



# LUNDS UNIVERSITET

Ekonomihögskolan

*Institutionen för informatik*

---

## **Kraften i det dokumenterade ordet:**

**En studie kring hur olika faktorer inom intern  
systemdokumentation kan leda till ogynnsamma konsekvenser.**

Kandidatuppsats 15 hp, kurs SYSK16 i Informatik

Författare: Linnéa Hansson  
Mathilda Hezsö

Handledare: Bo Andersson

Rättande lärare: Benjamin Weaver  
Christina Keller

# Kraften i det dokumenterade ordet: En studie kring hur olika faktorer inom intern systemdokumentation kan leda till ogynnsamma konsekvenser.

ENGELSK TITEL: The power of the documented word: A study on how factors in internal system documentation can lead to unfavorable consequences.

FÖRFATTARE: Linnéa Hansson och Mathilda Hezsö

UTGIVARE: Institutionen för informatik, Ekonomihögskolan, Lunds universitet

EXAMINATOR: Christina Keller, Professor

FRAMLAGD: maj, 2020

DOKUMENTTYP: Kandidatuppsats

ANTAL SIDOR: 109

NYCKELORD: Intern systemdokumentation, kunskapsdelning, uppdatering, dokumentationskvalitet, tillgänglighet, dokumentationsskuld

SAMMANFATTNING (MAX. 200 ORD):

Intern systemdokumentation innefattar alla de artefakter som skapas under mjukvarulivscykeln, bortsett från användardokumentation. Det är ett av de effektivaste sätten att hantera och förvalta kunskap inom organisationen med konkurrenskraft som mål. Däremot är det ett arbete som ofta blir bortprioriterat och således leder till ogynnsamma konsekvenser. Syftet med studien är därför att kartlägga hur faktorerna uppdatering, dokumentationskvalitet och tillgänglighet kan leda till konsekvenser som dokumentationsskuld. I ursprung från faktorerna, ska även konsekvenser av dokumentationsskuld undersökas. Vi har, baserat på tidigare forskning, genomfört semistrukturerade intervjuer med systemutvecklare. Resultaten visar att det finns problemområden inom faktorerna som leder till att kunskap antingen går förlorad, blir tvetydig eller otillräcklig vilket kan hindra utvecklingsarbetet från att fortgå. Detta kan sedan resultera i dokumentationsskuld, men att fallet inte alltid är sådant. De konsekvenser som dokumentationsskuld framförallt leder till är förlorad tid, systemfel samt höga kostnader.

## Innehållsförteckning

<b>1</b>	<b>Introduktion .....</b>	<b>2</b>
1.1	<i>Problemformulering</i> .....	3
1.2	<i>Forskningsfråga</i> .....	4
1.3	<i>Syfte</i> .....	4
1.4	<i>Avgränsningar</i> .....	4
<b>2</b>	<b>Tidigare forskning .....</b>	<b>5</b>
2.1	<i>Kunskapshantering</i> .....	5
2.1.1	<i>Kunskapsdelning</i> .....	5
2.2	<i>Systemdokumentation</i> .....	6
2.2.1	<i>Intern systemdokumentation</i> .....	7
2.3	<i>Faktorer som påverkar intern systemdokumentation</i> .....	7
2.3.1	<i>Uppdatering</i> .....	7
2.3.2	<i>Dokumentationskvalitet</i> .....	8
2.3.3	<i>Tillgänglighet</i> .....	10
2.4	<i>Dokumentationsskuld</i> .....	10
2.5	<i>Sammanfattning av centrala undersökningsområden</i> .....	11
2.6	<i>Modell för centrala undersökningsområden</i> .....	12
<b>3</b>	<b>Metod .....</b>	<b>15</b>
3.1	<i>Val av undersökningsmetod</i> .....	15
3.2	<i>Insamling av data</i> .....	16
3.2.1	<i>Metod för utveckling av teoretiskt ramverk</i> .....	16
3.2.2	<i>Val av organisation</i> .....	17
3.2.3	<i>Val av respondenter</i> .....	17
3.3	<i>Intervjuer</i> .....	18
3.4	<i>Intervjuguide</i> .....	19
3.4.1	<i>Intern systemdokumentation</i> .....	20
3.4.2	<i>Uppdatering</i> .....	21
3.4.3	<i>Dokumentationskvalitet</i> .....	22
3.4.4	<i>Tillgänglighet</i> .....	23
3.4.5	<i>Avslutande frågor</i> .....	24
3.5	<i>Bearbetning av data</i> .....	24
3.5.1	<i>Transkribering</i> .....	24
3.5.2	<i>Kategorisering av data</i> .....	25
3.6	<i>Etik</i> .....	25
3.6.1	<i>Samtycke</i> .....	25
3.6.2	<i>Anonymitet</i> .....	26
3.6.3	<i>Transkribering och inspelning</i> .....	26
3.7	<i>Validitet</i> .....	26
<b>4</b>	<b>Empiri .....</b>	<b>27</b>
4.1	<i>Presentation av respondenter</i> .....	27
4.2	<i>Intern systemdokumentation</i> .....	28
4.2.1	<i>Kunskapshantering inom organisationer</i> .....	28
4.3	<i>Uppdatering</i> .....	29
4.3.1	<i>Uppdatering beroende på kodändringens storlek</i> .....	30

4.3.2	Anledning till bristfälligt uppdateringsarbete .....	30
4.3.3	Användandet av föråldrad dokumentation .....	31
4.3.4	Konsekvenser .....	32
4.4	<i>Dokumentationskvalitet</i> .....	32
4.4.1	Korrekthet .....	33
4.4.2	Läsbarhet .....	33
4.4.3	Struktur .....	33
4.4.4	Fullständighet .....	34
4.4.5	Konsekvenser .....	34
4.5	<i>Tillgänglighet</i> .....	35
4.5.1	Åtkomstrestriktioner .....	35
4.5.2	Sökning, namngivning och indexering i dokumenthanteringssystemet .....	35
4.5.3	Versionshantering .....	36
4.5.4	Konsekvenser .....	36
<b>5</b>	<b>Analys och diskussion</b> .....	<b>38</b>
5.1	<i>Intern systemdokumentation</i> .....	38
5.1.1	Kunskapshantering inom organisationer .....	38
5.2	<i>Uppdatering</i> .....	39
5.2.1	Uppdatering beroende på kodändringens storlek .....	39
5.2.2	Anledning till bristfälligt uppdateringsarbete .....	40
5.2.3	Användandet av föråldrad dokumentation .....	41
5.2.4	Konsekvenser av dokumentationsskuld .....	41
5.3	<i>Dokumentationskvalitet</i> .....	42
5.3.1	Korrekthet .....	43
5.3.2	Läsbarhet .....	43
5.3.3	Struktur .....	43
5.3.4	Fullständighet .....	43
5.3.5	Konsekvenser av kvalitetsattribut .....	44
5.3.6	Konsekvenser av dokumentationsskuld .....	45
5.4	<i>Tillgänglighet</i> .....	45
5.4.1	Åtkomstrestriktioner .....	45
5.4.2	Sökning, namngivning och indexering i dokumenthanteringssystemet .....	46
5.4.3	Versionshantering .....	46
5.4.4	Konsekvenser av tillgänglighet .....	47
5.4.5	Konsekvenser av dokumentationsskuld .....	47
5.5	<i>Slutgiltig reflektion</i> .....	47
<b>6</b>	<b>Slutsats</b> .....	<b>49</b>
6.1	<i>Förslag till vidare forskning</i> .....	50
<b>7</b>	<b>Referenser</b> .....	<b>51</b>
<b>8</b>	<b>Bilagor</b> .....	<b>55</b>
	<i>Bilaga A – Intervjufrågor</i> .....	55
	<i>Bilaga B – Respondent 1, transkribering</i> .....	57
	<i>Bilaga C – Respondent 2, transkribering</i> .....	69
	<i>Bilaga D – Respondent 3, transkribering</i> .....	80
	<i>Bilaga E – Respondent 4, transkribering</i> .....	87
	<i>Bilaga F – Respondent 5, transkribering</i> .....	95
	<i>Bilaga G – Respondent 6, transkribering</i> .....	103

## Tabeller

Tabell 2.1: Modell för centrala undersökningsområden .....	14
Tabell 3.1: Översikt val av respondenter och intervjuer .....	19
Tabell 3.2: Introduktionsfrågor .....	20
Tabell 3.3: Inledande intervjufrågor tillhörande intern systemdokumentation.....	21
Tabell 3.4: Intervjufrågor tillhörande faktorn uppdatering.....	22
Tabell 3.5: Intervjufrågor tillhörande faktorn dokumentationskvalitet .....	23
Tabell 3.5: Intervjufrågor tillhörande faktorn tillgänglighet.....	24
Tabell 3.7: Avslutande frågor. ....	24
Tabell 3.8: Ämneskategori .....	25

## Definitionslista

Begrepp	Definition
Systemdokumentation	Systemdokumentation är all den dokumentation, både riktat mot användare och utvecklare, som beskriver vad ett system gör, dess funktionalitet samt hur det borde användas för att uppnå uttänkt nytta (Aghajani, Nagy, Vega-Márquez, Linares-Vásquez, Moreno, Bavota, & Lanza, 2019).
Intern systemdokumentation	I denna uppsats behandlas all dokumentation som berör systemutvecklingens livscykel, framförallt den dokumentation som är riktad mot utvecklare. Denna dokumentation har till syfte att hjälpa systemutvecklare i skapande och underhållsaktiviteter (Aghajani et al., 2019). Följaktligen kommer denna typ av dokumentation beskrivas som intern systemdokumentation. Således är användardokumentation inte en del av intern systemdokumentation.
Dokumentationsskuld	Dokumentationsskuld refererar till alla problem i den interna systemdokumentationen som hindrar en systemutvecklare från att uppdatera, skapa eller bibehålla ett system (Mendes, Farias, Mendonca, Soares, Kalinowski & Spínola, 2016). Dessa problem kan exempelvis vara att systemdokumentationen är utdaterad, ofullständig, otillräcklig eller helt avsaknads (Mendes et al., 2016).

# 1 Introduktion

Inom systemutveckling är förvaltning av kunskap en viktig process som påverkar varje flöde i mjukvarulivscykel (De Boer & Van Vliet, 2009; Rus, Lindvall & Sinha, 2002). Flödena är beroende av att ett utvecklingsteam fattar kompetenta beslut baserat på den kunskap som är nödvändig för att förstå systemets alla komponenter (Rus, Lindvall & Sinha, 2002). Det intellektuella kapitalet är således en mycket viktig tillgång som hjälper ett företag att utföra en mer effektiv systemutveckling (O'Connor & Ryan, 2012). Som en förespråkad artefakt för att hantera och förvalta denna kunskap, är därför intern systemdokumentation av avgörande betydelse (de Souza, Anquetil, & de Oliveira, 2005).

Arbetet och användandet av systemdokumentation väljs ofta bort, framförallt den interna systemdokumentationen som berör systemutvecklingens livscykel (Lethbridge, Singer & Forward, 2003; Forward & Lethbridge, 2002; Garousi, Garousi-Yusifoglu, Ruhe, Zhi, Moussavi, & Smith, 2014). Anledningarna är flera och mycket grundas i att det är ett tungt arbete som inte prioriteras, samt att utvecklare inte ser värdet i att lägga dyrbar tid på dokumentationsarbete (Lethbridge, Singer & Forward 2003). Den dyrbara tiden vill utvecklare istället lägga på arbete som anses mer värdefullt och som bidrar med desto mer självklar personlig nytta (Lethbridge, Singer & Forward, 2003). Ännu en bidragande faktor är även att ständigt dokumentationsarbete är en dyr aktivitet som är svår att underhålla (Cabrera & Cabrera, 2002; Sulaiman, Idris & Sahibuddin, 2002).

Det finns ett antal faktorer som påverkar utformning, användning och underhåll av intern systemdokumentation (Lethbridge, Singer & Forward, 2003) och tre av dessa faktorer ska undersökas vidare i studien. Den första av faktorerna berör uppdatering av dokumentationen och refererar till hur väl den tillhörande interna systemdokumentationen stämmer överens med systemets aktuella status (Lethbridge, Singer & Forward, 2003; Forward & Lethbridge, 2002; Hunt & Thomas, 2000). Den andra faktorn berör kvaliteten på innehållet i dokumentationen och behandlar hur enkelt det är för en läsare att förstå och ta till sig det som beskrivs i den interna systemdokumentationen, samt om all nödvändig information, som krävs för underhålls- och utvecklingsarbete, finns med (Plösch, Dautovic, & Saft, 2014; Garousi et al., 2014; Aghajani et al., 2019). Den sista faktorn är tillgänglighet och har en betydande roll för användandet av den interna systemdokumentationen eftersom det redogör för hur enkelt det är för mjukvaruutvecklare och andra i utvecklingsteamet att komma åt och hämta den interna systemdokumentationen (Zhi, Garousi-Yusifoglu, Sun, Garousi, Shahnewaz, & Ruhe, 2015).

När viktiga faktorer inom dokumentationsarbetet påvisas bristfälliga, kan det leda till att systemet inte uppdateras, underhålls eller används som tänkt (Mendes et al., 2016). En rad problem har uppdagats när dokumentationsarbetet tett sig otillräckligt och dessa kan sammanfattas som begreppet dokumentationsskuld (Mendes et al., 2016). Dokumentationsskuld innefattar de problem med systemdokumentationen som hindrar en utvecklare från att skapa, underhålla eller bibehålla ett system (Mendes et al., 2016). Otillräckligt arbete med dessa faktorer kan även leda till att utvecklare, istället för att använda dokumentationen, förlitar sig på den egna kompetensen och kunskapen kring att läsa och förstå kod (Stapel & Schneider, 2014). I en del sammanhang kan denna typ av systemutveckling fungera, men i det långa loppet anses däremot välskrivnen intern systemdokumentation ha stor betydelse för en lyckad systemutveckling, eftersom dokumentation förvaltar den kunskap som är nödvändig för den fulla förståelsen (Lethbridge, Singer & Forward, 2003; Stapel & Schneider, 2014).

## 1.1 Problemformulering

Under perfekta förhållanden har intern systemdokumentation en omfattande inverkan på systemutvecklingen, men i praktiken är det sedan länge känt att intern systemdokumentation inte används till den grad som är önskvärt (Lethbridge, Singer & Forward, 2003). De brister som påträffas handlar ofta om att dokumentationen varken är uppdaterad, uppnår kvalitetskraven (Lethbridge, Singer & Forward, 2003) eller är tillgänglig (de Graaf, Tang, Liang & van Vliet, 2014). Bristfällig intern systemdokumentation i form av ofullständiga dokument leder emellertid till bristande förtroende, förlorad kunskap och hindrad förståelsen för systemet (Lethbridge, Singer & Forward, 2003).

Med en begränsad tillgänglighet av systemdokumenten, minskar möjligheten till uppdateringsarbete men även möjligheten att ta del av kunskap kring systemet (Hunt & Thomas, 2000; Correia, 2008). När den interna systemdokumentationen inte är uppdaterad, försvinner trovärdigheten för innehållet och med en datumstämpel som är betydligt äldre än systemets aktuella status, blir utvecklarna tveksamma till om den faktiskt talar sanning (Lethbridge, Singer & Forward, 2003). Trovärdigheten hos systemdokumentationen påverkas även av dokumentationskvaliteten (Stapel & Schneider, 2014). Om innehållet är svårförstått påverkas motivationen till att ta sig an dokumentationen (Stapel & Schneider, 2014). Tvivel om trovärdigheten bidrar till att utvecklarna inte förses med den kunskap som krävs för att kunna underhålla systemet (Stapel & Schneider, 2014) och dokumentationen förlorar sin användbarhet (Briand, 2003), vilket sker i de flesta organisationer (Hunt & Thomas, 2000).

Det finns delade åsikter kring vilken av faktorerna som ter sig viktigast, samt till vilken grad det påverkar utformningen av den interna systemdokumentationen och systemutvecklingen. Många utvecklare ser att dokumentation, trots att den inte är uppdaterad till systemets aktuella status, ändå kan vara till användning eftersom den kan ge en övergripande förståelse för systemet och dess funktioner (Lethbridge, Singer & Forward, 2003; Forward & Lethbridge, 2002). Å andra sidan, pekar andra studier på att uppdaterad dokumentation är av stor vikt, eftersom det framförallt är nyexaminerade systemutvecklare som arbetar med underhåll av system och behöver mest vägledning i sitt utvecklingsarbete (Sulaiman, Idris & Sahibuddin, 2002; Stapel & Schneider, 2014; Selic, 2009). Följaktligen finns det en uppfattning hos systemutvecklare att systemdokumentation är användbar oberoende dess dokumentationskvalitet (Sulaiman, Idris & Sahibuddin, 2002). Andra studier påvisar dock att dokumentationskvalitet är essentiellt eftersom det bidrar till ökad förståelse och kunskap om systemets syfte, funktioner och komponenter (Plösch, Dautovic, & Saft, 2014; Garousi et al., 2014; Briand, 2003). Till hur stor del tillgängliga dokument påverkar dokumentationsarbetet är inte brett diskuterat i litteraturen, annat än att arbetet inte kan utföras ordentligt (Hunt & Thomas, 2000).

Tidigare studier har kommit fram till att bristfällig systemdokumentation kan leda till dokumentationsskuld, men endast att det uppstår när dokumenten inte är uppdaterade eller saknas helt, samt när de är ofullständiga eller otillräckliga (Soares, Alves, Mendes, Mendonça & Spínola, 2015; Campos, Valente & Terra, 2016; Guo, Spínola & Seaman, 2014). Vad som menas med otillräckliga dokument framgår inte tydligt och det lyfts inte heller hur olika faktorer leder till dokumentationsskuld.

Med dessa delade åsikter, den mindre utforskade faktorn tillgänglighet samt dokumentationsskuld som grund, är studiens ändamål att undersöka faktorerna uppdatering, dokumentationskvalitet och fullständighet inom intern systemdokumentation med målet att



klarlägga om och hur dessa kan leda till ogynnsamma påföljder som dokumentationsskuld. Med ursprung i de valda faktorerna ska även konsekvenser av dokumentationsskuld undersökas.

## 1.2 Forskningsfråga

Problemformuleringen resulterar i följande forskningsfråga:

*Hur kan faktorerna uppdatering, dokumentationskvalitet och tillgänglighet inom intern systemdokumentation leda till ogynnsamma påföljder som dokumentationsskuld? Vilka är de främsta konsekvenserna som dokumentationsskuld leder till?*

## 1.3 Syfte

Syftet med studien är att undersöka hur konsekvenser, som dokumentationsskuld, kan uppstå när faktorerna uppdatering, dokumentationskvalitet och tillgänglighet av intern systemdokumentation ter sig otillräckliga. Om dokumentationsskuld skapas är det även av intresse att studera vilka konsekvenser det kan leda till för systemet, individen och organisationen.

## 1.4 Avgränsningar

Studien är avgränsad till att intervjua personer vars arbetsbeskrivning är eller har varit systemutvecklare. Ytterligare en avgränsning har gjorts till att endast intervjua personer på IT-genomgående företag.

Följaktligen, varje gång orden 'systemdokumentation' eller 'dokumentation' nämns i uppsatsen åsyftas intern systemdokumentation.

## 2 Tidigare forskning

*Tidigare forskning beskriver den litteratur och forskning som tagits upp för att, tillsammans med empirin, besvara uppsatsens forskningsfråga. Inledningsvis diskuteras begreppen kunskapshantering och kunskapsdelning. Följaktligen beskrivs definition, användningsområde samt typer av systemdokumentation. Detta leder ner till en specifik förklaring av intern systemdokumentation. Där efter tas olika faktorer inom intern systemdokumentation upp, samt dokumentationsskuld. Slutligen sammanfattas centrala begrepp följande av en modell över nämnda begrepp.*

### 2.1 Kunskapshantering

*Kunskapshantering* är en strategi som skapar, behandlar och delar kunskap med organisatorisk prestation som mål (Rus, Lindvall & Sinha, 2002). Hanteringen av kunskap är en livsviktig strategi för att säkerställa att kunskap skapas och utnyttjas fullt ut inom det egna företaget. Det beror på att kunskap, i kombination med kompetens, tankar, innovation och idéer, skapar en konkurrenskraftig och effektiv organisation (Dalkir, 2005). Kunskapshantering inom systemutvecklingsföretag strävar efter att omvandla data till information och information till kunskap (Rus, Lindvall & Sinha, 2002).

System- och mjukvaruutvecklande företag förlitar sig på det intellektuella kapitalet vid skapande och utveckling av sina produkter. Detta kapital är en av de främsta och viktigaste tillgångarna ett företag kan hantera och förvalta, eftersom det bidrar till en effektivare produktutveckling (O'Connor & Ryan, 2012). Kunskapen inom ett team påverkar produktens alla processer genom hela dess livscykel. Således är processerna beroende av att teammedlemmarna fattar rätt beslut vid rätt tillfälle, baserat på ny eller redan befintlig kunskap (Rus, Lindvall & Sinha, 2002). I mindre projektteam influeras denna kunskap av egna tankar och idéer, men ju mer ett projekt växer desto viktigare blir det med aktiv kunskapsdelning och samordning inom projektteamet (Rus, Lindvall & Sinha, 2002). *Kunskapshantering* är således grunden till att arbete inom mjukvaruutvecklingsteam fokuserar på kollektiv kunskap snarare än den enskilde individens kunskap.

För att inom kunskapshantering främja arbetet med att skapa, behandla och dela kunskap och information inom organisationen, kan systemdokumentation användas (Forward & Lethbridge, 2002). Kunskapshantering är stort begrepp som kan delas in i flera mindre områden, där bland, kunskapsdelning (Henriks, 1999).

#### 2.1.1 Kunskapsdelning

En av dem mest kritiska komponenterna inom *kunskapshantering* är delandet av intern kunskap (Henriks, 1999). Delning av kunskap och information skapar en länk mellan den enskilda individen och den stora organisationen (Henriks, 1999). Individen besitter den konkurrenskraftiga kunskapen och organisationen förvaltar denna kunskap genom att utvinna dess tänkta värde (Henriks, 1999). För att dela kunskap inom en organisation krävs det att det finns en relation mellan minst två involverade parter, en som besitter kunskap, och är villig att dela med sig, samt en som ska erhålla kunskapen (Henriks, 1999).

Inom systemutveckling anses kunskap vara en av de viktigaste konkurrenspelarna. I utvecklingsstadiet kan kunskap delas muntligt mellan kollegor i informella situationer, men den mesta kunskapen inom ett utvecklingsteam delas med hjälp av systemdokumentation (de Boer & Van Vliet, 2009).

För att bevara och sprida kunskap inom organisationen är systemdokumentation en artefakt som kan komma till användning och om dokumentationsarbetet inte utförs ordentligt, kan kunskap komma att gå förlorad (Lethbridge, Singer & Forward, 2003; Stapel & Schneider, 2014). Däremot är det många som inte ser värdet i att dela kunskapen de besitter, eftersom de på ett personligt plan inte får någon större vinst utav det (Cabrera & Cabrera, 2002). Självklart finns det mycket värde att vinna på att aktivt dela kunskap, som status inom företaget och hos ledningen, men även en bredare kunskap inom företaget (Cabrera & Cabrera, 2002). Däremot finns det ett stort motstånd till att skapa och underhålla systemdokumentation (Cabrera & Cabrera, 2002). Arbetet ses som kostsamt och utvecklare lägger hellre tiden på arbete som kan anses vara mer värdefullt (Cabrera & Cabrera, 2002).

