ACM-SIGDOC Proceedings of the 23<sup>rd</sup> annual international Conference on Design of communication: documenting & designing for pervasive information. pp 68-75. ACM, New York.
- Boyd, A., Pucciarelli, J., Webster, M., 2012. *It's Worse than You Think: Poor Document Processes Lead to Significant Business Risk*. IDC Corporate Headquarters, Framingham, MA
- Correia, F. F., Serano Ferreira, H., Aguiar, A., Flores, N., 2009. *Patterns for Consistent Software Documentation*. Ploplog Proceedings of the IGM Conference on Pattern languages of Programs. ACM, New York.
- Lethbridge, T.C., Foward, A., Singer, J., 2003. *How Software Engineers Use Documentation: The State of the Practice*, IEEE Software vol 20 No 6.
- Grubb, P., Takang A., 2003. *Software Maintenance: Concepts and Practice*. World Scientific Publishing, Singapore, 2nd ed.
- Görling, S., 2009. *Att arbeta med IT-projekt*. Studentlitteratur, Lund
- Highsmith, J., 2002. *Agile Software Development Ecosystems*. Addison-Wesley.
- Jacobsen, I. 2002. *Vad, hur och varför? Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen*. Studentlitteratur Lund.
- Kramer, D., 1999. *API Documentation from Source Code Comments; A Case Study of Javadoc*. ACM-SIGDOC 199, Proceedings of the 17<sup>th</sup> annual international conference on Computer documentation. ACM, New York.
- Magnusson, Å., 2004. *Användningen och nyttan av systemdokumentation*. C-uppsats Institutionen för informatik. Handelshögskolan vid Göteborgs Universitet. Göteborg
- McConnell, S., 1996. *Rapid Development: Taming Wild Software Schedules*. 1 Edition. Microsoft Press.
- Parnas, D., L., 2011. *Precise Documentation: The Key to Better Software*. In S. Nanz (ed.), *The Future of Software Engineering*. Springer, Berlin.

Prause, C.R., Durdik, Z., 2012. *Architectural design and documentation: Waste in agile development?*, Software and System Process (ICSSP), 2012 International Conference on Software and System Process.

Rai, K., Madan, L. and Anand, K., 2014 Software Crisis. IJIRT, Vol.1, No. 11.

Raskin, J., 2005. *Comments Are More Important Than Code*, ACM Queue. Vol. 3, No. 2.

Rüping, A., 2003. *Agile documentation*. Hoboken, NJ: Wiley.

Sommerville, I. 2011. *Software engineering*. Boston: Pearson.

Sulaiman, S., Bashah Idris, N., Sahibuddin, S., 2002. *Production and Maintenance of System Documentation: What Why, When and How Tools Should Support the Practice*. APSEC 2002, Proceedings Asia-Pacific Software Engineering Conference, Queensland Australia, IEEE Computer Society.

van Vliet, H., 2007. *Software Engineering Principles and Practice*. Wiley.

Wallace J., 2008. *Using Wikis for Internal Documentation*. Tillgänglig på: <http://www.idealware.org/articles/using-wikis-internal-documentation-0>. [2016-04-25].