Eftersom kunskapsdelning är ett paraplybegrepp som innefattar systemdokumentation, kommer uppsatsen fortsatt att behandla systemdokumentation som artefakter för förvaltning och delning av kunskap.

## 2.2 Systemdokumentation

Som ett av de äldsta rekommenderade sätten att hantera och förvalta kunskap, har systemdokumentation en vital roll i systemutveckling (de Souza, Anquetil, & de Oliveira, 2005). Systemdokumentation är nämligen en nedskrivna artefakt som ska stödja skapandet, underhållet och användandet av systemet (Ding, Liang, Tang, & Van Vliet, 2014). Dokumentationen ska vara sanningsenligt på så sätt att det klart och tydligt beskriver systemet utan att ge tvivel om vad som menas (Parnas, 2011). Ordet dokument kan användas väldigt generellt då det refererar till all information, felaktig likväl som ofullständig, som följer med systemet (Parnas, 2011). Med två målgrupper i fokus, har systemdokumentation däremot skiftande syften och delas därför följande in i två olika begrepp.

*Användardokumentation* är systemdokumentation i form av manualer som underlättar förståelsen för hur systemet ska användas och är riktad till användaren (Sommerville, 2011).

*Intern systemdokumentation* innefattar de artefakter som berör mjukvarulivscykeln (Garousi et al., 2014) och är framförallt ägnad åt systemutvecklaren (Sommerville, 2011). Den här typen av dokumentation ska redogöra för systemets uppbyggnad och funktionalitet med målet att förenkla vidareutvecklingen av system för framtida systemutvecklare (Sommerville, 2011). Det ska även kommunicera information och kunskap om det givna systemet (Aghajani et al., 2019). Givet vår forskningsfråga är intern systemdokumentation all form av systemdokumentation bortsett från användardokumentation.

### 2.2.1 Intern systemdokumentation

Intern systemdokumentation behandlar krav, design, implementation, test och underhåll av system (Zhi, et al., 2015). Syftet med systemdokumentation är att hjälpa företagets systemutvecklare att förbättra utvecklingskurvan, samt underlätta för framtida underhåll av systemet (Garousi et al., 2014). Intern systemdokumentation är inte ett bestämt dokument, utan snarare all dokumentation som behandlar de många olika funktioner, moduler och artefakter som skapas under hela systemutvecklingens livscykel (Garousi et al., 2014). Tanken med alla existerande interna dokument, som relaterar till ett system, är att det ska hjälpa utvecklarna förstå sig på systemet och bidra till att alla uppgifter kring systemet sker så effektivt och sömlöst som möjligt (Garousi et al., 2014).

Intern systemdokumentation prioriteras bort av utvecklare på grund av att det är för kostsamt och tidskrävande (Forward & Lethbridge, 2002), trots det blir behovet av välutvecklad dokumentation större och större ju mer tekniken går framåt och utvecklas (Selic, 2009). System blir ständigt mer komplexa och eftersom det framförallt är juniora systemutvecklare som tilldelas underhållsarbetet krävs det en välskriven intern systemdokumentation för att arbetet ska hålla hög nivå (Selic, 2009).

Fortsättningsvis kommer studien att fokusera på tre faktorer inom intern systemdokumentation, som sällan ter sig tillräckliga. Dessa faktorer är att systemdokumentationen ska vara *uppdaterad*, ha hög och rätt *dokumentationskvalitet* samt vara *tillgänglig*.

## 2.3 Faktorer som påverkar intern systemdokumentation

Inom intern systemdokumentation finns det ett antal faktorer som påverkar arbete, skapande, underhållning och utveckling av intern systemdokumentation. Kapitlet kommer diskutera tre valda faktorer, dess problematik, samt konsekvenser.

### 2.3.1 Uppdatering

Intern systemdokumentation är generellt aldrig uppdaterad i jämförelse med systemets aktuella status (Lethbridge, Singer & Forward, 2003; Forward & Lethbridge, 2002; Hunt & Thomas, 2000). Det beror bland annat på att underhållet av dokumentation inte håller måttet samt att när uppdateringar väl sker, är det flera veckor efter att koden i sig har ändrats (Lethbridge, Singer & Forward, 2003). För att se till att dokumentation uppdateras, är det optimalt att gå in med tanken om att dokumentation ska produceras så fort kod produceras och likaså ska systemdokument uppdateras så fort ändringar sker i koden (Hunt & Thomas, 2000). Det är däremot betydelsefullt att fundera kring varför dokumentationen behöver uppdateras eftersom uppdatering bör ske när det bringar systemutvecklarna nytta och inte bara för att datumstämpeln är gammal (Forward, 2002). Nyttas skapas när utvecklarna har behov av att uppdateringen sker och kommer ha användning av uppdateringen (Forward, 2002).

En annan aspekt kring varför intern systemdokumentation inte uppdateras grundas i värdet hos dokumentationen. Utvecklarna prioriterar framförallt den dokumentation som är av värde för dem, det vill säga enkel dokumentation som resulterar i relativt lite underhållsarbete (Lethbridge, Singer & Forward, 2003). Tyngre, mer komplicerad och tidskrävande dokumentation väljer de däremot att bortse från. När utvecklarna tvingas att skapa, uppdatera

och använda sig av den tyngre dokumentationen, blir de motstridiga (Lethbridge, Singer & Forward, 2003). För att motivera utvecklarna till uppdateringsarbete, behöver ledningen istället sätta standarder för en simplare dokumentation med högre värde (Lethbridge, Singer & Forward, 2003).

Uppdateringsproblematiken bidrar till bristande förtroende för intern systemdokumentation (Lethbridge, Singer & Forward, 2003). Dokumentationens trovärdighet beror ofta på om de ändringar som gjorts i systemet sedan den senaste uppdateringen av dokumentationen har varit stora eller små. Små kodändringar leder oftare till att dokumentationen faktiskt uppdateras, medan större ändringar snarare bidrar till sämre uppdaterad dokumentation (Lethbridge, Singer & Forward, 2003). Här finns det dock ett delat tycke, då en del systemutvecklare snarare menar att mindre kodändringar är onödiga att uppdatera i dokumentationen och eftersom småändringar sker hela tiden, betyder det att dokumentationen ständigt behöver uppdateras för att möta systemet (Sulaiman, Idris & Sahibuddin, 2002). I en del systemdokumentation krävs det dessutom att större delar, alternativt hela dokumentationen, behöver uppdateras av endast en liten kodändring. De ständiga kodändringarna med dokumentationsuppdatering som följd, kan därför anses som kostsamt och tidskrävande (Sulaiman, Idris & Sahibuddin, 2002).

Trots omständigheterna tycker många utvecklare att föråldrad intern systemdokumentation kan vara till nytta eftersom det kan ge en övergripande förståelse för komponenternas funktionalitet (Lethbridge, Singer & Forward, 2003; Forward & Lethbridge, 2002). Det finns däremot en grupp av utvecklarna som drar stor nytta av systemdokumentationen är uppdaterad, nämligen dem som står för underhållsarbetet av systemet (Sulaiman, Idris & Sahibuddin, 2002). Dessa personer sitter oftast på mindre kunskap om projektet än de personer som är med i utvecklingsarbetet. Till följd av dåligt uppdaterad systemdokumentation går kunskap förlorad om systemets syfte, frustration hos utvecklarna byggs upp och underhållsarbetet kan inte utföras ordentligt (Stapel & Schneider, 2014). Följaktligen bidrar även icke-uppdaterat systemdokumentation till att utvecklare förlitar sig på personliga konversationer för att delge kunskap (Tang, Liang & van Vliet, 2011). Problemet med samtal, snarare än dokumentation, är att konversationer kan te sig ineffektiva och fulla av felaktigheter (Tang, Liang & van Vliet, 2011). Kunskapen bör komma från personen som sitter på kunskapskällan och inte genom återberättelser, vilket betyder att det måste gå att identifiera den person som besitter kunskapen, att hen är villig att dela med sig av sin kunskap, samt kan delge sin kunskap på ett tydligt sätt (Tang, Liang & van Vliet, 2011).

### 2.3.2 Dokumentationskvalitet

En del författare anser att dokumentationskvalitet är en av de viktigare aspekterna vid utformning av systemdokumentation (Garousi et al., 2014; Plösch, Dautovic, & Saft, 2014), då det bidrar till ökad förståelse och kunskap om systemet i dess utveckling-, användning- och uppdateringsstadier (Briand, 2003). Om ett företag lyckas med att förbättra sin dokumentationskvalitet finns det goda förutsättningar för att förbättra systemet (Ding et al., 2013). Trots att dokumentationskvalitet tydligt uppvisar sin vikt inom systemutveckling uppfyller det sällan en systemutvecklarens förväntningar (Plösch, Dautovic, & Saft, 2014). Dokumentationskvalitet som inte uppfyller kartlagda krav kan komma att resultera i lägre effektivitet, systemdefekter och bidra till lägre systemkvalitet (De Graaf, 2011). Dock anser vissa författare att systemdokumentation kan te sig användbar oberoende av felaktigheterna i kvaliteten (Sulaiman, Idris & Sahibuddin, 2002).

Dokumentationskvalitet är ett brett ämne som består av många olika typer av kvalitetsattribut. År 2014 utförde Plösch, Dautovic, & Saft en empirisk studie med syfte att försöka klargöra vilka kvalitetsattribut som ansågs allra viktigast för olika typer av systemutvecklare. 88 personer avslutade undersökningen. Ur deras svar framkom sex kvalitetsattribut; korrekthet, klarhet, läsbarhet, struktur, begriplighet och konsekvent (Plösch, Dautovic, & Saft, 2014). Även Garousi et al. (2014) tog sig an ämnet och påstår i sin artikel att intern systemdokumentation består av kvalitetsattribut såsom korrekthet, fullständighet, läsbarhet och noggrannhet. Aghajani et al. gjorde 2019 en stor undersökning för att försöka förstå den dokumentationsproblematik som utvecklare utsätts för. Genom automatiska sökningar på relevanta problemord samlades data in från fyra olika källor: Stack Overflow, email, nerhämtningsförfrågningar och problemtrådar på GitHub (Aghajani et al., 2019). Resultatet presenterades i modeller tillsammans med förklarande paragrafer för varje specifikt problemområde. Undersökarna kom fram till att de huvudsakliga problemområdena för kvalitet inom intern systemdokumentation är fullständighet, korrekthet, läsbarhet, användbarhet och underhållbarhet (Aghajani et al., 2019). Eftersom det inte finns utrymme i uppsatsen att ta upp alla kvalitetsattribut har en avgränsning gjorts och de kvalitetsattribut som författarna är ense om kommer diskuteras vidare. Fortsatt kommer kvalitetsattributen korrekthet, läsbarhet, struktur och fullständighet att studeras.

*Korrekthet* är en av de kvalitetsattribut som väger tyngst och behandlar rätt information i enlighet med existerande fakta (Aghajani et al., 2019; Plösch, Dautovic, & Saft, 2014). Exempel på problemområden inom korrekthet är felaktiga översättningar, kommentarer och programmeringsguider (Aghajani et al., 2019). Systemdokumentation som bygger på dessa fel kan innebära att utvecklare försöker replikera felaktig kod eller följer fel steg, vilket kan resultera i ödslad tid (Aghajani et al., 2019).

*Läsbarhet* berör vikten av korrekt stavning, grammatik, namnkonventioner och språkuttryck (Aghajani et al., 2019; Plösch, Dautovic, & Saft, 2014). Trots bristfällig läsbarhet, kan utvecklare fortfarande dra nytta av systemdokumentation. Skillnaden blir däremot att det tar längre tid än nödvändigt (Aghajani et al., 2019).

Ofta har ett företag ingen definierad *struktur* för vad systemdokumentation ska innehålla och vilka utformningsregler som ska efterhållas (Briand, 2003). Detta bidrar till att systemdokumentation många gånger är svår för utvecklare att förstå, vilket i sin tur leder till att den är besvärlig att följa (Briand, 2003).

*Fullständighet* behandlar färdigställd systemdokumentation, men ska inte blandas ihop med uppdaterad dokumentation. Fullständig, eller färdigställd, systemdokumentation innehåller all den relevanta information som är nödvändiga för att förstå, underhålla och använda sig av det utvecklade systemet (de Souza, Anquetil, & de Oliveira, 2005). Om nödvändig information saknas tolkas systemdokumentationen som icke-färdigställd och kommer inte uppnå sitt tänka syfte, samt blir oanvändbart i många användarscenarion (de Souza, Anquetil, & de Oliveira, 2005). Trots att det finns mycket som talar för att företag ska fokusera tid och resurser på att färdigställa dokumentation har det blivit något av en norm för utvecklare att använda sig av ofullständig intern systemdokumentation i sitt dagliga arbete (Briand, 2003). Vid större kodändringar är det desto mer lönsamt att använda färdigställd dokumentation (Falessi, Briand, Cantone, Capilla & Kruchten, 2013). Däremot anses det inte rimligt för det flesta företag att försöka färdigställa all dokumentation eftersom ständigt skapande och underhåll av systemdokumentation bidrar till stora kostnader (Garousi et al., 2014). Istället borde

systemdokumentation skapas till den bredd att komplexa funktioner och aktiviteter som behöver utvecklas, får stöttning nog att bli förstådda (Garousi et al., 2014).

### 2.3.3 Tillgänglighet

Tillgänglighet beskriver hur enkelt det är för mjukvaruutvecklare och andra användare att komma åt och hämta ner systemdokumentation (Zhi et al., 2015). När kunskap går att hitta och hämta på ett snabbt och effektivt sätt sparas dyrbar tid och eventuella fel reduceras, vilket i sin tur kommer leda till bättre systemkvalitet (de Graaf et al., 2014). Begränsad tillgänglighet kommer därav leda till att systemet inte utformas som överenskommet och kommer således kosta företaget desto mer pengar (de Graaf et al., 2014). Tillgänglighet av systemdokumentation innefattar även hurvida det finns dokumentation som hjälper utvecklarna att utföra sitt arbete och om dessa dokument innehåller fullständig information, som är presenterat på ett sätt som går att förstå (Matulevicius, Kamseu, & Habra 2009).

Intern systemdokumentation som är fysiskt svårtillgänglig kan även leda till att dokumentationen inte uppdateras, eftersom utvecklarna varken har intresse eller energi att leta efter den (Hunt & Thomas, 2000). Som följd, går kunskap förlorad (Correia, 2008). För att undkomma detta problem och istället motivera till uppdateringsarbete och kunskapsdelning, bör dokumentationen vara belägen lättillgängligt, exempelvis online eller i ett dokumenthanteringssystem (Hunt & Thomas, 2000).

Många företag använder någon form av dokumenthanteringssystem för att automatiskt importera, kategorisera, lagra, söka, hämta ner och arkivera sin interna systemdokumentation (Zhi et al., 2015). Trots detta dokumenteras fortfarande en stor mängd personlig kunskap och information i egna dokument, anteckningar, email, serverar et cetera (Tang, Liang & van Vliet, 2011). Denna typ av kunskap delas således inte vidare och bidrar till att kunskapen om moduler, system och funktioner blir osammanhängande, ofullständig och vitt spridda (Tang, Liang & van Vliet, 2011).

Eftersom det finns en risk att olika projekt eller utvecklare arbetar med samma system och dess dokumentation samtidigt, blir det essentiellt att rätt version av systemdokumentation är tillgänglig i dokumenthanteringssystemet (Sommerville, 2001). Om olika versioner av systemdokumentationen finns tillgänglig kan detta bidra till att utvecklare gör enkla misstag, eftersom de inte arbetar med rätt version av dokumentet (Sommerville, 2001).

## 2.4 Dokumentationsskuld

Problem kring intern systemdokumentation kan leda till en rad konsekvenser, denna uppsats åsyftar däremot att främst behandla dokumentationsskuld. Dokumentationsskuld är en typ av teknisk skuld och refererar till alla problem i systemdokumentationen som hindrar en utvecklare från att underhålla, skapa eller bibehålla ett system (Mendes et al., 2016).

Faktorer inom intern systemdokumentation, som kan resultera i dokumentationsskuld, är utdaterad, ofullständig, avsaknad och otillräcklig dokumentation (Guo & Seaman, 2011; Mendes et al., 2016; Rios, de Mendonça & Spínola, 2018). Adekvata förklaringar till hur dessa faktorer leder till dokumentationsskuld finns inte att hitta i mycket av litteraturen, endast uppdatering förklaras enkelt. Guo och Seaman (2011) anser att om ett system, dess funktioner

eller moduler, ändras bör även den tillhörande systemdokumentationen uppdateras jämsides, annars kan detta leda till dokumentationsskuld. Däremot, om ett system, modul eller funktion skapas med intentionen att det aldrig kommer uppdateras, återskapas eller användas i framtiden är det inte heller nödvändigt att vare sig skapa eller uppdatera dess tillhörande systemdokumentation (Guo & Seaman, 2011). Detta kommer spara dyrbar tid för utvecklaren och samtidigt inte resultera i onödiga problem längre fram i arbetsprocessen (Guo & Seaman, 2011). Dock är det svårt för utvecklare och andra involverade parter att alltid veta om en artefakt i systemet kommer behöva modifieras längre fram i tiden (Guo & Seaman, 2011). Därav är det viktigt för ett projekt att kunna göra informerade avvägningar kring dem systemdokument som skapas, samt kunna förutse och hantera eventuella risker (Guo & Seaman, 2011).

Följaktligen finns det ingen konkret förklaring till hur ofullständig eller otillräcklig intern systemdokumentation resulterar i dokumentationsskuld. Det påvisas till och med att faktorerna inte nödvändigtvis kommer hindra användandet av systemdokumentationen (Alves, Ribeiro, Caires, Mendes, & Spínola, 2014). Däremot kan användandet av systemdokumentation med bristfällig kvalitet bidra till att den systemkvaliteten, som vill uppnås inom projektet, kommer bli lidande (Alves et al., 2014).

Slutligen, i många projekt är det en normalitet att producera lite skuld under projektidens gång, eftersom detta kan öka den interna produktiviteten (Guo & Seaman, 2011). Riskerna är dock höga och om skulden inte planeras att betalas av i framtiden kommer koden och systemet slutligen bli svårare att underhålla (Guo & Seaman, 2011). När det finns för mycket otillräcklig eller utdaterad systemdokumentation kommer detta även tvinga utvecklare att spendera dyrbar tid på att bekanta sig med utvecklingsspråket, verktygen, funktionerna eller det specifika problemet (Tom, Aurum & Vidgen, 2013).

## 2.5 Sammanfattning av centrala undersökningsområden

Syftet med studien är kartlägga hur faktorer inom intern systemdokumentation kan leda till ogynnsamma påföljder som dokumentationsskuld, samt vilka konsekvenser som dokumentationsskuld kan resultera i. Intern systemdokumentation är en artefakt som används för att bevara kunskap inom organisationen, samt ska redogöra för systemets uppbyggnad och funktionalitet. Det innefattar all dokumentation som berör livscykeln för mjukvaruutveckling, bortsett från användardokumentation.

De faktorer som undersöks är uppdatering av dokumentationen, dokumentationskvaliteten samt tillgängligheten av dokumenten. Det finns delade åsikter kring hur väl intern systemdokumentation måste vara uppdaterad för att kunskap och förståelse ska kunna bevaras. En del utvecklare tycker att den inte behöver vara uppdaterad eftersom den förståelse som krävs går att utvinna även om den inte är uppdaterad enligt systemets aktuella status. Andra anser däremot att fullt uppdaterad dokumentation är mycket viktigt, och det framförallt för att nyexaminerade systemutvecklare med mindre kunskap och erfarenhet ska kunna ta sig an systemet med rätt förutsättningar.

Dokumentationskvaliteten anser de flesta vara viktig för att kunna få full förståelse för innehållet, men det finns även ett antal som tycker att intern systemdokumentation kan användas oberoende kvaliteten. Dokumentationskvaliteten är uppbyggd av olika kvalitetsattribut som kan te sig olika viktiga och därför påverka kvaliteten i det hela på olika sätt beroende på systemutvecklarens personliga tyckande.



Tillgängligheten av intern systemdokumentation påverkar framförallt om dokumenten fortsättningsvis kommer att uppdateras eller inte. Med begränsad tillgänglighet kan utvecklare inte ta del av den kunskap som krävs för att kunna arbeta med systemen.

Svårigheter med uppdatering, kvalitet och tillgänglighet kan leda till kommunikationsproblematik där fel innehåll och svårigheter med förståelsen bidrar till att kunskap om systemet samt dess dokumentationen, går förlorad. När systemdokumentationen inte går att förstå och ta sig an, är den inte heller användbar till den nivå som krävs. Denna typ av bristfällig intern systemdokumentation kan leda till ogynnsamma påföljder, möjligen dokumentationsskuld. Dokumentationsskuld refererar till de felaktigheter i intern systemdokumentation som hindrar utvecklare från att kunna arbeta ordentligt med systemen. Några faktorer har påvisats i litteratur leda till dokumentationsskuld, men på vilket sätt är inget som diskuteras.

## 2.6 Modell för centrala undersökningsområden

Nedanstående tabell är den modell som behandlar all insamlad litteratur och delar in underlaget i olika kategorier, som kommer undersökas senare i uppsatsen. Syftet med dessa kategorier, samt dess undersökningsområde, är att underlätta för skapandet av intervjufrågor och säkerställa att struktur upprätthålls. Den är också ett hjälpmedel som ska garantera att forskningsfrågan besvaras.

Huvudkategori	Undersökningsområde	Förklaring	Tillhörande litteratur
<b>Intern systemdokumentation</b>	Användande av systemdokumentation	Hjälpa utvecklarna förstå systemet och bidra till att alla uppgifter kring systemet sker så effektivt och sömlöst som möjligt.	Garousi et al. (2014), Sommerville (2011)
	Syfte	Förmedla information och kunskap kring uppbyggnad och funktionalitet med målet att förenkla utveckling och underhåll för framtida utvecklare	Garousi et al. (2014)
	Problematik	Kostsamt och tidskrävande	Cabrera & Cabrera (2002), Forward & Lethbridge (2002)
	Kunskapsdelning	Intern systemdokumentation som artefakt för att förvalta, hantera och sprida kunskap.	Lethbridge, Singer & Forward (2003), Stapel & Schneider (2014)

<b>Uppdatering</b>	Anledning till bristfälligt uppdateringsarbete	Motvillighet från utvecklare då de inte ser nyttan.	Lethbridge, Singer & Forward (2003)
	Uppdatering beroende på kodändringens storlek	Små vs. stora ändringar i koden.	Lethbridge, Singer & Forward (2003), Sulaiman, Idris & Sahibuddin(2002)
	Användande av föråldrar dokumentation	Kan vara till nytta då det kan ge en övergripande förståelse.	Forward & Lethbridge (2002), Forward (2002)
	Utdaterade systemdokuments påverkan på kunskapsdelning	Icke-uppdaterade dokument leder till att utvecklare delar kunskap genom konversationer	Tang, Liang & van Vliet (2011)
	Konsekvenser av dåligt uppdaterade interna systemdokument	Kunskap går förlorad om systemets syfte, frustration hos utvecklarna byggs upp och underhållsarbetet kan inte utföras ordentligt.	Stapel & Schneider (2014), Tang, Liang & van Vliet (2011)
<b>Dokumentationskvalitet</b>	Vikten av olika kvalitetsattribut	Fyra kvalitetsattribut som är viktiga för utformning.	Aghajani et al. (2019), Plösch, Dautovic, & Saft (2014)
	Korrekthet	Rätt och korrekt innehåll.	Aghajani et al. (2019), Plösch, Dautovic, & Saft (2014)
	Läsbarhet	Grammatik, stavning, språkuttryck et cetera.	Aghajani et al. (2019), Plösch, Dautovic, & Saft (2014)
	Struktur	Innehåll och utformningsstandard.	Briand (2003)
	Fullständighet	Fullständigt innehåll. Utvecklare använder ofullständiga dokument.	Briand (2003), de Souza, Anquetil, & de Oliveira (2005)
	Konsekvenser av dålig dokumentationskvalitet	Lägre effektivitet, systemfel samt lägre systemkvalitet.  Kan te sig användbar oberoende av felaktigheterna i olika kvalitetsattribut.	de Graaf (2011)  Hunt & Thomas (2000), Sulaiman, Idris & Sahibuddin (2002)

	Dokumentationskvalitetens påverkan på kunskapsdelning	Bidrar till ökad förståelse och kunskap om systemet i dess utveckling-, användning- och uppdateringsstadier.	Briand (2003)
<b>Tillgänglighet</b>	Åtkomstrestriktioner	Systemutvecklare sparar dokument på privata platser.  Dokumenthanteringssystem.	Tang, Liang & van Vliet (2011)  Zhi et al. (2015)
	Versionshantering	Vikten av rätt version, då utvecklare kan arbeta med system och dess dokumentation samtidigt som varandra.	Sommerville (2001)
	Hur tillgänglighet påverkar kunskapsdelning	Sparar in dyrbar tid och eventuella fel reduceras, vilket i sin tur kommer leda till bättre systemkvalitet.	de Graaf et al. (2014)
	Konsekvenser av otillgänglig intern systemdokumentation	Systemet utformas inte som överenskommet och kommer således kosta företaget desto mer pengar.  Kan leda till att systemdokumentation inte uppdateras.	de Graaf et al. (2014)  Hunt & Thomas (2000)
<b>Dokumentations-skuld</b>	Definition	Alla problem som hindrar en utvecklare från att arbeta med systemet.	Mendes et al. (2016)
	Faktorer som kan resultera i dokumentationsskuld	Utdaterad, ofullständig, avsaknad och otillräcklig dokumentation	Guo & Seaman (2011), Mendes et al. (2016)

Tabell 2.1: Modell för centrala undersökningsområden

## 3 Metod

*Den insamlade data som uppsatsen grundar sig på består dels av omfattade textanalyser av vetenskapliga artiklar dels av kvalitativa undersökningsintervjuer med olika typer av systemutvecklare. I metodkapitlet redogörs för hur det empiriska materialet insamlats och bearbetats, samt vilka etiska frågor som är relevanta för studien.*

### 3.1 Val av undersökningsmetod

Den undersökningsmetod som vi valde att använda oss av var den *kvalitativa* undersökningsmetoden. Detta eftersom denna metod riktar in sig på all sorts datainsamling som inte är numerisk – nämligen ord, intryck, åsikter och ljud (Oates, 2005). Då vi var intresserade av data som beskriver ett ämne utefter en människas tankesätt, attityd och erfarenheter ansåg vi denna metod vara den lämpligaste (Oates, 2005).

Vi ansåg även att den kvalitativa metoden var mest lämpad för vår studie, eftersom tre av dess ledande ord, *öppenhet, nyanserat och flexibilitet* (Jacobsen, 2002) är viktiga för oss. Eftersom vi undersökte ett område som ter sig tämligen olika på arbetsplatser, med olika arbetssätt och inställning till arbete, var det viktigt att respondenterna kände *öppenhet* till att våga tala ut om hur de faktiskt tänkte och tyckte. Uttrycket *öppenhet* betyder att vi, som intervjuare, endast till viss del hade bestämt vad vi letade efter, utan bestämda frågor och svar. Respondenten fick istället ta frågorna dit hen kände. I slutändan kan den här typen av upplägg av intervjuerna leda till en bredare förståelse för fenomenet, i vårt fall intern systemdokumentation (Jacobsen, 2002).

När respondenten fick lösa tyglar och kunde förklara mer ingående kring hur hen kände och tyckte, blev det lättare för oss att utforma *nyanserade* intervjuer, där varje intervju blev anpassad efter respondenternas olika åsikter. Det var även på grund av det här som vi ansåg att respondenter med olika åsikter skulle ge en mer nyanserad bild av intern systemdokumentation. Det nyanserade perspektivet åsyftar att ta fram de unika synsätten hos respondenten (Jacobsen, 2002).

Eftersom studiens område var nytt och inget vi hade en bredare kunskap om, var det även viktigt för oss att ha möjligheten att tänka om och ändra perspektiv senare under studiens gång. Exempelvis, efter dem två första intervjuerna, förstod vi att respondenterna gärna ville vara helt anonyma, detta eftersom känslig information presenterades. Således bestämde vi att alla intervjuobjekt fick vara anonyma, då information som företagsnamn ändå inte var av större vikt för uppsatsen. Denna typ av *flexibilitet* gör att det sedan tidigare bestämda problemområdets omfång kan komma att ändras efter respondenternas svar och perspektiv (Jacobsen, 2002).

Vi såg värdet i att väga fördelar och nackdelar mot varandra, eftersom kvalitativa data kan vara svårare att analysera då den ter sig olika med varje individ som intervjuas. Den data som samlades in är komplex med många, ostrukturerade ord, vilket leder till en tidskrävande analys av data (Jacobsen, 2002). Däremot ansåg vi att det tidskrävande arbetet var nödvändigt för vår typ av studie eftersom vi inte såg den vara statisk mätbar. Vi var intresserade av respondenternas egna åsikter, synpunkter och upplevelser när de arbetar eller har arbetat med intern systemdokumentation. För att åstadkomma klarhet i hur människor ser på en given

omständighet, ska en kvalitativ metod användas (Jacobsen, 2002) och därför kändes det som ett självklart val.

Ytterligare en omständighet att ha i åtanke är att en kvalitativ studie på kandidatnivå resulterar i endast ett fåtal intervjuer. Det har givit oss ett avsmalnat perspektiv, men samtidigt är vi ute efter att klarlägga fenomenet intern systemdokumentation med unika åsikter, och inte genom generalisering.

## 3.2 Insamling av data

*I detta avsnitt beskrivs tillvägagångssätt av metod för utveckling av teoretiskt ramverk, val av organisationer och val av respondenter. Dessa val har baserats på den information som presenterats i stycket 2.1 Tidigare forskning.*

### 3.2.1 Metod för utveckling av teoretiskt ramverk

Eftersom uppsatsens giltighet är starkt kopplad till valet av källor (Jacobsen, 2002) använde vi oss av säkra, rekommenderade källhanteringsverktyg som LUBsearch och Google Scholar. Med hjälp av dessa verktyg fick vi tag i relevanta artiklar kring ämnet, men även utdrag ur vetenskapliga böcker och studier. För att välja ut relevanta källor använde vi oss först och främst av sökord som ligger nära det ämne vi valt att studera (Jacobsen, 2002). Dem sökorden som används i olika kombinationer med varandra är:

- *Software documentation*
- *System documentation*
- *Knowledge management*
- *Knowledge sharing*
- *Documentation debt*

Dessa sökord har även används tillsammans med andra stödord som *problems, issues, quality, availability, consequences* etcetera. Då vi endast var intresserade av den allmänna bilden av intern systemdokumentation har data som presenterats i samband med arbetssätt (agile eller scrum), dokumenthanteringsverktyg eller andra irrelevanta fenomen valts bort. Detta på grund av att denna information vinklats mot det angivna fenomenet vilket gjort den mindre intressant för vår studie.

Vi var noga med att använda oss av litteratur som tog upp vårt ämne, men som även ansågs högst trovärdigt. När samma teori lyfts i flera olika litteraturavsnitt som är oberoende av varandra, stärktes vårt förtroende för att innehållet är sannhetsenligt (Jacobsen, 2002). På grund av detta valdes en referens ut om innehållet stämde överens med annan litteratur. Om en källa ansågs trovärdig och tog upp information relaterat till vårt ämne användes dess referenslista för att hitta mer information. Nackdelen är att dessa källor kan vara beroende av varandra och därav visa en förskönad bild av fenomenet (Jacobsen, 2002). För att motverka denna nackdel kontrollerades källor relaterade till varandra även mot andra oberoende källor.

### 3.2.2 *Val av organisation*

För att förenkla valet av organisation staplade vi upp ett antal kriterier som företagen var tvungna att uppfylla (Jacobsen, 2002). Först och främst var vi intresserade av företag som har ett starkt fokus på systemutveckling. Därmed var vi inte intresserade av företag som har en enskild IT-avdelning utan snarare företag där IT genomsyrar hela organisationen. Mognad hos företag var också ett kriterium som vi ansåg viktigt. Vi bedömde därför att äldre företag, som funnits en längre tid på marknaden, skulle ha ett bredare perspektiv och således kunna ge oss en mer nyanserad bild av användandet av intern systemdokumentation. Några aspekter som diskuterades men som ansågs mindre betydelsefulla är företagets storlek, om det är rent konsultföretag och mångfalden inom företaget. Utefter dessa kriterier fick vi en bred bank av företag att välja mellan.

Eftersom vi såg starkt på företag med en bredare erfarenhet, riktade vi in oss på att ta kontakt med äldre företag med många års erfarenhet. Vi började med att höra med kontakter, och kom i kontakt med respondenter från tre av företagen den vägen. Det fanns ett stort intresse att tala med företag som vi sedan innan inte visste mycket om, men under rådande virustider var det inte helt enkelt att komma i kontakt med sådana företag eftersom de behöver prioritera annat arbete. Vi kontaktade 16 företag, men endast sex hörde av sig. De tre företag där vi hade kontakter på ställde upp direkt, ett annat meddelade att det inte fanns tid och två skulle höra sig omkring på företaget. Endast ett av dessa tre företag återkopplade.

Två av de valda företagen, F1 och F3 (se Tabell 3.1) är båda stora och äldre företag med kunder över hela världen som sitter på många års erfarenhet inom systemutveckling. F1 kom vi i kontakt med genom att skicka ett mejl till en mellanchef på företaget. Hen skickade sedan ut förfrågan till sina medarbetare och tre utvecklare av olika slag hörde av sig och ville vara med på intervju. För att komma i kontakt med F3 pratade vi med personen vi kände på företaget, som i sin tur kontaktade en person som hen ansåg sitta på kompetens och erfarenheter inom intern systemdokumentation. Även F2 är ett äldre företag, men ett betydligt mindre är både F1 och F3. Det gick till på ungefär samma sätt när vi kontaktade det här företaget, där vår kontaktperson skickade ut en förfrågan till sina medarbetare. F4 fick vi tag igenom personlig kontakt, men även här var respondenten inte någon vi kände sedan innan.

### 3.2.3 *Val av respondenter*

Valet av respondenter är viktigt eftersom det direkt påverkar perspektivet på studien. Syftet med den kvalitativa metoden i vår uppsats är att få en djupare förståelse för ämnet ut respondenternas egna perspektiv och åsikter (Jacobsen, 2002). Vi valde respondenter som vi trodde kunde bringa den typ av information som vi behövde för att uppnå vårt forskningssyfte (Jacobsen, 2002).

Eftersom vi önskar ett brett men samtidigt specifikt perspektiv ifrån den valda arbetsrollen, var respondenternas yrkesroll viktig. De skulle alla jobba eller ha jobbat som systemutvecklare. Dock var utvecklings- och programmeringspråkmiljöer inget vi tog i beaktning. Som systemutvecklare ansåg vi även arbete med intern systemdokumentation som ett måste. Däremot behövde det inte vara ett aktivt arbete, utan respondenterna kunde ha gjort det under tidigare arbetsförhållanden. Vi ansåg detta var ett måste eftersom en respondent utan faktiskt insyn på ämnet, leder till att det blir otroligt svårt att lita på den information som hen förmedlar.

För att få ett så säkert perspektiv som möjligt, ansåg vi att respondenterna helst, men inte nödvändigtvis, skulle ha flera års erfarenhet av kontakt eller arbete med intern systemdokumentation. Det grundas i att det är lättare att lita på den information som kommer från en person som är mycket kunnig inom området, snarare än från en nykomling (Jacobsen, 2002). Så blev även fallet, och en större del av respondenterna började till och med närma sig pensionsåldern. Vi såg framförallt deras ålder och år i arbetslivet som en stor tillgång, men vi är även medvetna om att personer som arbetat med samma eller liknande saker under många år kan ha ett inarbetat synsätt och kan ha svårt att se saker ur nya eller andra perspektiv. Vi upplevde dock inte det här och fick istället många nyanserade svar.

De olika respondenterna fick komma ifrån samma företag, men i så fall från olika avdelningar eftersom det önskas ett bredare synsätt på undersökningsområdet. Vi tror nämligen att en avdelning kan ha ett likasinnat sätt att se på arbetet med intern systemdokumentation och anser därför det vara mer lämpligt att respondenterna inte har en för nära arbetsrelation. Tre utav fem intervjuer skedde med respondenter på samma företag. Vi har förståelsen för att det kan ge ett företagssynsätt. Däremot var dessa från olika arbetslag och hade alla tidigare erfarenheter från tidigare arbetsroller och företag. Det här tyckte vi gjorde att åsikterna de lyfte fram ändå kunde användas för studiens ändamål.

För att komma i kontakt med respondenterna skickades förfrågan ut via e-mail till enskilda personer. I mejlet förklarades ämnet och syftet och även förfrågan om att mottagaren gärna fick skicka vidare mejlet till någon som ansågs lämpad. På så sätt applicerades snöbollsmetoden (Jacobsen, 2002). Under rådande virustider, var det knapert med svar från företagen eller så fick vi flera svar om att det inte fanns tid. Därför fick alla som svarat vara med på intervju.

### 3.3 Intervjuer

För att samla in data och information till vår kvalitativa studie valde vi att använda oss av metodvalet intervju. Vi bestämde oss för att förhålla oss till semistrukturerade intervjuer. För att kunna förhålla oss till detta intervjusätt, förberedde vi ett antal fasta frågor som bearbetats fram, däremot behövdes dessa inte ställas i någon given ordning (Oates, 2005). Semistrukturerade intervjuer handlar snarare om att få igång en konversation och således kunde vi ställa följdfrågor eller ändra intervjufrågorna beroende på respondenternas svar (Oates, 2005). Respondenterna fick sedan diskutera sina personliga åsikter och erfarenheter fritt, istället för att lägga upp intervjun som en faktacheck (Oates, 2005). Vi valde att utföra intervjuerna på så vis eftersom vi ansåg att den data vi sökte grundar sig i personliga erfarenheter och inte ett bestämt tankesätt.

Fremst var det av intresse att hålla intervjuer i person eftersom vi ansåg det vara lättare att få en personlig kontakt men även för att respondenten talar mer öppet vid ett fysiskt möte (Jacobsen, 2002). Tyvärr finns en risk för att respondenten talar osanning eller förskönar ett ämne via telefon (Jacobsen, 2002), men i och med rådande virustider förstod vi att det skulle finnas en önskan från respondenterna att inte genomföra intervjuerna i person. Vid första mailkontakt erbjöd vi därav möjligheten att hålla intervjun genom ett internetbaserat kommunikationsverktyg som Zoom, Microsoft Teams eller Skype. Logistik blev således inget problem och vi hade möjlighet att kontakta företag som låg utanför vårt geografiska omfång. Detta bidrog även till att vi sparade in dyrbar tid som annars hade lagts på att ta sig till och från intervjuplatser.

Vi diskuterade ett längre tag om det ansågs lämpligast att hålla våra intervjuer enskilt eller i grupp. Då systemutvecklare ofta arbetar i team kunde det tänkas lämpligt att hålla intervjuerna i samlad skara, framförallt eftersom individuella intervjuer kräver mycket tid om en större undersökning är det som åsyftas (Jacobsen, 2002). Däremot kan gruppintervjuer lätt blir kaotiska och osammanhängande när ett öppet och brett område ska diskuteras (Jacobsen, 2002). Eftersom intern systemdokumentation är ett område där det finns många skildrande åsikter, ansåg vi att individuella intervjuer skulle göra analyseringen av data enklare. Vi trodde även att individuella intervjuer skulle uppmana till en större bekvämlighet där respondenterna i ro skulle kunna framför sina egna åsikter.

Följande är en översikt av respondenter och intervjuer med tillhörande bilaga.

Respondent	Företag	Yrkesroll/titel	Metod	Intervjulängd	Bilaga
R1	F1	Systemarkitekt	Skype	42 min	B
R2	F1	Systemutvecklare	Skype	45 min	C
R3	F2	QA Manager	Skype	34 min	D
R4	F1	Senior systemutvecklare	Skype	26 min	E
R5	F3	Product Owner	Microsoft Team	28 min	F
R6	F4	Software Development Consultant	Zoom	44 min	G

Tabell 3.1: Översikt val av respondenter och intervjuer

### 3.4 Intervjuguide

Den områdesmodell som skapades i kapitel 2.6, låg till grund för intervjuguiden. Med hjälp av modellen, som skapats utefter granskad litteratur, framtoogs relevanta intervjufrågor. Syftet var att modellen skulle hålla oss på rätt spår igenom hela intervjun och säkerställa att vi berörde alla ämnen som behövdes för att kunna besvara vår forskningsfråga (Jacobsen, 2002). Vi valde en semi-strukturerad intervju, vilket betyder att vi utformade ett antal fasta frågor tillhörande varje ämne (Oates, 2005) och ett antal följdfrågor, men dessa ställdes endast beroende på svar från respondenten. Detta eftersom respondenten ibland tog in svaret på underfrågan i huvudfrågan, eller att svaret blev sådan att underfrågans innehåll inte längre var viktigt. ”Varför och hur”-frågor kunde även läggas till om vi inte tyckte respondenten besvarade detta genom den huvudsakliga frågan. Intervjufrågorna, samt eventuella underfrågor finns i sin helhet under kapitlet Bilagor (se Bilaga A).

Innan vi påbörjade intervjun började vi med att, i enlighet med Oates (2005), introducera oss, förklara vår uppsats, dess syfte och vårt slutgiltiga mål. Vi fortsatte med att förklara att intervjun var helt frivillig och frågade sedan om hen godkände att vi spelade in intervjun. Vi såg gärna att vi kunde spelade in intervjun eftersom det skulle underlätta transkriberingsarbetet, samt att



vi skulle kunna fokusera på intervjuprocessen istället för att behöva anteckna (Oates, 2005). Samtliga respondenter godkände inspelningen varav en önskade att inspelningen raderades efter transkriberingen. Efter transkriberingsarbetet var utfört, raderades samtliga ljudfiler. Avslutningsvis, innan ljudupptagningen sattes igång, förmedlade vi att namn på person och att företaget hen arbetar på eller har arbetet på kommer att vara helt anonymt.

Vi påbörjade intervjun med några introduktionsfrågor så som respondentens företag, arbetsroll och arbetsuppgifter (se Tabell 3.2). Syftet med detta var att få igång en konversation, då vi hade valt en semi-strukturerad intervju och ville att det skulle bli ett flöde i samtalet, snarare än en faktacheck (Oates, 2005). Vi ville också ha en grundtanke om vem intervjupersonen var och hur deras arbetsroll potentiellt skulle komma att påverka ämnet.

<b>Introduktionsfrågor</b>
<ul style="list-style-type: none"> <li>• Vill du förklara vad företaget du arbetar på jobbar med?</li> <li>• Vad är din arbetsroll/titel?</li> <li>• Vilka arbetsuppgifter ingår i din arbetsroll?</li> </ul>

**Tabell 3.2: Introduktionsfrågor**

### 3.4.1 Intern systemdokumentation

Med hjälp av dessa intervjufrågor tog vi upp intern systemdokumentation i dess enklaste form. Vi var intresserade av vad respondenten hade för uppfattning kring den interna systemdokumentation som finns på deras företag, om denna dokumentation används av respondenten, samt till vilken grad respondenten förlitar sig på den information som presenteras i intern systemdokumentationen (se Tabell 3.3). Syftet var att få en överskådlig bild av respondentens relation till ämnet, då intern systemdokumentation ofta delar åsikter hos utvecklare. En negativ, positiv eller neutral uppfattning av intern systemdokumentation kunde bli en indikation på hur resten av intervjun skulle komma att te sig, vilket även gjorde det möjligt för oss att ändra, ta bort eller omformulera intervjufrågorna (Oates, 2005).

<b>Kategori</b>	<b>Undersökningsområde</b>	<b>Intervjufrågor</b>
<b>Intern system-dokumentation</b>	Användande av systemdokumentation	<ul style="list-style-type: none"> <li>• Använder du dig aktivt av intern systemdokumentation i ditt vardagliga arbete?               <ul style="list-style-type: none"> <li>a. Anser du att intern systemdokumentation förenklar ditt arbete? Varför/hur?</li> </ul> </li> </ul>
	Syfte	<ul style="list-style-type: none"> <li>• Vad ska intern systemdokumentation uppnå för kriterier för att vara användbar för dig?</li> </ul>
	Problematik	<ul style="list-style-type: none"> <li>• Hur väl förlitar du dig på den systemdokumentation som redan finns?</li> </ul>

	Kunskapsdelning	<ul style="list-style-type: none"> <li>• Hur fungerar kunskapsdelning inom ert företag?</li> <li>• Hur gör ni när någon kunskapsbesittande person närmar sig att lämna företaget?</li> <li>• Arbetar du med underhållsarbete av ett system?             <ul style="list-style-type: none"> <li>a. Hur brukar du hitta den kunskap som krävs för att utföra arbetet?</li> <li>b. Hur användbar är dokumentationen kring systemet?</li> </ul> </li> </ul>
--	-----------------	---

Tabell 3.3: Inledande intervjufrågor tillhörande intern systemdokumentation.

### 3.4.2 Uppdatering

Efter att vi hade frågat ganska generella frågor kring intern systemdokumentation och kunskapsdelning gick vi vidare till frågor som berörde de tre faktorerna som diskuterats i vår tidigare forskning (se kapitel 2). För att vara sammanhängande valde vi att ta upp faktorerna i samma ordning som de skrivits i tidigare kapitel. Således började vi med uppdatering. Tanken var att förstå hur varje respondent ställer sig till uppdatering av intern systemdokumentation, hur viktigt uppdatering är för den intern systemdokumentationen och andra mer personliga åsikter kring eget arbete med uppdatering. På ett antal frågor ställdes följdfrågor beroende på hur svaren tedde sig, dessa följdfrågor går att se i Bilaga A.

Kategori	Undersökningsområde	Intervjufrågor
<b>Uppdatering</b>	Anledning till bristfälligt uppdateringsarbete	<ul style="list-style-type: none"> <li>• Hur viktigt är det för dig att uppdatera systemdokumentation?</li> <li>• Varför känner du behovet av att uppdatera/inte uppdatera din dokumentation?</li> <li>• Av vilken anledning uppdaterar du den interna systemdokumentationen?</li> <li>• Hur ofta uppdaterar du i genomsnitt din interna systemdokumentation?</li> </ul>
	Uppdatering beroende på kodändringens storlek	<ul style="list-style-type: none"> <li>• Dokumenterar du annorlunda beroende på storlek av ändringar i koden?</li> </ul>
	Användande av föräldrar dokumentation	<ul style="list-style-type: none"> <li>• Använder du systemdokumentation även om den är gammal och inte är uppdaterad till aktuell status?</li> </ul>

	Utdaterade systemdokuments påverkan på kunskapsdelning	• Hur påverkar icke-uppdaterad dokumentation delandet av kunskap?
	Konsekvenser av dåligt uppdaterade interna systemdokument	• Hur tycker du att icke-uppdaterad intern systemdokumentation påverkar systemutvecklingen/underhållsarbete/systemet?

Tabell 3.4: Intervjufrågor tillhörande faktorn uppdatering

### 3.4.3 Dokumentationskvalitet

Nästa faktor berörde dokumentationskvaliteten. Eftersom det har visats sig att dokumentationskvaliteten på den interna systemdokumentationen ofta är bristfällig, var det av stort intresse att fråga respondenterna om deras syn på kvaliteten. Även om vi har funnit en del kvalitetsattribut som från undersökningar anses viktiga, tyckte vi att det var vitalt att fråga varje individ om dess syn på vad som motsvarar hög dokumentationskvalitet. Det kan självklart se olika ut för olika individer, och vi menade att det därför var bra att, genom första och andra frågan, bestämma vad respondenten utgår ifrån innan vi fortsatte med frågor baserade på kapitlet tidigare forskning.

Kategori	Undersökningsområde	Intervjufrågor
<b>Dokumentationskvalitet</b>	Vikten av olika kvalitetsattribut	<ul style="list-style-type: none"> <li>• Vad anser du motsvarar hög dokumentationskvalitet?</li> <li>• Vad är dina förväntningar på kvaliteten?</li> <li>• Tycker du dina förväntningar uppfylls i den systemdokumentation du använder?</li> <li>• Tar du hänsyn till dessa kvalitetsattribut när du själv skriver systemdokumentation?</li> </ul>
	Korrekthet	• Hur viktiga är följande kvalitetsattribut för kvaliteten hos den systemdokumentation du använder dig av?
	Läsbarhet	• Hur viktiga är följande kvalitetsattribut för kvaliteten hos den systemdokumentation du använder dig av?
	Struktur	• Hur viktiga är följande kvalitetsattribut för kvaliteten hos den systemdokumentation du använder dig av?
	Fullständighet	• Hur viktiga är följande kvalitetsattribut för kvaliteten hos den systemdokumentation du använder dig av?

		<ul style="list-style-type: none"> <li>• Hur ofta innehåller systemdokumentation all nödvändig information som krävs för att förstå, underhålla och använda systemet?</li> </ul>
	Konsekvenser av dålig dokumentationskvalitet	<ul style="list-style-type: none"> <li>• Hur tycker du att intern systemdokumentation med dålig kvalitet påverkar systemutvecklingen/underhållsarbete/systemet?</li> </ul>
	Dokumentationskvalitetens påverkan på kunskapsdelning	<ul style="list-style-type: none"> <li>• Hur bidrar kvalitetsnivån till delningen av kunskap?</li> </ul>

Tabell 3.5: Intervjufrågor tillhörande faktorn dokumentationskvalitet

#### 3.4.4 Tillgänglighet

För att undersöka tillgängligheten av intern systemdokumentation hos företag valdes följande frågor baserade på tidigare forskning. Precis som med tidigare faktorerna, var vi intresserade av att veta respondenternas personliga åsikter kring tillgängligheten av intern systemdokumentation. Vi tyckte att det kunde finnas en intressant skildring på hur dokumentation faktiskt är förvarad gentemot hur den önskvärt borde vara förvarad, och dessa frågor var därför av intresse. Även versionshanteringen ansåg vi vara viktig, eftersom vi förstätt från tidigare forskning att det kan bidra till förvirring.

Kategori	Undersökningsområde	Intervjufrågor
Tillgänglighet	Åtkomstrestriktioner	<ul style="list-style-type: none"> <li>• Är det lätt att få tillgång till intern systemdokumentation när du behöver den?</li> <li>• Var och hur är systemdokumentationen förvarad?</li> <li>• Använder ni er av ett dokumenthanteringssystem?</li> <li>• Hur önskar du att den var förvarad, för att bringa mest nytta?</li> </ul>
	Versionshantering	<ul style="list-style-type: none"> <li>• Sparas endast den senaste versionen av dokumentationen eller sparas även äldre versioner?</li> </ul>

		• Är det lätt att veta vilken som är relevant/senast uppdaterad?
	Hur tillgänglighet påverkar kunskapsdelning	• Hur bidrar tillgängligheten av systemdokumentation till delningen av kunskap?
	Konsekvenser av otillgänglig intern systemdokumentation	• Hur tycker du att begränsad tillgänglighet inom intern systemdokumentation påverkar systemutveckling/underhållsarbete/systemet?

Tabell 3.5: Intervjufrågor tillhörande faktorn tillgänglighet

### 3.4.5 Avslutande frågor

Avslutningsvis ställde vi frågor angående tillägg av information eller vidare kontakt (se Tabell 3.7). Detta för att runda av intervjun samt fånga upp ifall att vi har missat att fråga någonting som respondenten ansåg viktigt för vår studie (Oates, 2005). Sist tackade vi för medverkan och frågade vi om det var okej att vi skickade den färdigställda transkriberingen till respondenten för granskning.

<b>Avslutande frågor</b>
<ul style="list-style-type: none"> <li>• Finns det någonting du hade velat lägga till som du kommit på under intervjuens gång?</li> <li>• Är det okej om vi kontaktar dig om vi skulle ha några andra frågor kring ämnet eller förtydligande av det som redan sagts?</li> <li>• Är det okej om vi skickar transkriberingen till dig, så du kan gå igenom den och se så allt ser bra ut?</li> </ul>

Tabell 3.7: Avslutande frågor.

## 3.5 Bearbetning av data

Intern systemdokumentation och dess problemområden är ett ämne som väckte många olika åsikter hos respondenterna. Trots att vi hade specifika frågor, som egentligen inte krävde alltför långa svar, märktes entusiasm hos respondenterna och vi kunde få väldigt långa och utsvävande utläggningar. Vi såg därför att analysarbetet skulle komma att bli krävande och för att förenkla detta arbete behövde vi få en helhetssyn över vår data. Genom att översätta våra intervjuer från ljud till text, samt kategorisera texten beroende på diskuterat ämne, fick vi en smalare, mer översiktlig bild av vår data, någonting som skulle komma att förenkla vårt empiriarbete.

### 3.5.1 Transkribering

För att förenkla analys av vår insamlade data valde vi att transkribera intervjuerna (Oates, 2005). Transkribering av intervjun gjordes direkt efter intervjun av en av uppsatsskrivarna, eftersom samtalet fortfarande var färskt i minnet. Vid transkribering förhöll vi oss till exakt vad

som sagts, men valde att ta bort harklande ljud eller upprepningar då dessa kom att försämra transkriberingens läsbarhet. Transkriberingen kontrollerades sedan av den uppsatsskrivaren som inte hade genomfört skrivarbetet. Eventuella stavfel, meningsuppbyggnader eller förvirrande fraser diskuterades mellan parterna och rättades. Efter detta skickades den transkriberade intervjun tillbaka till tillhörande respondent, detta för att säkerställa att all fakta ansågs korrekt, samt att innehållet tedde sig så som respondenten hade menat (Oates, 2005). Om respondenten ville ändra eller ta bort någonting i transkriberingen reviderades detta och fördes sedan in i uppsatsen.

### 3.5.2 Kategorisering av data

För att ytterligare underlätta vårt framtida empiriarbete gjordes en kategorisering på all insamlad intervjudata (Jacobsen, 2002). Den aspekt som framkom, som fick oss att överväga kategorisering, var att olika ämnen lätt överlappade. Trots att vi hade ett naturligt flöde av intervjufrågor där ett ämne diskuterades åt gången, bredde respondenterna gärna ut sina svar och kunde lätt vandra över på ett annat ämne som redan tagits upp eller som låg framför oss. För vår egen skull ansåg vi det därför enklast att dela in varje stycke i sifferföljd, samt att tilldela varje stycke en eller flera kategoriseringskoder. Siffrorna och koderna gjorde det tydligare och enklare att hitta eftersökt innehåll. Detta underlättade vårt arbete då vi inte behövde gå igenom varje intervju gång på gång för att hitta det ämne vi ville diskutera i empirin.

För att säkerställa att ämneskategoriseringen är korrekt, det vill säga att rätt kod har angivits i rätt stycke, har uppsatsskrivarna gått igenom varje intervju och tillsammans diskuterat varje stycke, samt tilldelat rätt ämneskategori.

Koderna är som följande:

Kod	Kategori
ISD	Intern systemdokumentation
KD	Kunskapsdelning
U	Uppdatering
K	Kvalitet
T	Tillgänglighet
KO	Konsekvenser

Tabell 3.8: Ämneskategori

## 3.6 Etik

Eftersom den kvalitativa undersökningsmetoden ofta innebär personliga intervjuer som baseras på mänskliga åsikter, erfarenheter och synpunkter blir etik ett relevant ämne. För att alla intervjuer skulle te sig lika valde vi att skriva ett etikavsnitt. Detta avsnitt är avsett att hjälpa oss replikera och efterhålla vissa aspekter, så att alla intervjuer utgår ifrån samma grund.

### 3.6.1 Samtycke

Alla studier som kräver personliga intervjuer ska baseras på respekt för respondenten (Orb, Eisenhauer & Wynaden, 2001). Innan påbörjad intervju förklarade vi för respondenten att hen medverkade av egen, fri vilja och att hen under hela intervjuprocessen, samt efteråt, hade rätten

att dra sig ur undersökningen utan att behöva ge en förklaring till varför (Orb, Eisenhauer & Wynaden, 2001). Där efter var det godtagbart för respondenten att ifrågasätta hur dess svar behandlades samt kunde önska utdrag av den text som berörde hans svar (Orb, Eisenhauer & Wynaden, 2001). Dessa aspekter kändes självklara för oss och klargjordes för alla respondenter innan intervjustart. Det berodde på att vi ville ha fortsatt bra relation med respondenterna och önskade att de ska vara stolta över den insats de hade gjort. Vi vill inte att de skulle känna obehag över att något de önskar osagt skulle komma att användas som underlag för studien.

### 3.6.2 Anonymitet

Anonymitet handlar om att den medverkande respondenten ej skall gå att spåra med hjälp av dem detaljer som presenteras om hen i undersökningen (Saunders, Kitzyng & Kitzyng, 2015). Till en början hade vi planerat att endast respondenternas namn skulle vara anonyma, men efter vidare funderingar, samt efter diskussion med en intervjuperson kom vi fram till att även företagsnamnen fick vara anonymt. Det grundade vi i att ämnet kan anses känsligt och att en person har större benägenhet att delta i en intervju om möjligheten till anonymitet presenteras (Saunders, Kitzyng & Kitzyng, 2015). För att styrka på möjligheten för anonymitet inledde vi varje intervju med att förklara att respondentens namn och företag inte kommer tas med i uppsatsen. Däremot ansågs ansvarsområde och jobbtitel vara essentiellt för studien, således var det inte möjligt att säkerställa anonymitet på den fronten, vilket ingen av respondenterna motsatte sig.

### 3.6.3 Transkribering och inspelning

Den sista punkten vi tog upp innan intervjustart var inspelning och intervjun avrundades med att nämna transkribering. Intervjuerna spelades gärna in just för att kunna underlätta det framtida transkriberingsarbetet. Eftersom alla intervjuer skedde via ett internetbaserat kommunikationsverktyg, kände vi obligationen att ställa frågan om det var okej för respondenten att intervjun spelades in som ljudupptagningen. Om fallet inte var sådant, var tanken att en av intervjuerna tog anteckningar medan den andre stod för själva intervjufrågorna. Alla respondenter godkände dock transkribering så anteckning blev aldrig aktuellt. För att genomföra transkriberingen så effektivt som möjligt, användes verktyget Express Scribe Transcription Software för att sakta ner tempot på ljudfilerna. Det bidrog till att vi inte behövde pausa och hoppa tillbaka allt för mycket för att hinna med i tempot.

## 3.7 Validitet

Validitet är ett begrepp som redan tagits i beaktning genom hela metodikkapitlet och diskuterats i anknytning till berört område. Konsekvenser av varje område diskuteras genomgående i kapitlet. Däremot ändrades studiens fokus från en typ av skuld till en annan, samt från *vilka* till *hur*, sent och efter att intervjuerna utförts och en del av intervjufrågorna kan därför ses som felriktade. Detta har tagits hänsyn till och slutsatser från dessa frågor har inte gjorts. Vår empiriska studie är även begränsad till ett fåtal respondenter vilket bör tas i beaktning vid användande av slutsatserna som inte bör ses som faktum, utan snarare som anvisningar.

## 4 Empiri

*I avsnittet presenteras det relevanta innehållet ur dem sex intervjuerna. Innehållet från de olika respondenterna jämförs med varandra och är uppdelade inom undersöksområdena intern systemdokumentation, uppdatering, kvalitet och tillgänglighet. Avsnittet börjar med en introduktion av respondenterna.*

### 4.1 Presentation av respondenter

Respondent 1 har arbetat som systemarkitekt på ett stor IT-företag (R1, rad 154), som finns på flera platser över hela världen. Hen var specifikt inriktad på hälso- och sjukvårdssystem för kommuner och regioner (R1, rad 152). Företaget utvecklar administrativa system som exempelvis journalsystem (R1, rad 152). Hens arbetsuppgifter inkluderade att se till att få in all nödvändig information från säljarna till upphandlingen, samt kontrollera att utvecklarna förstod sina arbetsuppgifter och skrev tillhörande dokumentation (R1, rad 154). Idag är hen pensionerad, men jobbar fortfarande deltid på företaget med diverse konsultuppdrag (R1, rad 22).

Respondent 2 arbetar som systemutvecklare på ett stort IT-företag (R2, rad 8). Hen förklarade att företaget bygger en mängd olika IT-system till olika typer av branscher, exempelvis till oljeindustrin, hälso- och sjukvårdsindustrin, skogsindustrin och till myndigheter och kommuner (R2, rad 5). Tidigare har hen varit projektledare på ett medicintekniskt företag (R2, rad 24), men idag är hens arbetsuppgifter relaterade till front-end programmering (R2, rad 140).

Respondent 3 arbetar i dagsläget som Quality Assurance Manager på ett svenskt konsultföretag, med inriktning på IT, främst integration och analytics (R3, rad 4). Företaget har även ett par andra specialavdelningar, men de två nämnda anses vara företagets huvudområden (R3, rad 4). Hens arbetsuppgifter är främst kvalitetssäkring och testning (R3, rad 6). Tidigare har hen arbetat som systemutvecklare (R3, rad 8).

Respondent 4 arbetar på ett stort IT-företag, som utvecklar olika system till många olika branscher, men hen jobbar främst med sjukvårdssystem (R4, rad 7). Dock inte journalsystem, snarare stödsystem som kallas PAS, det vill säga patient administrativa stödsystem (R4, rad 7). Idag har hen titeln senior systemutvecklare, men arbetsuppgifterna kan vara allt från systemarkitekt till utveckling och support (R4, rad 9).

Respondent 5 jobbar med interna system på IT-avdelningen på ett stort IT-företag som bygger övervakningskameror (R5, rad 2). Hen har titeln Product Owner och är ägare av en roadmap och är därmed systemägare (R5, rad 6). Respondent 5 har innan arbetat som systemutvecklare i tio år (R5, rad 33). Det som respondent 5 lyft under intervjun är hur arbetet fungerar på hens avdelning.

Respondent 6 arbetar på ett konsultföretag som hjälper företag att jobba smartare med mjukvaruutveckling (R6, rad 2) och för tillfället är hen positionerad på ett danskt företag (R6, rad 4). Hen har ingen fast titel men skulle säga sig vara Continuous Developer Coach eller Software Developer Consultant och arbetar med systemutveckling (R6, rad 8 & 8).



## 4.2 Intern systemdokumentation

Respondent 2 ser på intern systemdokumentation ur ett managementperspektiv (R2, rad 56) och menar att det måste finnas ett ramverk att följa snarare än ett fritt tänkande kring hur och om arbetet ska göras (R2, rad 22). Även respondent 3 anser att dokumentationsarbetet beror på organisationen (R3, rad 70, 112). Ramverket, ett kvalitetssystem, ska bestämma vilka typer av dokument som ska tas fram och hur dessa ska hanteras genom dess livscykel (R2, rad 22). Med ett sådant ramverk, menar respondent 2 att osämja kring arbetet med systemdokumentation inte behöver finnas (rad 32). Osämja uppstår snarare när ett bestämt ramverk inte finns. Då blir det lätt att skilda åsikter framkommer som gör ett gemensamt arbetssätt svåråtkomligt (R2, rad 32).

Respondent 2 fortsätter med att berätta att eftersom de inte har ett kvalitetssystem, är det svårt att svara på om hen utför ett aktivt dokumentationsarbete eller inte (R2, rad 41). Hen menar att det inte går att utföra ett sådant arbete ordentligt om ett kvalitetssystem inte används (R2, rad 41). Däremot använder hen sig av utvecklardokumentation som baseras på vad som behöver göras för att kunna kommunicera effektivt inom teamet (R2, rad 50). Utvecklardokumentationen förenklar kommunikationen, då parter slipper gå och fråga andra medarbetare utan kan istället slå upp eftertraktad information i dokumentet (R2, rad 54).

På företaget där respondent 4 arbetar är arbetet med intern systemdokumentation slarvigt (R4, rad 14). Hen försöker själv dokumentera på en övergripande nivå, det vill säga vad det viktigaste är som systemet gör (R4, rad 33), med arkitekturbild, systemskiss, vilket ökar förståelsen (R4, rad 35 & 37). Intern systemdokumentation förenklar arbetet både för hen och andra. Framförallt när nya resurser kommer in, då intern systemdokumentation ger en stadigare bas (R4, rad 19).

Respondent 1 anser att någon typ av intern systemdokumentation alltid måste finnas (R1, rad 10). Det är en förutsättning för att överhuvudtaget kunna utföra systemutvecklingsarbete, men även ett arbete som systemutvecklare sällan skriver på eget bevåg (R1, rad 12). Trots det, anser respondent 1 att den här typen av dokumentation oftast är användbar (R1, rad 32).

Respondenterna 4 och 5 vill att intern systemdokumentation ska ge en övergripande förståelse för systemet och dess funktioner (R4, rad 21; R5, rad 19), men även vara lättillgänglig, och lättläst samt enkel att uppdatera (R5, rad 14). Respondent 5 menar att det är svårt att få en övergripande bild av systemets funktioner, eftersom allt inom IT-världen rör sig väldigt snabbt (R5, rad 19). Även respondent 3 ser att dokumentationsarbetet är svårt att utföra ordentligt inom ett område där allting går väldigt fort (R3, rad 15).

### 4.2.1 Kunskapshantering inom organisationer

Inom samtliga företag och team, fungerar den största kunskapsdelningen med hjälp av konversationer i person eller överlämningsprojekt (R1, rad 20; R2, rad 66; R3, rad 24, R4, rad 30; R5, rad 23, R6, rad 30) istället för via dokumentation.

Både respondent 1 och 4 berättar att kunskapsdelningen inom företaget fungerar ganska dåligt (R1, rad 18; R4, rad 26). Anledningarna till att muntliga konversationer är den främsta sättet att kunskapsdela, är olika. Respondent 1 föredrar att förmedla kunskap genom både konversationer, men också dokumentation och berättar att på företaget hen jobbar på sitter medarbetarna över hela världen (R1, rad 20). För att kunna återvinna kunskapen krävs ett större

dokumentationsansvar och även då på engelska (R1, rad 20). Det fungerar liknande på det företaget där respondent 5 arbetar; överlämning och informationsspridning har gått från person till person istället för via dokumentation (R5, rad 23). Dokumentation är istället ett sätt att sprida kunskap till de personer som ofta inte går att träffa (R5, rad 23).

Eftersom företaget som respondent 2 arbetar på inte har något kvalitetssystem, sker kunskapsöverföringen till största del muntlig där medarbetarna ständigt hjälps åt att lösa problem (R2, rad 66). Teamet har tillsammans byggt upp ett sätt att hantera kunskap om viktiga komponenter (R2, rad 66). När respondent 4 började på sin arbetsplats fanns det ingen intern systemdokumentation överhuvudtaget (R4, rad 26). På grund av det här fungerar kunskapsdelningen till stor del av samtal med andra kollegor (R4, rad 30). Hos respondent 6 arbetar de i roterande par för att sprida kunskap till medarbetarna, då sitter fler än en på kunskapen och informationen (R6, rad 33). Utan detta blir det komplicerade överlämningar när någon ska sluta (R6, rad 35).

På avdelningen där respondent 1 arbetat är medelåldern hög. Utan någon väletablerad strategi för att hantera kunskap, går kunskapen lätt förlorad när de anställda går i pension (R1, rad 22). Det händer ofta att anställda hör av sig till de som gått i pension för att diskutera tidigare arbete som pensionären utfört (R1, rad 22). Företaget där respondent 2 arbetar använder sig av överlämningsprojekt. Det är viktigt eftersom företaget har många experter inom smala områden, som ingen annan har kunskap om (R2, rad 69). För att bevara kunskap inom respondent 3:s företag, används *knowledge based*-dokument där medarbetare beskriver hur de löst problem. Detta för att andra personer i framtiden ska vet hur dessa problem ska hanteras (R3, rad 20), men de har, precis som hos respondent 2, även en överlämningsperiod när någon närmar sig att lämna företaget. Överlämningsperioden är dock endast ett par dagar och det är mycket svårt att föra över någon större kunskap på så kort tid (R3, rad 24). Hos respondent 4 ber företaget personen som ska lämna att uppdatera Wiki-sidan (R4, rad 39). Respondent 5 brukar låta medarbetaren, som ska lämna, hoppa in på andra uppgifter för att se i vilka projekt som personen varit alldeles för viktig och således var problem kommer uppstå (R5, rad 27). Eftersom det ändå inte kommer att vara möjligt att hitta all information i den interna systemdokumentationen så är det bra när personerna får sätta sig med problemen själva och således får ett hum om var de ska börja. Även koden är en del av dokumentationen (R5, rad 29).

Inom konsultvärlden förklarar respondent 2 att det finns konsulter som inte har intresset av att dela med sig av sin kunskap, istället tar de ut stora kostnader för att kunna bidra med den kunskap och erfarenhet de besitter (R2, rad 114). Respondent 2 fortsätter att förklara att sådana personer ofta är de enda som kan systemet, ändra i systemet samt har kämpat hårt för att systemet inte ska bytas ut. Det medför att konsulten kan fortsätta ta högt betalt (R2, rad 114).

### 4.3 Uppdatering

Respondenterna anser att uppdatering av intern systemdokumentation är viktigt, även om det bara är respondent 4 som gör det aktivt (R1, rad 38; R2, rad 85; R3, rad 32; R4, rad 48; R5, rad 45; R6, rad 41). Att dokumentationen är uppdaterad är essentiellt för att respondenterna ska kunna lita på innehållet (R6, rad 41 & 43). Respondent 4 arbetar aktivt med uppdatering eftersom interna systemdokument måste matcha det nuvarande systemet (R4, rad 50). Om det är för stor skillnad mellan verkligheten och systemdokumentationen kan detta bidra till att dokumentationen förlorar sin långsiktiga nytta (R4, rad 50; R5, rad 60).

#### 4.3.1 *Uppdatering beroende på kodändringens storlek*

Eftersom det inte finns någon standardisering, ser uppdateringsarbetet annorlunda ut beroende på storlek av ändringen i koden (R1, rad 52). Respondent 1 ser skillnad på om kodändringarna är stora eller små och tycker att de större kodändringarna bör prioriteras i systemdokumentationen (R1, rad 54 & 56). Detta tankesätt håller respondent 4 med om och menar på att hen prioriterar större ändringar som kommer ändra systemet huvudfunktion, således väljer hen att inte dokumentera kosmetiska, mindre ändringar (R4, rad 56).

Å andra sidan menar respondent 2 att det finns ett omvänt professionellt förhållande när det gäller vilka kodändringar som prioriteras att uppdateras. Ju större ändring i koden, desto mindre benägen är utvecklare att ändra i dokumentationen (R2, rad 94). Det beror på att utvecklare har ett visst antal timmar på sig att göra ändringar i koden och kommer alltid prioritera att det arbetet ska bli färdigt före att uppdatera dokumentationen. Det finns ingen tid över för att göra större uppdateringar i dokumentationen, eftersom större kodändringar leder till större dokumentationsuppdateringar (R2, rad 98). När dokumentationsarbetet inte är obligatoriskt och timmarna är begränsade, finns det en stor risk att uppdateringarna av dokumenten inte blir gjorda (R2, rad 98). Små ändringar däremot, bidrar till små uppdateringar i dokumenten, vilket går fort (R2, rad 98). Även respondent 3 skulle säga att det är lättare att uppdatera dokumenten när det skett små ändringar i koden eftersom dokumentändringen inte blir särskilt stor (R3, rad 39). Respondent 5 tycker snarare uppdatering av ändringar handlar om svårigheten på funktionaliteten, och inte ändringens storlek (R5, rad 53). Från ett annat perspektiv, kan respondent 6:s uttalande tolkas som att ändringens storlek inte är det som spelar roll, men att större kodändringar blir mer uppmärksammade eftersom de berör fler personer (R6, rad 51).

Respondent 5 anser att dokumentationsarbete ska ske när det gäller kluriga funktioner som kräver att utvecklare annars måste lägga ett mer arbete på att endast läsa koden för att få full förståelse (R5, rad 48). Ett bevis på om en funktion måste dokumenteras är om funktionaliteten ofta ifrågasätts och inte går att förstå utan skriftlig eller muntlig förklaring (R5, rad 50).

#### 4.3.2 *Anledning till bristfälligt uppdateringsarbete*

Hur ofta och varför respondenterna uppdaterar sin dokumentation är olika. Respondent 1 uppdaterar emellanåt (R1, rad 45) och endast för att komma ihåg vad hen själv har gjort (R1, rad 42 & 47). I respondent 2:s team uppdateras den information som dem tycker att dem själva har mest nytta av. Däremot finns det ingen som tvingar medarbetarna att dokumentera vilket gör att testning och omedelbara behov prioriteras istället (R2, rad 87). Respondent 2 menar ändå att de har som minimum att uppdatera sin systemdokumentation minst varje månad (R2, rad 91). Respondent 3 tror det beror på vad det är för typ av leverans (R3, rad 37), men även beroende på situation. Respondent 4 försöker göra uppdateringsarbete så ofta som möjligt på dem dokument hen själv har skapat och känner till, då det är viktigt för att få en övergripande syn av systemet (R4, Rad 52). Respondent 5 har en gång i halvåret en avstämning med sitt team där de dokumenterar, diskuterar vad som saknas i systemdokumentationen, vilka de viktigaste delarna är samt kontrollerar gammal dokumentation (R5, rad 45). Även varje gång en ny funktionalitet läggs tills tycker hen utvecklarna måste fundera kring om dokumentationen behöver uppdateras (R5, rad 45). Respondent 6 anser att svårigheten att nå ett systemdokument är det som påverkar hens uppdateringsarbete. Om ett dokument är lättillgängligt och strukturerat på ett visst sätt, kan hen tänka sig uppdatera dokumentationen, men om det är alldeles för svårt och tidskrävande att nå dokumentation struntar hen i att uppdatera den (R6,

rad 46). Kodkommentarerna uppdaterar respondent 6 däremot direkt när ändringar sker i koden (R6, rad 46).

Något som respondent 1 däremot är duktig på är att flitigt dokumentera i koden genom kodkommentarer (R1, rad 49). Hen menar att kommentarer i koden kan räcka långt (R1, rad 49) men tycker även att utvecklare borde dokumentera mer än vad de faktiskt gör (R1, rad 140). Den interna systemdokumentationen ska inte överarbetas och det är oklart om utvecklarna läser den överhuvudtaget (R1, rad 49). Även respondent 2, 5 och 6 föredrar att läsa programkod men menar att den ofta är dåligt skriven (R2, rad 80; R5, rad 39; R6, rad 37). Designdokumentation som beskriver vad systemet gör hade däremot kunnat vara bra istället för att få det muntligt från en kollega (R2, rad 80). Utöver att läsa koden, utför respondent 6 även gärna tester istället för att direkt läsa intern systemdokumentation (R6, rad 37). Testfall och testdata är även något som respondent 4 tycker är mycket viktigt (R4, rad 152).

Att dokumentationsarbetet inte görs ordentligt tror respondent 1 och 2 bero på framförallt två faktorer: det finns vare sig tid eller tradition till hur systemdokumentationsarbetet ska utföras (R1, rad 140; R2, rad 98 & 26). Någon måste betala för att tid ska prioriteras, i slutändan blir det kunden som får ta den kostnaden. För att kunden ska betala, krävs det att säljaren av systemet förmedlar att dokumentationstid krävs utöver projektledning och utveckling (R1, rad 140). Systemutvecklare får oftast tid för att utföra ett arbete, inte för att skriva om det (R1, rad 40). Det ställs inte heller några krav, hos företag generellt, på att intern systemdokumentation är något som måste lämnas efter sig (R1, rad 30). Här håller respondent 2 med, det enda som krävs är att de håller reda på vad de själva gör (R2, rad 56). Respondent 3 vänder istället på det och beskriver att hur väl dokumentationsarbetet ska utföras beror på mottagarnivån, vem den är skriven för (R3, rad 57).

### 4.3.3 *Användandet av föråldrad dokumentation*

Respondent 1 brukar titta på systemdokumentationen även om den är gammal för att urskilja vad intentionerna är samt vilken version och revision den har till skillnad från koden (R1, rad 60). Är det väldigt stor skillnad så läser hen endast ingressen (R1, rad 60). Här har respondent 4 ungefär samma svar. Hen använder sig av gammal dokumentation eftersom hen inte har något annat val (R4, rad 62). Det är bättre att få en översiktlig förståelse för systemet, vad det gör samt vad det vill uppnå än att inte få någon förståelse alls, och således inte kunna gå in och ändra i koden (R4, rad 64). Även respondent 3 skulle använda sig av dokumentation som inte är uppdaterad för att kunna förstå systemet i det stora hela (R3, rad 42). Respondent 2 menar snarare att när hen stöter på gammal dokumentation, är det möjligt att hen skulle kolla upp vad systemet gör för att sedan uppdatera dokumentationen innan hen går vidare och gör en ändring (R2, rad 110).

Föråldrad dokumentation kan vara bra på så sätt att det finns information om vem som har byggt systemet. Det bidrar till att det blir enklare att ta kontakt om hjälp behövs (R1, rad 63) och intern systemdokumentation ersätts med personliga nätverk (R6, rad 54). Även respondent 3 anser att personen som ska ta sig an systemet kan lära sig genom att ta kontakt med personen som utvecklat systemet, men att det självklart kommer finnas en längre inkörningsperiod än om dokumentationen var uppdaterade (R3, rad 48). Respondent 6 uttrycker att icke-uppdaterade dokumentation skapar silos eftersom endast de personer som känner utvecklaren tillhörande dokumentationen kommer våga ta kontakt med personen (R6, rad 59). Nyanställda utvecklare med mindre nätverk kommer därför att gå miste om kunskap (R6, rad 59).

Respondent 1 och 2 anser att när en ny person ska sättas sig in i att arbeta med ett nytt system, kommer föråldrad dokumentation leda till att det tar tid samt kostar pengar (R1, rad 81; R2, rad 76). På grund av detta, är det viktigt att utvecklare som ska byta projekt eller lämna över systemutvecklingen till någon ny, uppdaterar den interna systemdokumentationen innan de lämnar (R2, rad 85; R4, rad 71; R3, rad 32), så att det blir lättare att ta över (R3, rad 46). Speciellt vid fall där personen lämnar företaget helt, är det viktigt eftersom personen inte längre går att ta hjälp av (R3, rad 32). Följs däremot ett kvalitetssystem så får systemet inte överlämnas om inte dokumentation har gjorts, alla steg i kvalitetssäkringen måste göras för att kunna godkännas (R2, rad 85).

#### 4.3.4 Konsekvenser

Den interna systemdokumentationen måste vara uppdaterad för att det överhuvudtaget ska kunna förändra och utveckla systemet. Utan dokumentation blir det väldigt kostsamt och tidskrävande att lära sig systemet innan det är möjligt att börja arbeta med det (R1, rad 81; R4, rad 71). Det påverkar systemutvecklingen på så vis att utvecklare till slut inte vågar gå in och ändra i systemet eftersom dem inte förstår hur det fungerar (R4, rad 73), vilket kan leda till att dem tappar kontrollen över systemet (R4, rad 75). Detta beror dock på om det är en äldre utvecklare eller någon ny som ska ta sig an systemet. En äldre utvecklare, eller en utvecklare som tidigare arbetat med systemet, kommer inte ens behöva öppna systemdokumentationen, men en ny utvecklare kommer inte kunna arbeta med eller utföra några ändringar i systemet om det inte finns väsentlig dokumentation (R2, rad 118; R4, rad 71; R6, rad 66). Detta kommer således bidra till att arbetet går långsammare och systemet blir sämre (R6, rad 62; R3, rad 53).

Uppdaterad dokumentation är självklart mer värdefull och ju färskare desto bättre (R1, rad 65). Respondent 1 och 4 menar att skillnaden är så pass stort att uppdateringsarbetet absolut är värt bördan (R1, rad 68; R4, rad 68). Om ändringar inte görs förrän någon gång varje år finns det risk att utvecklare börjar blanda ihop olika system de arbetar med (R4, rad 68). Dock är frekvent uppdateringsarbete någonting som kunden måste vara villig att betala för (R1, rad 68).

Icke-uppdaterad dokumentation kan bringa ett team att dra fel slutsatser, vilket i sin tur kan leda till att fel beslut tas och att fel ändringar görs (R5, rad 70). Således kan andra system, som är beroende av systemet som ändras, också haverera, vilket kommer resultera i stora kostnader för företaget (R5, rad 72). Respondent 3 anser att hur förödande konsekvenserna blir för systemet, utvecklingen och underhållsarbetet beror på hur djupa ändringarna av systemet är. Om det är en del av användargränssnittet som inte har blivit dokumenterat kan detta bidra till en dålig web, samt förlorad inkomst, men det är inte lika allvarligt som ändring av en huvudkomponent, som ligger långt ner i värdekedjan (R3, rad 53). Om en sådan ändring inte dokumenteras kan det bli fel i hela back-end systemet, vilket är riktigt illa och kan resultera i teknisk skuld (R3, 53).

## 4.4 Dokumentationskvalitet

Respondenterna ser lite olika på vad som motsvarar hög kvalitetsnivå. Respondent 2 och 5 tycker att det är viktigt att den är bland annat lättläst, tydlig och enkel att förstå (R2, rad 121; R5, rad 75) medan respondent 1 anser att strukturen är det viktigaste (R1, rad 87). Likt respondent 2 och 5, menar respondent 6 att lättillgänglighet, tillförlitlighet och enkelheten till att använda och uppdatera dokumentationen motsvarar hög kvalitet (R6, rad 69). Både

respondent 4 och 5 tycker att bilder som visar systemsambanden bidrar till tydligheten (R4, rad 80; R5, rad 77). Däremot säger respondent 3 att fullständiga dokument och hur väl den är skriven för andra utvecklare på samma kunskapsnivå är det viktigaste för kvaliteten (R3, rad 57). Respondent 4 tycker snarare att testdata och testfall är den viktigaste delen i dokumentationen (R4, rad 84).

Förväntningarna är olika, där vissa menar att de har dessa kvalitetsattribut som krav och andras förväntningar är betydligt lägre. Praktiskt hos de flesta respondenterna uppnås däremot sällan eller aldrig en hög kvalitet på dokumentationen (R2, rad 125; R3, rad 59; R4, rad 84, R5, rad 85; R6, rad 71).

En viktig del av dokumentationskvaliteten som lyfts av flera respondenter är vikten av spårbarhet. Spårbarheten hanteras bland annat genom versionshanteringssystem (R4, rad 95). Det är viktigt att kunna spåra vem som gjort ändringarna och när de gjorts (R1, rad 63; R2, rad 134; R5, rad 103) eftersom det bidrar till att utvecklare kan gå till personen i fråga och be om hjälp (R1, rad 63). Det kan vara viktigt för lärandet (R1, rad 83). Det blir även enklare att hitta den person som är mest lämpad för att utföra ändringar i systemet när spårbarheten är hög (R4, rad 95).

#### 4.4.1 *Korrekthet*

Att ett dokument är korrekt är viktigt (R2, rad 128; R3, rad 64; R4, rad 86; R6, rad 79) och en förutsättning, annars finns det ingen anledning att läsa det (R1, rad 91). Det viktigaste är att systemdokumentationen inte är skriven på modersmålet, eftersom detta gör det svårt för andra delar av organisationen, som inte arbetar på modersmålet, att ta del av dokumenten (R1, rad 91). Således är en del av korrekthet att all dokumentation ska vara skriven på engelska (R1, rad 93; R3, rad 64; R5, rad 90).

Ju mer korrekt desto snabbare kommer utvecklaren in i arbetet och blir villigare till att utföra underhållet (R4, rad 86). Däremot menar respondent 3 att med korrekthet kommer många sidor text, vilken användarna inte orkar läsa. Istället väljer utvecklarna att testa sig fram vilken resulterar i att allt tar längre tid med en längre inlärningsperiod (R3, rad 64).

#### 4.4.2 *Läsbarhet*

Läsbarheten i form av grammatik och stavning, anses av de flesta respondenterna inte vara viktigt för användbarheten (R1, rad 95; R4, rad 89; R5, rad 90; R6, rad 81). Respondent 4 menar att namnkonventioner är viktigare och att dessa ska vara konsekventa genom både dokumentation och system (R4, rad 91). Respondent 2 och 3 tycker dock att det är viktigt (R2, rad 121; R3, rad 66) och respondent 3 fyller i med att utvecklare om några ska förstå värdet i läsbarhet, eftersom det ofta talas om läsbar kod. Det borde därför vara ett krav på läsbara kommentarer och dokument (R3, rad 66).

#### 4.4.3 *Struktur*

Strukturen är värdefullt (R1, rad 87; R5, rad 97), men ska inte utformas av företagets utvecklare utan snarare av specialiserade företag, högskolor eller universitet (R1, rad 87). Det liknas vid respondent 2:s resonemang om kvalitetssystem samt respondent 3:s användningsområde för

programvaror (R3, rad 68). I ett kvalitetssystem finns formalia och strukturering på hur ett dokument ska se ut (R2, rad 134). Det finns flera programvaror som styr strukturen, bland annat DevOps och Jira (R3, rad 68).

Dessvärre följer strukturen inte respondenternas förväntningar (R1, rad 89; R2, rad 125; R4, rad 84; R6, rad 71). I respondent 5:s team uppnås förväntningarna till största del, men det beror på att det är hens ansvar att se till att det blir så (R5, rad 85). När det gäller andra är det både högt och lågt (R5, rad 85).

Om bestämd struktur inte efterföljs, finns det en risk för att utvecklare skriver alldeles för långa dokument (R1, rad 87). Med en bra standard är det enkelt att söka efter och uppdatera information i dokumenten (R4, rad 93). Tyvärr finns det inte alltid en struktur att följa, men det blir viktigare ju större systemen blir (R6, rad 84).

#### 4.4.4 Fullständighet

I teorin strävas det efter komplett systemdokumentation men i praktiken är det bättre att få lite än inget alls (R6, rad 87). Respondent 2 anser att så länge ett dokument inte är färdigställt så är det endast ett utkast (R2, rad 136), medan respondent 4 och 5 inte tycker att fullständiga dokument in i detalj är viktigt (R4, rad 97; R5, rad 105). Det som snarare är vitalt för utvecklaren är den översiktliga förståelsen (R4, rad 97; R5, rad 105). Enskilda metoder går istället att läsa i källkoden (R3, rad 97).

Respondent 1, 2 och 3 ser snarare fullständighet som subjektivt. Den ska vara så pass fullständig och tillräcklig för att en viss typ av mottagare ska förstå den (R3, rad 77). Är den skriven åt någon annan än sig själv, ska den vara mer noggrann (R1, rad 105). Om mottagaren har mindre arbetserfarenhet och kunskap, ska den ha mer text (R3, rad 83). Är det däremot utvecklaren som kan systemet sedan innan, spelar innehållet mindre roll (R2, rad 118).

Vid arbete med mindre system behöver otillräcklig information inte vara ett problem eftersom det förmodligen finns möjlighet att prata med personen som utvecklade systemet (R1, rad 111). I större sammansatta grupper är det däremot systemarkitektens uppgift att hålla samman dokumenten (R1, rad 111). Om dokumentationen är bra och välarbetad leder detta också till betydligt bättre arbete (R1, rad 111).

Den interna systemdokumentationen innehåller dock sällan all information som krävs (R3, rad 80; R6, rad 93) och det finns stora förbättringsmöjligheter (R4, rad 101). Startsträckan är ofta lång (R6, rad 93) och systemutvecklare behöver oftast själva lista ut hur systemet fungerar eftersom innehållet ofta är bristfälligt (R4, rad 101). Däremot har det blivit enklare att hitta information på annat håll i och med Google-generationen (R3, rad 80). Respondent 4 tycker däremot att det är onödigt att lägga ner tid och pengar på att göra den perfekta dokumentationen, men ju bättre dokumentationen är desto mer tjänar du på det i framtiden (R4, rad 104).

#### 4.4.5 Konsekvenser

Konsekvenserna av dålig kvalitet inom intern systemdokumentation anser respondent 1 beror på utvecklarna, samt hur kompetenta de är (R1, rad 114). Allt hänger helt enkelt på om en utvecklare kan se igenom dålig dokumentation, välja bort den, eller kan välja ut de delar av innehållet som kan användas i utvecklingen (R1, rad 114). Dock blir det ju enklare för alla om

det finns bra dokumentation (R1, rad 114). Respondent 2 anser att svaret är detsamma som på för uppdatering; om en person redan kan systemet framlänges och baklänges så är det möjligt att utveckla eller ändra systemet ändå (R2, rad 140). Problemet uppstår främst när medlemmar byts ut inom ett team, blir sjukskrivna eller går i pension och nya människor ska behöva sätta sig in i någonting som de inte kan och som inte stämmer (R2, rad 140). Dålig kvalitet bidrar också till att ett dokument kommer vara svårt att läsa och förstå, vilket gör att arbetet tar längre tid samt att sannolikheten för fel höjs (R4, rad 112).

Respondent 3, 5 och 6 ser desto allvarigare på dålig kvalitet. Respondent 5 menar att om utvecklare missförstår systemdokumentation bidrar detta till att fel slutsatser dras, vilket kommer skapa fel i systemet, samt högre kostnader för företaget (R5, rad 125). Respondent 6 anser att systemdokumentation med dålig kvalitet kommer bidra till en stor mängd spill och att utvecklare kommer implementera någonting som inte fungerar (R6, rad 100). När det sedan blir alldeles för dyrt att försöka ändra på felen kommer företag istället välja att ägna dem resurserna på andra arbetsuppgifter och således skapas teknisk skuld, som kommer ligga kvar i företaget för alltid (R6, rad 100).

## 4.5 Tillgänglighet

Av intervjuerna av respondenterna framkom ett antal olika problemområden inom tillgänglighet av intern systemdokumentation, som kommer redogöras för nedan.

### 4.5.1 Åtkomstrestriktioner

Respondent 1, 3, 4 och 5 anser allihop att det är enkelt att få tag i intern systemdokumentation när de behöver det, mycket på grund av att företaget har ett internt system där det är möjligt att dela dokument (R1, rad 116; R3, rad 93 & 96, R4, rad 115; R5, rad 129 & 136). Dock är tillträde för olika roller ett problem. Respondent 1 berättar att alla medarbetare inte har tillgång till alla olika dokument, snarare får en medarbetare tillgång till rätt dokument när hen får ett uppdrag kopplat till dokumentet (R1, rad 118). Även respondent 3 förklarar att det inte är svårt med tillgängligheten så länge den anställdes roll är systemutvecklare (rad 91). Om den anställda har en annan roll är det desto svårare då det kräver godkända lösenord och användarnamn för att nå olika dokument (R3, rad 91). Respondent 5 anser att det är enkelt att få tag i systemdokument inom hens IT-avdelning, men berättar att R&D-avdelningen håller sina dokument nedlåsta vilket gör dem desto svårare att få tag i (R5, rad 129). Vidare anser hen också att tillgänglighet även handlar om hur enkelt det är att ta till sig informationen som står i dokumentet; långa texter, som tar lång tid att läsa är inte lättillgängliga (R5, rad 85).

Respondent 2 och 6 tycker att tillgängligheten av systemdokumentation beror på var dokumentet finns sparad. Båda parter anser att det är enkelt att få tag i dokumentation om det ligger på företagets Wiki-sida, men att det ofta är så att medarbetare sparar ner det i egna mappar eller system som respondenterna inte kommer åt (R2, rad 144; R6, rad 107).

### 4.5.2 Sökning, namngivning och indexering i dokumenthanteringssystemet

Både 1 och 2 önskar att deras företag haft ett bättre dokumenthanteringssystem där det hade varit möjligt att ladda dokument, revidera, söka och skapa index (R1, rad 126; R2, rad 144).



Respondent 2 tar detta ytterligare lite längre och förklarar att tillgängligheten hade blivit ännu bättre om det hade varit möjligt att lägga till ett ID, taggar och titel i dokumenten, samt att det skulle finnas möjlighet att söka på dessa attribut (R2, rad 144). Resterande respondenter är nöjda med hur deras systemdokumentation sparas idag. På respondent 3:s företag används Teams eller Git (R3, rad 96), respondent 5 använder sig av ett hanteringssystem som heter Confluence, men har också en SharePoint (R5, 136), nummer 6 använder Git och Mark Down (R6, rad 109) och respondent 4 använder sig av en Wiki (R4, rad 115). Respondent 4 och 6 är ense om att systemdokumentation som sparas i Word-dokument inte bidrar med någon nytta (R4, rad 115; R6, rad 109).

#### 4.5.3 *Versionshantering*

När det gäller versioner av systemdokument säger alla respondenter att både nuvarande och äldre versioner sparas (R1, rad 128; R2, rad 151; R3, rad 101; R4, rad 135; R5, rad, 150; R6, rad 111). Fyra av respondenterna är eniga om att det är enkelt att förstå vilken som är mest relevant (R1, rad 130; R3, rad 103; R4, rad 133; R4, rad 48 & 50) då deras dokumenthanteringssystem antingen tydligt visar den senaste versionen genom att lägga den på en förstasida (R4, rad 119) eller genom versionsnamn i form av siffror (R1, rad 130). Dock anser både respondent 2 och 6 att versionshantering inte fungerar som det borde och att en del förvirring ofta uppstår. I respondent 2's företag döps inte dokumenten till tillräckligt tydliga namn, vilket ofta resulterar i många dokument som är överflödiga och sällan blir använda (R2, 151). Problemet som uppstår i respondent 6:s företag är att alla medarbetare inte alltid har tillgång till samma version av ett dokument, vilket har orsakat problem då underkonsulter har suttit och arbetat med en kopia av ett gammalt dokument (R6, 115).

#### 4.5.4 *Konsekvenser*

På frågan om hur begränsad tillgänglighet påverkar systemutveckling, underhållsarbete och systemet svarar respondent 1 att om en systemutvecklare inte får tillgång till rätt dokument kan detta hindra personen från att arbeta med systemet (R1, rad 135). Respondent 4 håller med genom att förklara att begränsad tillgänglighet kommer kräva att det tar längre tid för en utvecklare att förstå systemet, då de kommer behöva sätta sig ner för att lära sig systemets utformning eller problematik (R4, rad 147). Respondent 2 håller med till viss del. Hen menar på att en ny eller oerfaren utvecklare kommer få problem om systemdokumentationen inte är tillgänglig, men en erfaren systemutvecklare kommer kunna arbeta med systemet ändå (R2, rad 153). Om en utvecklare inte når nödvändig dokumentation ökar det risken för att hen försöker implementera funktioner som hen inte förstår, vilket kommer resultera i systemfel (R4, rad 147). Respondent 1 och 6 anser även att en person som inte har tillgång till ett dokument inte heller kommer besitta rätt kunskap, vilket försvårar utveckling av systemet (R1, rad 135; R6, rad 119). Respondent 5 anser att ju fler dokument som hade varit tillgängliga desto mer hade medarbetare kunnat utvecklas samt hjälpa till med andra arbetsuppgifter (R5, rad 155). Två respondenter anser att tillgänglighet kan bidra till ökad kunskapsdelning då det ger en möjlighet att se vem det är som har gjort vad och sedan gå och diskutera materialet eller ställa frågor (R1, rad 132), men även att en medarbetare själv kan sitta och söka på information den vill ta del av, vilket också är en form av kompetensöverföring (R4, rad 142).

Respondent 6 och 2 anser att om teamet är tillräckligt litet så kommer systemet fortsätta utvecklas ändå, men att problem kommer uppstå när äldre, kunniga utvecklare försvinner och nya utvecklare tillkommer (R2, rad 153; R6, rad 119). Det är dock svårt att säga om detta

kommer bidra till att kvaliteten på systemet ökar eller minskar, men underhåll av systemet kommer ta desto längre tid, då utvecklare som i teorin borde vara kvar inte längre är det (R6, rad 119). Detta går i linje med vad respondent svarade 5. Hen förklarade att projekt som inte får tillgång till rätt systemdokumentation kommer bli försenade, och den extra tiden som läggs på utvecklings- eller uppdateringsarbetet kommer innebära högre kostnader för företaget (R5, rad 157). Otillgänglig dokumentation kommer även frånta utvecklare chansen att utvecklas och lära sig ännu mer, samt hjälpa till med andra projekt eller uppgifter (R5, rad 157).

## 5 Analys och diskussion

*I detta avsnitt diskuteras det empiriska resultatet tillsammans med den data som presenterats i avsnittet om tidigare forskning. Syftet är att analysera informationen och hitta liknelser, skillnad, samt eventuella gap mellan insamlade intervjudata och undersökt litteratur.*

### 5.1 Intern systemdokumentation

Respondent 1 tycker att någon typ av intern systemdokumentation finns är en förväntning för att överhuvudtaget kunna utföra systemutvecklingsarbete. Intern systemdokumentation ska även bidra till att alla uppgifter kring systemet sker så effektivt och sömlöst som möjligt (Garousi et al., 2014). I linje med det, tycker respondent 4 att intern systemdokumentation förenklar arbetet för alla inblandade, framförallt när nya resurser sätts på nytt arbete. Då bidrar intern systemdokumentation med en stadigare bas. För att en stadig bas ska vara möjlig, säger Hunt & Thomas (2000) att dokumentationen behöver vara uppdaterad och lättillgänglig, vilket stämmer överens med respondent 5:s uttalande om att det är viktigt att dokumentationen hen använder sig av är just det, lättillgänglig och enkel att uppdatera.

För att intern systemdokumentation ska kunna bidra till ett effektivare arbete, menar däremot respondent 2 att det måste finnas ett ramverk, ett kvalitetssystem, att följa och att det inte ska vara ett fritt tänkande kring hur och om arbetet ska göras. Även respondent 3 håller med om att effektiviteten som går att utvinna ur dokumentationsarbetet ligger i organisationens händer. Det går i riktning med Lethbridge, Singer & Forward (2003) som säger att ledningen bör sätta standarder för att motivera till dokumentationsarbete, men framförallt för att uppmana till att uppdatera dokumenten. Med ett kvalitetssystem menar respondent 2 att osämja kring arbetet med systemdokumentation inte behöver finnas. Osämja uppstår snarare när ett bestämt ramverk inte finns. Då blir det lätt att skilda åsikter framkommer som gör ett gemensamt arbetssätt svåråtkomligt.

Generellt sett är intern systemdokumentation ofta bristfälligt och respondent 3, 4 och 6 tycker att all den kunskap som krävs för att kunna utföra underhållsarbetet sällan finns med, vilket leder till att startsträckan blir lång. Lethbridge, Singer & Forward (2003) anser också att det finns mycket problematik kring arbetet med intern systemdokumentation och att arbetet ofta väljs bort. Respondent 2 menar att det beror på att arbetet inte går att utföra ordentligt om ett kvalitetssystem inte används, vilket är något hens företag inte gör. Respondent 4 styrker Lethbridge, Singer & Forward (2003) genom att tillägga att arbetet kring intern systemdokumentation är mycket slarvigt.

#### 5.1.1 Kunskapshantering inom organisationer

Inom samtliga team och företag, har det aktivt valts att kunskapsdela med hjälp av konversationer eller överlämningsprojekt, vilket går emot Forward & Lethbridge (2002) sägande om att intern systemdokumentation är det effektivaste sättet att förvalta den kunskapen som finns inom det egna företaget. Däremot säger både respondent 1 och 5 att på grund ut av att deras företag är stora runt om i världen, finns det någon grad av systemdokumentation för att kunna sprida kunskapen internationellt.

De Boer och van Vliet (2009) har kommit fram till att kunskap i utvecklingsstadiet kan delas i person mellan kollegor, men att den mesta kunskapen inom utvecklingsteamet bör delas med hjälp av intern systemdokumentation. Som förstått, gör inget av företagen som rekommenderat. De Souza, Anquetil, & de Oliveira (2005), Lethbridge, Singer & Forward (2003) och Stapel & Schneider (2014) anser att dåligt skriven dokumentation bidrar till att viktig kunskap går förlorad, vilket respondent 1 lyfter när hen säger att de utan intern systemdokumentation lätt förlorar kunskap när en kunskapsbesittande person lämnar företaget. Majoriteten av respondenterna säger sig ha strategier för att bevara kunskap, trots det uttrycker även resterande av respondenterna att mycket kunskap går förlorad när en anställd slutar. Henriks (1999) menar att det är viktigt för organisationer att förvalta kunskap för att bli konkurrenskraftiga. Intern systemdokumentation kan därför användas för att säkerställa att kunskapen förvaras (Lethbridge, Singer & Forward, 2003; Stapel & Schneider, 2014).

Respondent 2 anser även att en del personer är mindre villiga till att dela med sig av sin kunskap eftersom de personligen kan vara konkurrenskraftiga gentemot andra systemutvecklare. Framförallt ter det sig på så sätt inom konsultbranschen där det är enklare att ta betalt mer för sin kunskap och erfarenhet. Även Cabrera & Cabrera (2002) tar upp att det finns systemutvecklare som är motstridiga eftersom de inte vinner på att dela med sig av sin kunskap. Respondent 2 backar upp med att sådana konsulter ofta ser till att de är dem enda som förstår sig på och kan ändra i systemet för att se till att det är dem som måste tas in vid underhållsarbete.

## 5.2 Uppdatering

I tidigare studier finns det delade åsikter kring hur uppdatering av intern systemdokumentation påverkar systemutvecklingsarbetet där en del studier påvisar att föråldrade dokument fortfarande är användbara (Lethbridge, Singer & Forward, 2003; Forward & Lethbridge, 2002), medan andra anser att uppdaterade dokument är mycket viktigt (Selic, 2009; Stapel & Schneider, 2014; Sulaiman, Idris & Sahibuddin, 2002). Respondenterna tycker generellt att uppdaterad systemdokumentation är vitalt för att kunna utveckla, underhålla och förstå ett system, samtidigt som hälften av respondenterna tycker att föråldrad dokumentation kan ge en användbar, övergripande förståelse för systemet. Trots detta är intern systemdokumentation väldigt sällan lika uppdaterad som det nuvarande systemet (Hunt & Thomas, 2000; Lethbridge, Singer & Forward, 2003; Forward & Lethbridge, 2002). Detta faktum stärks av respondenterna som, alla utom en, förklarar att de sällan arbetar aktivt med uppdateringsarbete. Det beror på att det inte finns en samstämmig anledning till att uppdatering bör genomföras – det finns inget krav på att utvecklare ska skriva intern systemdokumentation.

### 5.2.1 Uppdatering beroende på kodändringens storlek

En bidragande faktor till ett varierande uppdateringsarbete, men även hur tillförlitlig dokumentationen är, är storleken på kodändringarna som gjorts. Två av respondenterna anser att större kodändringar bör prioriteras eftersom dessa kodändringar har förmågan att ändra systemets huvudfunktion. Större ändringar påvisar dock en motsträvighet hos utvecklare och det är oftast dessa tyngre ändringar som gör att dokumentation inte uppdateras ordentligt (Lethbridge, Singer & Forward, 2003). Två andra respondenter går i led med Lethbridge, Singer & Forward (2003) och anser att mindre kodändringar istället är de som bör uppdateras eftersom dessa är enklare och snabbare att utföra. Respondent 2 styrker påståendet genom att förklara att

större ändringar gärna åsidosätts av utvecklare eftersom det är ett för tidskrävande arbete. Systemutvecklare tilldelas ett visst antal timmar för att utföra omskrivning av källkoden och kommer alltid att prioritera att denna kod blir färdigställd snarare än att uppdatera dokumentationen (R2). Sulaiman, Idris & Sahibuddin (2002) har genom studie däremot kommit fram till att uppdateringar av småändringar betyder att utvecklare konstant kommer tvingas ändra i systemdokumentationen, vilket både är kostsamt och tidskrävande (Sulaiman, Idris & Sahibuddin, 2002). De två sista respondenterna anser däremot att storleken på kodändringen inte är det som spelar roll. Det är snarare att större kodändringar uppmärksammas mer (R6) eller att det är kodändringens funktionalitet som prioriteras (R5).

Eftersom åsikterna är delade, finns det inget givet samband mellan vare sig litteratur eller respondenter som påvisar att en viss kodändring kommer bidra till mer eller mindre uppdaterade dokument. Däremot, i de team där olika ändringsstorlekar prioriteras, kommer de ändringar som inte prioriteras att vara sämre, eller inte alls, uppdaterade i dokumentationen. I riktning med respondent 5:s tyckande om att det är kodens essentiella funktionalitet som ska prioriteras, kan sämre dokumentation oberoende ändringens storlek, bidra till att betydelsefull funktionalitetsdokumentation går förlorad. När en ändring utförs i koden, behöver den även uppdateras i dokumentationen, annars blir resultatet dokumentationsskuld (Guo & Seaman, 2011). Två av respondenterna uttrycker däremot att det påträffas dokument som inte är uppdaterade enligt aktuell status på koden, men att de ändå bringar den nytta som önskas eftersom rätt funktionalitet är dokumenterad. I dessa fall, leder föråldrad intern systemdokumentation inte till dokumentationsskuld (Guo & Seaman, 2011). Även om detta förkommer, så är det mycket sällan och slutsatsen kan därför dras att föråldrad dokumentation, i detta anseende, leder till dokumentationsskuld.

### 5.2.2 Anledning till bristfälligt uppdateringsarbete

En av respondenterna uppdaterar framförallt dokumentation för egen vinning och för att komma ihåg vad hen själv har gjort, däremot sker det inte ofta. Lethbridge, Singer & Forward (2003) och Cabrera & Cabrera (2002) förklarar i samspel att utvecklare endast prioriterar den systemdokumentation de anser värdefull för den egna personen.

Tre andra respondenter, å andra sidan, prioriterar nyttan för gruppen snarare än för individen. Respondent 2 förklarar att hans team uppdaterar den dokumentation som kommer bringa utvecklingen mest nytta. Litteratur speglar att en sådan grundtanke påverkar intern systemdokumentation positivt eftersom utvecklare endast behöver göra uppdateringsarbete när behovet finns och ändringen faktiskt kommer användas (Forward, 2002). Samtidigt är det svårt för respondenterna att tydligen se nyttan som bringas av att använda sig av intern systemdokumentation och fyra av respondenterna läser hellre källkoden än dokumenten. Respondent 2 utvecklar med att det inte finns något som sporrar utvecklare att faktiskt utföra intern systemdokumentation och andra mer akuta behov prioriteras istället.

Som en andra anledning, menar respondent 1, 2 och 6 att det inte finns tid prioriterat för arbetet. För att tid ska finnas, behöver den betalas för, vilket inte är något som företag själva vill stå för, utan det är snarare kundens uppgift. Respondent 1 menar istället att systemutvecklare får betalt för att utveckla systemet, inte för att skriva om det. Även Cabrera & Cabrera (2002) och Forward & Lethbridge (2002) trycker på dokumentationsarbetet är en aktivitet som kan anses vara för dyr.

Då respondenterna har svårt att se nyttan av uppdateringsarbete till den grad att de faktiskt utför det, i samband med att det är ett arbete som inte prioriteras i form av tid och pengar, resulterar det i dokumentationsskuld eftersom det hindrar utvecklare från att kunna utföra olika typer av systemutvecklingsarbete (Mendes et al., 2016)

### 5.2.3 *Användandet av föråldrad dokumentation*

Tidigare forskning (Selic, 2009; Stapel & Schneider, 2014; Sulaiman, Idris & Sahibuddin, 2002). trycker på att den målgrupp som uppdaterade systemdokument bidrar till störst nytta för, är utvecklare som utför underhållsarbetet, vanligtvis nyexaminerade utvecklare med mindre erfarenhet och kunskap.

För mindre erfarna utvecklare, så väl som för andra utvecklare, kommer föråldrad intern systemdokumentationen bidra till att mycket dyrbar tid kommer behövas spenderas på att bekanta sig med systemet, dess funktioner, utvecklingsspråk och problem (Tom, Aurum & Vidgen, 2013). Alla respondenterna håller med om att ett visst fördröjsmål tillkommer när någon ska ta sig an ett nytt system och dokumenten inte är uppdaterade. Däremot lägger fem av dem istället mycket vikt på att läsa källkod. Å andra sidan är respondenterna ense om att det är viktigt att uppdatera intern systemdokumentation när en kunskapsbesittande person lämnar företaget. Det beror på att muntlig kommunikation inte längre blir möjligt och kunskap går förlorad.

Hälften av respondenterna talar i linje med litteraturen (Selic, 2009; Stapel & Schneider, 2014; Sulaiman, Idris & Sahibuddin, 2002) och menar att nya, mindre erfarna utvecklare inte kommer kunna uträtta uppdateringsarbete om det inte finns tillräckligt uppdaterad intern systemdokumentation med rätt kunskap. Mer erfarna utvecklare kommer däremot klara sig bättre trots att dokumentationen inte är uppdaterad, eftersom de efter många år med olika erfarenheter tagit del av mycket kunskap samt kan ha arbetat med systemet sedan innan. Antingen behöver de inte använda dokumentationen alls, eller så kan dokumenten ge en övergripande bild som bidrar med tillräcklig förståelse. Även tidigare studier påvisar att föråldrad dokumentation kan te sig användbart genom att förse utvecklare med en övergripande förståelse för systemet och dess funktioner (Lethbridge, Singer & Forward, 2003; Forward & Lethbridge, 2002; Sulaiman, Idris & Sahibuddin, 2002). De tre respondenterna fortsätter med att föråldrade dokument fortfarande bidrar till att utvecklare vet vem som skapat både systemet och dokumentet och därför kan ta del av kunskapen genom att ta kontakt med utvecklaren. På så vis behöver kunskapen inte gå helt förlorad.

När utvecklare, framförallt nyexaminerade, inte kan utföra sitt arbete som följd av föråldrad dokumentation som saknar den kunskap som krävs, har intern systemdokumentation inte uppfyllt sitt syfte och istället satt hinder för att utvecklaren ska kunna underhålla systemet. Enligt definition betyder det att dokumentationsskuld uppstår (Mendes et al., 2016).

### 5.2.4 *Konsekvenser av dokumentationsskuld*

Ur föråldrad dokumentation kan dokumentationsskuld leda till flera konsekvenser som påverkar både systemet, individen och företaget.

Den mest omfattande konsekvensen som egentligen ligger till grund för resterande konsekvenser, är inverkan på kunskapsförvaltningen i organisationen. Respondenternas svar

genomsyras av att föråldrad dokumentation oftast leder till att kunskap antingen går förlorad, blir tvetydig eller otillräckligt för att utvecklingsarbetet ska kunna fortgå obehindrat. Inom samtliga företag, ter sig kunskapsdelningen snarare via muntliga konversationer, vilket Tang, Liang & van Vliet (2011) menar kan vara riskfyllt eftersom utvecklare lätt förlitar sig på dessa konversationer för att delge kunskap. Respondenterna uttrycker att detta främst förekommer när viktiga personer lämnar företaget, samt när nyexaminerade utvecklare stiger in i branschen. I dessa fall kan mindre erfarna utvecklare misslyckas att följa den etablerade utvecklingsstandard som finns på företaget (Guo & Seaman, 2011) och således kommer arbetet gå långsammare och systemet bli sämre (R6). Ett sådant arbete menar respondent 3 i slutändan kommer att kosta mer pengar än om uppdaterad dokumentation hade kunnat användas.

När föråldrad dokumentation innehåller otillräcklig kunskap om systemet, kan systemutvecklare – oberoende erfarenhet, ålder eller vana av systemet – dra fel slutsatser och basera sitt utvecklings- eller underhållsarbete på felaktiga konstateranden (R5). Således kan andra system, som är beroende av systemet som ändras, också blir felaktiga, något som kommer resultera i stora kostnader för företaget (R5). För att motverka denna konsekvens bör alla ändringar som görs i systemet även uppdateras i tillhörande intern systemdokumentation (Guo & Seaman, 2011), vilket respondent 5 även håller med om.

Tre respondenter uttrycker att startsträckan inför att börja utvecklingsarbetet blir lång, kostsam och tidskrävande innan det ens är möjligt att börja programmeringsarbetet. Respondent 4 säger att detta påverkar systemutvecklingen genom att utvecklare till slut inte vågar gå in och ändra i systemet eftersom förståelsen och kunskapen kring hur det fungerar inte existerar. Slutligen tappar utvecklarna kontrollen över systemet.

Respondent 3 anser att hur förödande konsekvenserna blir för systemet, utvecklingen och underhållsarbetet beror på hur djupt ner i värdekedjan som ändringarna sker. Om en mindre betydande komponent blir felaktig till följd av föråldrad systemdokumentation, kan detta kosta företaget pengar, men konsekvenserna kommer inte bli allvarliga. Däremot, om ändringen utförs på en huvudkomponent, som ligger längre ner i värdekedjan, kan konsekvenserna bli paralyserande. Respondent 3 fortsätter med att om en sådan ändring inte dokumenteras kan det bli fel i hela back-end systemet, vilket är riktigt skadligt och kan resultera i teknisk skuld.

### 5.3 Dokumentationskvalitet

Av tidigare forskning har det framkommit att dokumentationskvalitet anses vara en av de viktigaste faktorerna vid utformning av intern systemdokumentation (Garousi et al., 2014; Plösch, Dautovic, & Saft, 2014) eftersom det bidrar till ökad förståelse om systemet i dess utveckling-, användning- och uppdateringsstadier (Briand, 2003). Respondenterna har dock alla olika syn på vad som motsvarar hög kvalitetsnivå, vilka förväntningar det finns på kvaliteten, samt till vilken grad det faktiskt påverkar utformning och användning av systemdokumentation. En del håller med tidigare studier (Garousi et al., 2014; Plösch, Dautovic, & Saft, 2014) om att kvalitet är betydande för systemdokumentationens övergripande förståelse, medan andra är desto mer tveksamma. Plösch, Dautovic, & Saft (2014) får dock medhåll från samtliga respondenter om att förväntningar på dokumentationskvaliteten sällan uppnås. För att undersöka synen på dokumentationskvalitet, samt hur och om det leder till dokumentationsskuld har följande kvalitetsattribut diskuterats: korrekthet, läsbarhet, struktur och fullständighet.

### 5.3.1 Korrekthet

Aghajani et al. (2019) och Plösch, Dautovic, & Saft (2014) menar att *korrekthet* är det kvalitetsattribut som väger tyngst. Även fem av respondenterna tycker att det är viktigt och till och med en förutsättning att dokumentationen är korrekt. En av respondenterna påpekar att om systemdokumentationen inte är korrekt är det inte värt att lägga ner dyrbar tid på att försöka läsa dokumentationen. Korrekt systemdokumentation bidrar till att en systemutvecklare snabbare kommer in i arbetet och är således mer villig att utföra underhållsarbete (R4). Respondent 3 tycker däremot att om korrektheten överarbetas, resulterar det i att dokumentationen blir väldigt lång. När systemdokumenten består av mycket text kommer utvecklare bortse från dokumentationen och istället testa sig fram vilket bidrar till utdragen tid. Aghajani et al. (2019) stödjer att problematik med korrekthet kan leda till ödslad tid.

### 5.3.2 Läsbarhet

Aghajani et al. (2019) uttrycker att utvecklare fortfarande kan dra nytta av bristfällig *läsbarhet*. Det styrker respondenterna då fyra av dessa inte tycker att läsbarhet i form av grammatik och stavning är viktigt för användbarhet. Däremot anser respondent 2 och 3 att det är essentiellt och respondent 3 menar att utvecklare om några borde förstå värdet i läsbarhet eftersom det ofta tals om läsbar kod.

### 5.3.3 Struktur

Tre av respondenterna anser att en bestämd *struktur* är viktig, men ofta har ett företag ingen definierad struktur för vad intern systemdokumentation ska innehålla och vilka utformningsregler som ska efterhållas (Briand, 2003). Hälften av respondenterna förklarar att deras företag inte har en definierad standard över hur utformning av systemdokumentation ska se ut. Majoriteten anser också att strukturen nästintill aldrig följer de förväntningar som finns. Respondent 1 tycker att om bestämd struktur inte efterföljs, finns det en risk för att utvecklare skriver alldeles för långa dokument. Långa dokument, anser respondent 3 leda till att utvecklarna inte kommer läsa dem och istället testa på eget bevåg. Precis som när korrektheten är bristfällig, leder arbete på eget bevåg till ett utdraget underhållsarbete. Även här går det i enlighet med Aghajani et al. (2019) ord om ödslad tid.

### 5.3.4 Fullständighet

Briand (2003) framför att det har blivit en norm för utvecklare att använda sig av ofullständig dokumentation. På så sätt, anser fem av respondenterna att *fullständighet* inte behövs i de flesta sammanhang. Det som snarare är viktigt är den översiktliga förståelsen och att enskilda metoder istället kan läsas ur källkoden. Det här går i hand med Garousi et al. (2014) som kommit fram till att det i de flesta företag inte anses rimligt att färdigställa all dokumentation eftersom det bidrar till stora kostnader. Istället menar Garousi et al. (2014) att dokumentation ska skapas till den grad att komplexa funktioner och aktiviteter som ska utvecklas får stöttning nog att bli förstådda, vilket fyra av respondenterna även har uttryckt. Respondent 3 uttrycker det genom att dokumentation ska vara så pass fullständig och tillräckligt för att en viss typ av mottagare ska förstå innehållet.



### 5.3.5 *Konsekvenser av kvalitetsattribut*

Plösch, Dautovic, & Saft (2014) har kommit fram till att dokumentationskvalitet är en av de viktigaste aspekterna av intern systemdokumentation, och Briand (2003) menar att det bidrar till ökad förståelse och kunskap om systemet. Eftersom respondenterna vet att sämre kvalitet gör dokumenten svåra att läsa och förstå, ser de generellt att hög kvalitet bidrar till enklare arbete. Trots det ser de inte värdet i alla kvalitetsattribut. Attributen påverkas mycket av individuellt tycke, samt vilken erfarenhet systemutvecklaren har. Respondent 1 menar att hur allvarliga konsekvenserna blir av dålig dokumentationskvalitet beror på hur kompetent läsaren är; om hen kan se igenom den dåliga dokumentationen, bortse från felaktigheter eller ta del av användbara delar. Detta ter sig framförallt inom läsbarhet och struktur. Eftersom majoriteten av respondenterna inte anser att läsbarhet är viktigt för användbarheten, kan slutsatsen inte dras att attributet skulle leda till dokumentationsskuld. Även respondenternas uttalanden om dokumentens struktur är tvetydigt eftersom hälften av respondenterna ansåg det viktigt och andra hälften mindre viktigt. På grund av det kan vi inte heller dra någon slutsats om att dålig struktur leder till dokumentationsskuld. Däremot kan båda attributen leda till olika konsekvenser beroende på användaren. Utvecklare med mindre erfarenhet och kunskap kommer ha svårare att följa dokumentation med bristande läsbarhet eller struktur. I dessa fall leder det däremot framförallt till att det tar längre tid för utvecklarna att ta sig an dokumentation, snarare än att fel görs i koden.

Attributen korrekthet och fullständigheter är däremot mer kritiska. Eftersom fem av respondenterna anser att korrektheten är viktig för att det ska vara av nytta att läsa den, kan konsekvenserna även te sig värre när dokumentationen inte är korrekt eftersom utvecklarna antingen får fel uppfattning av dokumenten eller att de bortser från dokumentationen överhuvudtaget. I samband med att respondenterna inte anser att fullständig dokumentation behövs i de flest sammanhang, tycker fyra av dessa att dokumentation ofta saknar innehåll som krävs för underhållsarbete. Den konsekvens som lyfts mest är att startsträckan blir längre. Det beror på att avsaknad av kunskap och förståelse som fullständig dokumentation bidrar med, gör att dokumentationen blir oanvändbar. Det går i riktning med de Souza, Anquetil, & de Oliveira (2005) som menar att om nödvändig information saknas, kommer dokumentationen inte uppnå sitt syfte och bli oanvändbar i flera scenarion. Däremot är hälften av respondenterna överens om att ofullständig systemdokumentation kan te sig användbart då det är möjligt att via spårbarhet ta reda på vem som tidigare arbetat med systemet och tillhörande dokumentation. Det betyder att trots att innehållet är ofullständigt, kan den nya utvecklaren be om hjälp av utvecklaren som besitter mer kunskap. Precis som med uppdatering, uppstår problematik kring spårbarhet första när utvecklaren lämnar företaget och inte längre går att komma i kontakt med. Således kan problematisk korrekthet och fullständighet leda till dokumentationsskuld i de fall när det hindrar användaren från ett fortsatt utvecklingsarbete (Mendes et al., 2016). Detta kan ske när utvecklaren inte har tillräckligt med kunskap och när spårbarheten inte bidrar till att ursprungsutvecklaren kan kontaktas.

Respondenterna trycker på att erfarenhet och kunskap hos utvecklaren avgör hur väl hen kan ta till sig dokumentationens innehåll, oberoende dokumentationskvalitet. Tre av respondenter menar att om utvecklaren kan systemet sedan innan, har de kunskapen att kunna se igenom en sämre kvalitet och kan därför uppdatera källkoden ändå. Det här styrks av Sulaiman, Idris & Sahibuddin (2002) som säger att intern systemdokumentation kan te sig användbart oberoende av felaktigheter i olika kvalitetsattribut. Dock anser hälften av respondenterna att nya eller mindre erfarna utvecklare kommer behöva en längre inlärningsperiod eftersom de har svårare att se igenom sämre kvalitet. Det går i linje med Selic (2009) som anser att juniora utvecklare

behöver välskrivna intern systemdokumentation för att arbetet ska hålla hög nivå. Bristfällig dokumentationskvalitet kan således leda till dokumentationsskuld beroende på individen som använder den.

### 5.3.6 *Konsekvenser av dokumentationsskuld*

Ur försämrade dokumentationskvalitet, kan dokumentationsskuld leda till konsekvenser som exempelvis att fel slutsatser dras, vilket kan resultera i en stor mängd spill samt att utvecklarna kommer att implementera något som inte fungerar. När utvecklare drar fel slutsatser, menar respondent 5 att det bidrar till höga kostnader eftersom det lätt blir fel i källkoden. När en ändring blir alldeles för dyr, väljer företaget att prioritera resurser på annat arbete och således blir den felaktiga ändringen kvar i systemet och kommer sällan åtgärdas. Det går i riktning med De Graaf (2011) som uttrycker att dokumentationskvalitet som inte uppfyller kraven, kan resultera i lägre effektivitet, defekter samt lägre systemkvalitet.

## 5.4 Tillgänglighet

Genom litteratur och intervjuer har det framkommit att det finns ett antal problemområden i anslutning till tillgänglighet samt hur dessa kan bidra till dokumentationsskuld.

### 5.4.1 *Åtkomstrestriktioner*

Det första, och utan tvekan största, problemområdet behandlar var och hur systemdokumentationen finns sparad. Zhi et al. (2015) diskuterar att det bästa sättet att spara systemdokumentation är i ett dokumenthanteringssystem. Författarna får medhåll av respondenterna, där majoriteten är nöjda med det dokumenthanteringssystem deras företag använder sig av. Dock anser dem att tillgängligheten påverkas av en persons arbetsroll. Om en anställd utan rätt arbetsroll vill ha tillgång till ett specifikt dokument kommer hen behöva skicka en förfrågan till ägaren av dokumentationen. Det här betyder att så länge utvecklare inte har tillgång till dokumentationen, kommer dokumentationsskuld skapas eftersom otillgängligheten hindrar att utvecklingsarbetet kan fortgå. Däremot, så fort utvecklaren får åtkomst till dokumentationen, kommer det inte längre resultera i dokumentationsskuld.

Två av respondenterna instämmer i att systemdokumentation som ligger i hanteringssystemet är enkelt att nå, dock anser dem att systemdokumentation ibland sparas av medarbetare i format eller på platser som resterande inte kommer åt. Tang, Liang & van Vliet (2011) diskuterar att detta kan leda till att personlig kunskap inte delas vidare och gör den allmänna kunskapen kring ett system osammanhängande och spretig. Respondent 2 menar att det kommer resultera i att erfarna utvecklare fortfarande kommer kunna arbeta med systemet, medan nya, oerfarna utvecklare kommer få problem.

Respondent 6 förklarar även att svårtillgängliga systemdokument är en av anledningarna till att hen inte utför ett aktivt uppdateringsarbete. Detta påstående går i linje med Hunt & Thomas (2000) som också menar att begränsad tillgänglighet kommer göra uppdateringsarbete lidande då utvecklare inte orkar lägga tid eller energi på att leta efter systemdokumentationen. Tidigare forskning (Guo & Seaman, 2011; Mendes et al., 2016; Rios, de Mendonça & Spínola, 2018) visar på att utdaterad systemdokumentation kommer bidra till dokumentationsskuld.

Respondent 6 visar således på att tillgänglighet inte endast är problematisk i sig själv, utan kan även påverka andra faktorer inom intern systemdokumentation.

#### 5.4.2 Sökning, namngivning och indexering i dokumenthanteringssystemet

Ytterligare ett problemområde berör hur enkelt eller svårt det är få fram rätt intern systemdokumentation ur hanteringssystemet. Zhi et al. (2015) förklarar att i ett dokumenthanteringssystem ska det vara möjligt att söka och kategorisera dokumenten, vilket de flesta av respondenterna anser är möjligt i deras dokumenthanteringssystem. Matulevicius, Kamseu, & Habra (2009) menar också på att tillgänglighet innefattar att information presenteras på ett sätt som gör den enkel att ta till sig och förstå. Respondent 1 och 2 tycker inte information presenteras på ett lämpligt sätt och önskar att det hade varit lättare att göra sökningar, kategoriseringar och indexering på systemdokumentationen. Respondent 2 anser att hans företag är ganska dåliga på bra dokumentnamn, vilket har resulterat i att det finns många systemdokument som har blivit överflödiga och inte används. När informationen inte presenteras på ett sätt som gör dokumenten användbara, förhindrar det att utvecklingsarbetet kan fortgå och leder därför till dokumentationsskuld (Mendes et al., 2016). Eftersom majoriteten av respondenterna däremot uttrycker att sökning och framtagning av information fungerar bra, kan slutsatsen dras att sökning, namngivning och indexering i dokumenthanteringssystem framförallt inte leder till dokumentationsskuld.

#### 5.4.3 Versionshantering

På versionshantering svarade alla respondenter att både äldre och nuvarande systemdokumentation sparas. De flesta anser även att det är enkelt att förstå vilket dokument som är mest relevant. Respondent 2 och 6 tycker däremot att det inte är tillräckligt tydligt vilken version som är relevant inom deras företag. Eftersom det finns en risk för att olika projekt eller systemutvecklare arbetar med samma system och dess tillhörande dokumentation samtidigt är det viktigt att korrekt och aktuell dokumentation är tillgänglig i systemet (Sommerville, 2001), vilket styrks av respondent 6 som förklarar ett verkligt scenario där medarbetare på hans företag inte hade tillgång till samma version av systemdokumentationen. Detta resulterade i att underkonsulter gjort ändringar av ett system baserat på en kopia av ett äldre dokument, vilket går i linje med Sommerville (2011) som förklarar att bristfällig versionshantering leder till att utvecklare gör enkla misstag.

Tre av respondenterna anser att versionshantering bidrar till bättre kunskapsspridning inom företag då det är möjligt att spåra vem som har skapat och ändrat systemdokumentationen. På så sätt är det möjligt för en systemutvecklare att diskutera innehållet eller ställa frågor, vilket kan bidra till ett ökat lärande. Därav behöver försämrade versionshantering inte leda till dokumentationsskuld, eftersom systemutvecklingen kan fortgå. Problematiken med samtal, som tidigare tagits upp av Tang, Liang & van Vliet (2011) är att konversationer kan te sig ineffektiva och innefatta felaktigheter. Författarna Tang, Liang & van Vliet (2011) menar även på att om konversationer ska te sig korrekta är det viktigt att informationen kommer ifrån ursprungskällan och inte från återberättelser. Så länge versionshanteringsverktyget visar på vem som senast ändrat eller skapat dokumentationen bör informationen komma från ursprungskällan och således inte bidra till dokumentationsskuld.

#### 5.4.4 *Konsekvenser av tillgänglighet*

Respondent 1 och 4 förklarar att all begränsad tillgänglighet, oavsett anledning, kommer leda till att en systemutvecklare inte kommer kunna göra underhålls- eller utvecklingsarbete av systemet, vilket är självaste definitionen av dokumentationsskuld (Mendes et al., 2016). Begränsad tillgänglighet kommer därav leda till att systemet inte utformas som överenskommet, vilket även respondent 4 håller med om, och kommer således kosta företaget desto mer pengar (de Graaf et al., 2014). Respondent 2 och 6 håller däremot inte med. De yrkar på att konsekvensen av begränsad tillgänglighet helt beror på storlek på projektteamet. I ett litet projektteam kommer utveckling och underhåll fortlöpa precis som tidigare eftersom det är enklare att ta del av varandras kunskap. Ju större ett projekt blir och ju fler erfarna utvecklare som försvinner desto fler problem kan uppstå. Respondent 2 och 6 anser däremot att det inte finns ett givet samband mellan begränsad tillgänglighet och sämre systemkvalitet. Respondent 6 säger istället, och får medhåll från respondent 1 och 5, att svårtillgängliga dokument kommer leda till försvårad och försenad utveckling av systemet, eftersom involverade personer inte kommer besitta rätt kunskap. Detta påstående håller författarna (de Graaf et al., 2014) med om genom att förklara att kunskap som kan hittas och hämtas ner snabbt och effektivt kommer spara in dyrbar tid.

Det är svårt att dra några definitiva slutsatser gällande tillgänglighet. Trots att det finns en gemensam konsensus över vilka specifika faktorer som kan bidra till dokumentationsskuld, finns det ingen gemensam tanke till vilken grad problemområdena kommer bidra. Tillgänglighet ter sig således som en väldigt individuell faktor, där flertal problemområden kan leda till konsekvenser som dokumentationsskuld, men inte att det nödvändigtvis behöver göra det. Mycket verkar således bero på vilka bestämmelser företagen har på management-nivå. Hanteringssystem, versionshantering och rollrestriktioner bestäms ofta av företaget längre upp i hierarkin och trots att respondenterna kan ha egna åsikter kring hur bra eller dåliga dessa verktyg är, har de inget annat val än att använda dessa.

#### 5.4.5 *Konsekvenser av dokumentationsskuld*

En definitiv konsekvens av dokumentationsskuld, som följd av begränsad tillgänglighet, är att utvecklingen och underhållsarbete av ett system kommer att ta mycket längre tid. Respondent 4 menar att denna försening beror på att systemutvecklarna tvingas lägga mer tid på att förstå systemet, även respondent 5, 6 och håller med. Tom, Aurum & Vidgen (2013) diskuterar att utdaterad systemdokumentation kommer resultera i att utvecklare måste lägga tid på att bekanta sig med utvecklingsspråket, verktygen, funktionerna eller det specifika problemet. Trots att författaren främst diskuterar detta i relation till utdaterade dokument, så visar all insamlade data från intervjuer på ett liknande scenario för begränsad tillgänglighet.

### 5.5 Slutgiltig reflektion

I svaren från respondenterna, samt viss litteratur, finns det tre genomsyrande teman, nämligen spårbarhet, erfarenhet och managementbeslutsfattning.

En detalj som blivit uppmärksammas i all insamlade data är att intern systemdokumentation ter sig väldigt individuell. Vad som anses vara viktigt för en utvecklare behöver inte nödvändigtvis vara viktigt för en annan. Det har framkommit att utvecklare arbetar med intern

systemdokumentation när det anses viktigt för den egna personen och när tid finns. Två av respondenterna uttrycker att ett management-beslut krävs för att arbetet kring intern systemdokumentation ska bli bättre. Även resterande respondenters svar talar från ett perspektiv där individen själv tar sig an dokumentationsarbetet på olika sätt. Således finns det varierande kvalitet på arbetet med intern systemdokumentation inom företag. För att effektivisera arbete med intern systemdokumentation borde beslut, standarder, kvalitetssystem etcetera ligga på management-nivå. Arbete, uppdatering och utveckling bör inte vara individbaserat, snarare borde det finnas en utformad standard eller tradition för hur, vad, varför och när intern systemdokumentation bör produceras.

Hur allvarliga konsekvenserna av uppdatering, dokumentationskvalitet och tillgänglighet blir beror på den kompetens och erfarenhet ett företags systemutvecklare besitter. Respondenterna talade ofta om att utvecklare som arbetat länge och därmed byggt upp erfarenheter, kunskap och nätverk, kan hantera sämre intern systemdokumentation bättre än vad nyexaminerade utvecklare kan. De menar däremot inte att erfarna utvecklare inte drar nytta av intern systemdokumentation. Men om de inte förstår innehållet tillräckligt eller kan använda dokumentationen, kan de snarare lösa det på andra sätt genom att exempelvis lägga utdragna timmar på att förstå koden och prata med andra utvecklare. De största problemen framkommer när utvecklare, gamla som nya, ska ta sig an nya system. Det är där respondenterna lägger tyngd vid att prata med andra utvecklare. Kunskapsdelningen inom alla företagen är framförallt genom konversationer med utvecklaren som tidigare arbetat med systemet, vilket gör att spårbarheten av både system- och dokumentationsskaparen är mycket viktigt. Det är en faktor som alla utvecklare behöver kunna använda sig av.

## 6 Slutsats

Genom att titta på tidigare forskning, samt komplettera med empirisk undersökning där systemutvecklare deltag i intervjuer, har forskningsfrågorna besvarats.

*Hur kan faktorerna uppdatering, dokumentationskvalitet och tillgänglighet inom intern systemdokumentation leda till ogynnsamma påföljder som dokumentationsskuld?*

Icke-uppdaterad intern systemdokumentation har direkta följder till dokumentationsskuld. Däremot, kan inga slutsatser dras kring om storleken på kodändringen leder till att dokumenten uppdateras eller inte. Däremot uppdaterar respondenterna olika beroende på kodändring, och någon typ av kodändringar kommer för alla respondenter att prioriteras bort. I de fall där systemdokumenten går att använda, trots att dokumenten är föråldrade, leder det inte till dokumentationsskuld. I alla andra fall är däremot dokumentationsskuld en direkt följd. Då respondenterna, med stöd från tidigare forskning, har svårt att se den faktiska nyttan, väljs arbetet ofta bort. Det här i kombination med att tid och pengar inte prioriteras, bidrar till att föråldrad intern systemdokumentation förhindrar att systemutvecklingsarbete kan genomföras, det vill säga, dokumentationsskuld skapas. Mest problematiskt är det när utvecklare, framförallt nyexaminerade, inte kan utföra sitt arbete som följd från att dokumenten inte är uppdaterade med all kunskap som krävs. Den intern systemdokumentation uppnår inte syftet och hindrar istället utvecklare från att kunna underhålla systemet. Enligt definition, betyder det att dokumentationsskuld uppstår.

*Dokumentationskvalitet* anses vara en av de viktigaste faktorerna i tidigare studier, men desto mindre viktigt bland respondenterna. Hur väl en systemutvecklare kan ta till sig och använda intern systemdokumentation med sämre kvalitet beror på individens kompetens och erfarenhet. Kvalitetsattributen läsbarhet och struktur verkar däremot sällan eller aldrig leda till dokumentationsskuld, medan korrekthet och fullständighet oftare kan leda till dokumentationsskuld. Det sker framförallt när utvecklaren inte har tillräckligt med kunskap eller när spårbarheten är otillräcklig.

*Tillgänglighet* av intern systemdokumentation ter sig som en väldigt individuell faktor där flera av områdena kan leda till dokumentationsskuld, men inte att de nödvändigtvis behöver göra det. Liksom med tidigare faktorer, beror konsekvenserna på systemutvecklarens erfarenheter, samt olika bestämmelser på management-nivå. Mycket visar dock på att om en systemutvecklare inte kommer åt rätt dokumentation kommer detta leda till dokumentationsskuld eftersom det hindrar den involverade från att arbeta med systemet.

*Vilka är de främsta konsekvenserna som dokumentationsskuld leder till?*

Från faktorerna uppdatering, dokumentationskvalitet och tillgänglighet, kan dokumentationsskuld leda till flera konsekvenser som påverkar systemutvecklingen. Konsekvenserna grundas framförallt i att föråldrade dokument, försämrade dokumentationskvalitet och bristande tillgänglighet leder till att kunskap om systemet antingen går förlorad, blir tvetydig eller otillräcklig för att utvecklingsarbetet ska kunna fortgå obehindrat. De främsta konsekvenserna blir att arbetet tar längre tid, det lätt blir fel samt blir en mycket kostsam aktivitet.

## 6.1 Förslag till vidare forskning

Genom vår kvalitativa studie hittades ett antal faktorer som anses viktiga för att arbete med intern systemdokumentation ska vara effektivt, samt att dess användningsförmåga ska vara så hög som möjligt. Problem inom de olika faktorerna är väldigt individbaserade och kvalitet på systemdokumentationen varierar inom ett och samma företag. Många av respondenterna förespråkade därför ett gemensamt arbetssätt när det kommer till dokumentationsarbete. Således anser vi det hade varit intressant att undersöka företag som har gemensamt, management-bestämt underlag för arbete med dokumentation. Även skillnader mellan ett sådant företag, och ett företag som inte har gemensamt direktiv för utformning och arbete med intern systemdokumentation.

## 7 Referenser

- Aghajani, E., Nagy, C., Vega-Márquez, O. L., Linares-Vásquez, M., Moreno, L., Bavota, G., & Lanza, M. (2019). Software Documentation Issues Unveiled, *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pp.1199-1210, Tillgänglig online: <https://www.inf.usi.ch/phd/aghajani/resources/papers/agma2019b.pdf> [Hämtad 27 mars 2020]
- Alves, N. S., Ribeiro, L. F., Caires, V., Mendes, T. S., & Spínola, R. O. (2014). Towards an ontology of terms on technical debt, *2014 Sixth International Workshop on Managing Technical Debt*, pp. 1-7, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 15 april 2020]
- Briand, L. C. (2003). Software documentation: how much is enough?, *Seventh European Conference on Software Maintenance and Reengineering, 2003. Proceedings*, pp. 13-15), Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 27 mars 2020]
- Cabrera, A., & Cabrera, E. F. (2002). Knowledge-sharing dilemmas, *Organization studies*, 23(5), pp. 687-710, Tillgänglig online: <https://journals.sagepub.com/doi/pdf/10.1177/0170840602235001> [Hämtad 9 april 2020]
- Campos, M., Valente, M. T., & Terra, R. (2016). Does Technical Debt Lead to the Rejection of Pull Requests?, arXiv preprint arXiv:1604.01450, Tillgänglig online: <https://arxiv.org/pdf/1604.01450.pdf> [Hämtad 15 april 2020]
- Correia, F. F. (2008). Extending and integrating wikis to improve software documentation, *Proceedings of Wikis for Software Engineering*, Tillgänglig online: [http://blog.invisivel.net/wp-content/papercite-data/pdf/correia\\_extending\\_2008.pdf](http://blog.invisivel.net/wp-content/papercite-data/pdf/correia_extending_2008.pdf) [Hämtad 9 april 2020]
- Dalkir, K. (2005). Knowledge management in theory and practice, Jordan Hill: Taylor & Francis Group, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 27 mars 2020]
- De Boer, R. C. & van Vliet, H. (2009). Writing and Reading Software Documentation: How the development process may affect understanding, *ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, Vancouver, BC, 2009, pp. 40-47, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 27 mars 2020]
- De Graaf, K. A. (2011). Annotating software documentation in semantic wikis, *Proceedings of the fourth workshop on Exploiting semantic annotations in information retrieval*, pp. 5-6, Tillgängligt online: <https://dl.acm.org/doi/pdf/10.1145/2064713.2064718> [Hämtad 9 april 2020]
- De Graaf, K. A., Liang, P., Tang, A., & Van Vliet, H. (2014). The impact of prior knowledge on searching in software documentation, *Proceedings of the 2014 ACM symposium on Document engineering*, pp. 189-198, Tillgänglig online: <https://dl.acm.org/doi/pdf/10.1145/2644866.2644878> [Hämtad 2 april 2020]



- De Souza, S. C. B., Anquetil, N. & de Oliveira, K.M. (2005). A Study of the Documentation Essential to Software Maintenance, Proc. 23rd Annu. Int. Conf. on Design of Communication: Documenting & Designing for Pervasive Inform., New York, NY, USA, pp. 68–75, Tillgänglig online: <https://dl.acm.org/doi/pdf/10.1145/1085313.1085331> [Hämtad 2 april 2020]
- Ding, W., Liang, P., Tang, A., & Van Vliet, H. (2014). Knowledge-based approaches in software documentation: A systematic literature review. *Information and Software Technology*, 56(6), pp. 545-567, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 2 april 2020]
- Falessi, D., Briand, L. C., Cantone, G., Capilla, R., & Kruchten, P. (2013). The value of design rationale information, *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 22(3), pp. 1-32, Tillgänglig online: <https://dl.acm.org/doi/pdf/10.1145/2491509.2491515> [Hämtad 5 april 2020]
- Forward, A. (2002). Software documentation: Building and maintaining artefacts of communication, University of Ottawa, Tillgänglig online: [https://www.site.uottawa.ca/~tcl/gradtheses/forward/forward\\_thesis.pdf](https://www.site.uottawa.ca/~tcl/gradtheses/forward/forward_thesis.pdf) [Hämtad 27 mars 2020]
- Forward, A., & Lethbridge, T. C. (2002). The relevance of software documentation, tools and technologies: a survey, *Proceedings of the 2002 ACM symposium on Document engineering*, pp. 26-33, Tillgänglig online: <https://dl.acm.org/doi/pdf/10.1145/585058.585065> [Hämtad 27 mars 2020]
- Garousi, G., Garousi-Yusifoglu, V., Ruhe, G., Zhi, J., Moussavi, M., & Smith, B. (2014). Usage and usefulness of technical software documentation: An industrial case study, *Information and Software Technology*, 57, pp. 664-682, Tillgänglig online: <https://www.sciencedirect.com/science/article/abs/pii/S095058491400192X?via%3Dihub> [Hämtad 5 april 2020]
- Guo, Y., & Seaman, C. (2011). A portfolio approach to technical debt management, *Proceedings of the 2nd Workshop on Managing Technical Debt*, pp. 31-34, Tillgänglig online: <https://dl.acm.org/doi/pdf/10.1145/1985362.1985370> [Hämtad 9 april 2020]
- Guo, Y., Spínola, R. O., & Seaman, C. (2014). Exploring the cost of technical debt management – a case study, *Empirical Software Engineering*, 21(1), p. 159, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 9 april 2020]
- Henriks, P. (1999). Why Share Knowledge? The Influence of ICT on the Motivation for Knowledge Sharing. *Knowledge and Process Management*. Vol 6, no. 2, pp. 91-100, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 9 april 2020]
- Hunt, A. & Thomas, D. (2000). *The Pragmatic Programmer: From Journeyman to Master*, Boston: Addison-Wesley Professional, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 2 april 2020]

- Jacobsen, D. I. (2002). Vad, hur och varför: om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen, Studentlitteratur.
- Lethbridge, T. C., Singer, J., & Forward, A. (2003). How software engineers use documentation: The state of the practice. *IEEE software*, 20(6), pp. 35-39, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 2 april 2020]
- Matulevicius, R., Kamseu, F., & Habra, N. (2009). Measuring Open Source Documentation Availability, Proceedings of the international Conference on Quality Engineering in Software Technology, pp. 83-102, Tillgänglig online: [https://www.researchgate.net/profile/Raimundas\\_Matulevicius/publication/267680608\\_Measuring\\_Open\\_Source\\_Documentation\\_Availability/links/55acd33e08aea3d0868614c1.pdf](https://www.researchgate.net/profile/Raimundas_Matulevicius/publication/267680608_Measuring_Open_Source_Documentation_Availability/links/55acd33e08aea3d0868614c1.pdf) [Hämtat 27 mars 2020]
- Mendes, T., Farias, M., Mendonca, M., Soares, H., Kalinowski, M., & Spínola, H. (2016). Impacts of agile requirements documentation debt on software projects; a retrospective study, Tillgänglig online: <https://dl.acm.org/doi/pdf/10.1145/2851613.2851761> [Hämtad 9 april 2020]
- Oates, B. J. (2005). Researching information systems and computing. Sage.
- Orb, A., Eisenhauer, L., & Wynaden, D. (2001). Ethics in qualitative research, *Journal of nursing scholarship*, 33(1), pp. 93-96, Tillgängligt online: <https://sigmapubs.onlinelibrary.wiley.com/doi/epdf/10.1111/j.1547-5069.2001.00093.x> [Hämtad 15 april 2020]
- Parnas, D. L. (2011). Precise documentation: The key to better software, *the Future of Software Engineering*, pp. 125-148, Springer, Berlin, Heidelberg. Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 3 april 2020]
- Plösch, R., Dautovic, A., & Saft, M. (2014). The value of software documentation quality, *2014 14th International Conference on Quality Software*, pp. 333-342, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 5 april 2020]
- Rios, N., de Mendonça Neto, M. G., & Spínola, R. O. (2018). A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners, *Information and Software Technology*, 102, pp. 117-145, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 9 april 2020]
- Rus, I., Lindvall, M., & Sinha, S. (2002). Knowledge management in software engineering, *IEEE software*, 19(3), pp. 26-38, Tillgänglig online: <https://www.csiac.org/wp-content/uploads/2016/02/KM-in-SWE-SOAR.pdf> [Hämtad 1 april 2020]
- Ryan, S., & O'Connor, R. V. (2013). Acquiring and sharing tacit knowledge in software development teams: An empirical study, *Information and Software Technology*, 55(9), pp. 1614-1624, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 3 april 2020]

- Saunders, B., Kitzinger, J., & Kitzinger, C. (2015). Anonymising interview data: Challenges and compromise in practice, *Qualitative Research*, 15(5), pp. 616-632, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 9 april 2020]
- Selic, B. (2009). Agile documentation, anyone?. *IEEE software*, 26(6), pp. 11–12, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 28 mars 2020]
- Soares, H. F., Alves, N. S., Mendes, T. S., Mendonça, M., & Spínola, R. O. (2015, April). Investigating the link between user stories and documentation debt on software projects, 2015 12th International Conference on Information Technology-New Generations, pp. 385-390, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 12 april 2020]
- Sommerville, I., (2011). *Software Engineering (9th Edition)*, 9th Edition. Addison-Wesley
- Sommerville, I. (2001). Software documentation, *Software engineering*, pp. 143-154
- Stapel, K., & Schneider, K. (2014). Managing knowledge on communication and information flow in global software projects, *Expert Systems*, 31(3), pp. 234-252, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 8 april 2020]
- Sulaiman, S., Idris, N. B., & Sahibuddin, S. (2002), Production and maintenance of system documentation: what, why, when and how tools should support the practice, *Ninth Asia-Pacific Software Engineering Conference*, pp. 558-567, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 8 april 2020]
- Tang, A., Liang, P., & Van Vliet, H. (2011, June). Software architecture documentation: The road ahead, *Ninth Working IEEE/IFIP Conference on Software Architecture*, pp. 252-255, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 5 maj 2020]
- Tom, E., Aurum, A., & Vidgen, R. (2013). An exploration of technical debt, *Journal of Systems and Software*, 86(6), pp. 1498-1516, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 10 april 2020]
- Zhi, J., Garousi-Yusifoglu, V., Sun, B., Garousi, G., Shahnewaz, S., & Ruhe, G. (2015). Cost, benefits and quality of software development documentation: A systematic mapping. *Journal of Systems and Software*, pp. 175-198, Tillgänglig via: LUBsearch <http://lubsearch.lub.lu.se/> [Hämtad 3 april 2020]

## 8 Bilagor

### Bilaga A – Intervjufrågor

Innan start förklarades:

- a) Uppsatsens syfte och mål
- b) Det här är frivilligt att dra sig ur när som än önskas
- c) Respondenterna kommer vara anonyma till namn och företag
- d) Inspelning av intervju

#### Introduktionsfrågor

- Vad gör ditt företag?
- Vad är din arbetsroll/yrkestitel?
- Dina arbetsuppgifter?

#### Intern systemdokumentation

- Använder du dig aktivt av intern systemdokumentation i ditt vardagliga arbete?
  - Anser du att intern systemdokumentation förenklar ditt arbete? Varför/hur?
- Vad ska intern systemdokumentation uppnå för kriterier för att vara användbar för dig?
- Hur väl förlitar du dig på den systemdokumentation som redan finns?
- Hur fungerar kunskapsdelning inom ert företag?
  - Delar du helst kunskap via dokumentation eller muntliga konversationer?
- Hur gör ni när någon kunskapsbesittande person närmar sig att lämna företaget?
- Arbetar du med underhållsarbete av ett system?
  - Om ja:
    - Hur brukar du hitta den kunskap som krävs för att utföra arbetet?
    - Hur användbar är dokumentationen kring systemet?

#### Uppdatering

- Hur viktigt är det för dig att uppdatera systemdokumentation?
  - Varför känner du behovet av att uppdatera/inte uppdatera din dokumentation?
  - Av vilken anledning uppdaterar du (inte) den interna systemdokumentationen?
- Hur ofta uppdaterar du i genomsnitt din interna systemdokumentation?
- Dokumenterar du annorlunda beroende på storlek av ändringar i koden?
  - Om ja:
    - Ser arbetet annorlunda ut beroende på stora eller små kodändringar?
- Använder du systemdokumentation även om den är gammal och inte är uppdaterad till aktuell status?
  - Om ja:
    - På vilket sätt är den användbar?
    - Tror du det hade varit ännu bättre om den hade varit uppdaterad?
    - Anser du att skillnaden hade varit så pass stor att det ansetts värt att lägga ner tid på att uppdatera systemdokumentationen?
  - Om nej:
    - Av vilken anledning?
- Hur påverkar icke-uppdaterad dokumentation delandet av kunskap?

- Hur tycker du att icke-uppdaterad intern systemdokumentation påverkar systemutvecklingen/underhållsarbete/systemet?

### Kvalitet

- Vad anser du motsvarar hög dokumentationskvalitet?
- Vad är dina förväntningar på kvaliteten?
  - Tycker du dina förväntningar uppfylls i den systemdokumentation du använder?
  - Om nej:
  - Hur ser det ut istället?
- Hur viktiga är följande kvalitetsattribut för kvaliteten hos den systemdokumentation du använder dig av?
  - Korrekthet (översättningar, kommentarer, kodguider)
  - Läsbarhet (stavning, grammatik, namnkonventioner, språkuttryck)
  - Struktur (standarder för hur dokumentationen ska vara utformad)
  - Fullständighet (färdigställda dokument, nödvändigt innehåll)
- Tar du hänsyn till dessa kvalitetsattribut när du själv skriver systemdokumentation?
- Hur ofta innehåller systemdokumentation all nödvändig information som krävs för att förstå, underhålla och använda systemet?
  - Är det lönsamt att använda dokumentation som inte innehåller all denna information? Vid vilka fall?
  - Anser du det vara viktigt att dokumentationen alltid är fullständig?
- Hur bidrar kvalitetsnivån till delningen av kunskap?
- Hur tycker du att intern systemdokumentation med dålig kvalitet påverkar systemutvecklingen/underhållsarbete/systemet?

### Tillgänglighet

- Är det lätt att få tillgång till intern systemdokumentation när du behöver den?
- Var och hur är systemdokumentationen förvarad?
  - Använder ni er av ett dokumenthanteringssystem?
- Hur önskar du att den var förvarad, för att bringa mest nytta?
- Sparas endast den senaste versionen av dokumentationen eller sparas även äldre versioner?
  - Är det lätt att veta vilken som är relevant/senast uppdaterad?
- Hur bidrar tillgängligheten av systemdokumentation till delningen av kunskap?
- Hur tycker du att begränsad tillgänglighet inom intern systemdokumentation påverkar systemutveckling/underhållsarbete/systemet?

### Avslutande frågor

- Finns det någonting du hade velat lägga till som du kommit på under intervjuens gång?
- Är det okej om vi kontaktar dig om vi skulle ha några andra frågor kring ämnet eller förtydligande av det som redan sagts?
- Skulle du vilja att vi skickar transkriberingen till dig efteråt så att du kan kolla igenom den så att allt ser bra ut?

## Bilaga B – Respondent 1, transkribering

R1: Respondent 1

LH: Linnéa Hansson

MH: Mathilda Hezsö

Rad	Person	Meningsenhet	Kod
1	R1	Ska jag berätta lite om min erfarenhet av systemdokumentation?	
2	LH	Ja, eller vi kan faktiskt först köra såhär, vi vill bara fråga om du vill vara anonym? Vi kommer inte skriva ditt namn i vilket fall, men om vi kan nämna din organisation?	
3	R1	Gör du det, det är helt okej.	
4	LH	Vad bra.	
5	R1	Jag ska börja med att säga det att systemdokumentation har vi diskuterat ända sedan mitten på 80-talet. Vad ska va med, vad ska inte vara med? Det finns ju många olika infallsvinklar på vad som är systemdokumentation egentligen. Att det man programmerar, alltså kodverk, metoder och sådana saker. Det brukar man oftast dokumentera i själva koden. Vad gör den här metoden? Vad kan du förvänta dig, osv. Men vad man skall dokumentera är vad man skall göra, vad det är man ska programmera. Det är mycket eftersatt.	SD
6	MH	Ja okej.	
7	LH	Okej.	
8	R1	Det beror ju mycket på att det inte har funnits några direkta krav. Jag ska säga det att, när det gäller *företagets namn*, så hoppas jag att det har blivit bättre. Det kan ni ju undersöka med någon som är lite yngre än mig. Det enda system som jag hållit på med som har varit fullständigt dokumenterat i alla sina aspekter, det är Nationell patientöversikt som jag höll på med i början på 2000-talet. Det var ju då drivet av det som heter Inera och där var det ju tvunget eftersom vi jobbade nationellt och det skulle va en mängd olika intressenter med det arbetet, så då var vi tvungna att dokumentera allting. Men det var inte heller nere på kodnivå eftersom det var vår kod. Så det var vår interna affärshemlighet kan man säga, hur man kodar. Men vad den gör, och vad man kan förvänta sig. Det här med modeller, är ju en dokumentation. Det finns ju olika typer av modeller. Informationsmodeller, datamodeller osv. Informationsmodellen borde vara offentligt tycker jag. Vilken information hanteras i det här systemet och hur ser flödet ut. Datamodellen, det vill säga om man använder relationsdatabas. Vilka tabeller som finns och hur de relaterar till varandra i förhållande till kod, det är också affärshemlighet, oftast.	SD
9	LH	Men om vi ska göra såhär, är det så att du använder dig aktivt av en sån här intern systemdokumentation i ditt vanliga arbete? Du sa ju att du endast gjort det fullt ut i ett system,	ISD

		men är det ändå något du alltid gjort igenom hela din systemutvecklingshistoria?	
10	R1	Det kan man säga. Någon form av dokumentation måste ju finnas. Den aktiva delen det är ju den delen som är kravställarens, det vill säga kundens, förväntning. Vad är det egentligen kunden har beställt.	ISD
11	LH	Ja men precis. Anser du att det förenklar arbetet att ha dokumentation, för dig som systemutvecklare?	ISD
12	R1	Ja visst, det är ju en förutsättning för att överhuvudtaget kunna göra någonting. Det finns ju olika befattningshavare i en organisation där man utvecklar IT. Den som sitter och kodar gör ju inte det här alltid på eget bevåg, utan man har ju då också en systemarkitekt som bestämmer vilka metoder, och flödena i informationsmodellen. Och den är ju jätteviktigt, och den används av alla och det vet jag att det finns. Den är ofta sådan att den är dynamisk i den meningen att man skriver den först och sen får man revidera den ett antal gånger. Det beror ju på att man gör ett test först, och så visar man upp en typ av mock up-version. Naturligtvis kan man använda sig av systemdokumentation.	ISD
13	LH	Om du sitter och ska vidareutveckla eller underhålla ett system, och du har en dokumentation av det som redan finns, hur mycket litar du på den dokumentation som är skriven?	ISD
14	R1	Det beror på hur gammal den är och vem som har skrivit den.	ISD+U
15	LH	Ja okej, så det handlar om person och hur gammal den är helt enkelt. Jag förstår.	
16	R1	För det kan ju vara så att om det finns en dokumentation, dokument eller ett antal dokument om det systemet som jag ska börja med, om det är reviderat ett antal gånger utan att det dokumenterats, då kan jag inte lite på systemdokumentationen alls. Utan då får jag testa mig fram. Förhoppningsvis då så är det så att jag har reviderat den själv, och det är arkitektens roll i det hela. Det är egentligen inte själva programmeraren som ska sitta och revidera ett sådant dokument. Programmeraren får ett uppdrag att lösa, sen får han eller hon rapportera det till arkitekten. Sen får arkitekten rapportera dokumentet, eller presentera dokumentet för kunden.	ISD+U
17	LH	Men om man ska ta det här lite mer in på kunskap, hur skulle du säga att kunskapsdelningen fungerar?	KD
18	R1	Ganska dåligt faktiskt. Vi jobbar ju då med objektorienterad programmering och då är det så att man i teorin skulle kunna skriva en metod som sen återanvänds igenom organisationens alla system i alla tidsaspekter. Det där är det si och så med.	KD
19	LH	Men du personligen, skulle du helst dela din kunskap via dokumentation eller via konversationer och diskussioner och liknande, med dina medarbetare?	KD
20	R1	Både och naturligtvis. Det är ju också så att i ett sånt företag som vårt så är inte alla programmerare på plats i lokalen utan de är i Indien och Danmark och överallt. Det blir ju mycket engelska även om man tycker att engelska är lätt och nytt och	KD, ISD

		att vi förstå, så gör vi inte det. Därför finns ett dokumentationsansvar mycket högre på sådana system än om det bara är jag och *namn på kollega* som sitter och jobbar på någonting. Vi har jobbat ihop på det här sedan -92. Då är det en helt annan sak. Men ändå så ska det finnas systemdokumentation.	
21	LH	Men hur skulle ni göra när medelåldern börjar bli hög och många som närmar sig pensionsåldern eller bara ska byta jobb, hur gör man då för att inte gå miste om kunskapen?	KD
22	R1	Man gör inte det, man tappar den. Det är väldigt vanligt. Vi har ju, mig själv som exempel, som har gått i pension. Och det är ju många som har kontaktat mig och sagt, men *R1* hur gjorde du det här egentligen? Hur funkar det egentligen? Om kunden vill ha en ändring, då är det ju så att jag jobbar som konsult och känner att, det kan jag ju göra själv. Och då gör jag ju det, och då har jag på Teamer ett antal dokument, och är det så att jag ser att det måste ändras, så gör jag det där.	KD+ISD
23	LH	Okej, men arbetar du med underhåll av tidigare system, eller har du gjort det mycket?	
24	R1	Det är ju det jag gör nu.	
25	LH	Ja okej. Hur brukar du gå tillväga för att hitta den kunskapen som krävs för att kunna utföra underhållsarbetet?	KD
26	R1	Eftersom det är jag som har gjort systemet så går jag till mig själv haha.	KD
27	LH	Ja okej, så det är din personliga kunskap?	KD
28	R1	Ja precis.	KD
29	LH	Ja okej.	
30	R1	Ja och det är faktiskt så att det är en brist hos företag, och inte bara hos oss. Då är det såhär att det ställs inte krav på att man ska lämna efter sig systemdokumentation på det sättet. Det tycker jag skulle vara en viktig uppgift för er att få fram till organisationer. Och också en sak som ni tar upp här ser jag är ju standardiseringar. Vilka standarddokument ska finnas? Det finns inte, inte som jag har jobbat med. Men förhoppningsvis som med lite yngre personer, som kanske jobbar med det. Framförallt med de internationella systemen som man jobbar med i Indien till exempel.	ISD+K
31	LH	Men de gånger som du faktiskt har dokumentation när du ska göra underhållsarbete, tycker du att den är användbar då?	ISD
32	R1	Oftast.	ISD
33	LH	Oftast, det var ju skönt att höra i alla fall.	
34	R1	Om det nu är ett system som man ska ändra i som inte är ändrad utan dokumentation, lagt till metoder, tagit bort metoder, ändrat i databasen osv. Men de flesta systemen som jag har gjort har inte varit sådär gigantiska. Men det är skillnad på det och ett stort journalsystem som då gör jättemycket jobb. Och där det är kanske en 30–40 systemarbetare som har varit inne och jobbat i den. Det är övertid, där är det ju så, det vet ju jag från kollegor från andra håll som jobbar med journalsystem, inte minst i Skåne – att det är svårt att få fram	ISD



		tankarna bakom hur man har gjort en datamodell tillexempel. Ofta är det ju så att de som gör de här systemen och fixar och trixar, de är inte så skrivkunniga.	
35	MH	Nej okej.	
36	R1	Det är inga författare. Därför behövs det standardiseringar.	ISD+K
37	LH	Men du sa lite innan, du pratade lite om hur uppdaterade dokumenten var och sådär. Vi tänkte att vi skulle ta oss in lite på det spåret. Hur viktigt är det då för dig att uppdatera dokumentationen?	ISD+U
38	R1	Det borde vara mer viktigt än vad jag gör.	ISD+U
39	LH	Ja okej. När du då väl gör det. Varför känner du behovet av att göra det, alternativt när du inte gör det, varför känner du behovet av att inte göra det?	ISD+U
40	R1	Det handlar säkert, om jag tänker efter, mest på tid. Vilken tid har jag fått avsatt att göra det här. Oftast får man tiden avsatt att göra något, inte att skriva om vad man ska göra eller har gjort. Så är det och den tiden är jag inte riktigt säker på att kunden förstår att man måste betala för. *Namn på kollega*, han och jag har jobbat ihop jättemycket. Så när jag estimerade min utvecklingstid, då la jag alltid på 10% av min utvecklingstid, minst. Projektledningsjobb sa man att det var. Men det betyder ju att man skulle skriva om det, vad man har gjort både offentlig och konfidentiell. Våra mallar, exempelvis en Word-mall. Då måste man typa den om den är offentlig eller konfidentiell eller intern. Man ska skriva både och, så offentlig till kunden och konfidentiell till interna, så i ert fall med interna system, så ska de vara konfidentiella. Konfidentiella betyder inte bara att den är konfidentiell inom grupper som jobbar med system utan den är konfidentiell för alla som inte får tillstånd att jobba med den. Man måste ha anledning att titta i den.	ISD+U, ISD+T
41	LH	Ja, jo men det är klart. Men av vilken anledning uppdaterar du den interna dokumentationen?	ISD+U
42	R1	Det är för att komma ihåg vad jag har gjort.	ISD+U
43	LH	Ja, det är så pass enkelt. Men hur ofta sker det då?	ISD+U
44	MH	Att du uppdaterar?	ISD+U
45		Emellanåt, hahaha. Det beror på liksom. Och så som vi håller på nu exempel, *namn på kollega* och jag som jobbar på befolkningssystem. Jag ska göra en ny webbservicetjänst i Värmland. Det är beställt en ny webbservicetjänst som de ska kunna anropa från sitt journalsystem så man får fram patient med vissa typer av nummer. Det är en enskild webbservicetjänst som ska göra ett arbete. Den kommer jag ju att dokumentera för kunden. Alltså såhär anropar ni systemet, ni har de har parametrarna in, ni kan förvänta er att få ut en xml-sträng som ser ut såhär. Det är egentligen vad jag dokumenterar.	ISD+U
46	LH	Men hur ofta händer det att du dokumenterar för dig och för andra utvecklare?	ISD
47	R1	Väldigt sällan.	ISD

48	MH	Väldigt sällan, okej.	
49	R1	När det gäller kod och metoder. Man gör ju metoder i den objektorienterade världen som gör vissa saker. Då dokumenterar jag i kod. Det anropas såhär och den gör det här och den får tillbaka det här returneringsvärdet. Det är systemdokumentation. Något annat tror jag inte behövs. Jag tror inte man ska överarbeta det heller. Jag vet inte hur mycket folk läser det överhuvudtaget.	ISD
50	MH	Nej	
51	LH	Nej, men tycker du att dokumentationen eller dokumenterar du annorlunda beroende på vilken storlek av ändring i koden som har skett, säg om det är en mindre ändring, en större ändring. Skulle du säga att dokumentationsarbetet ser annorlunda ut?	ISD+U
52	R1	Ja, absolut. Det beror ju på mycket att det inte finns något standardiserat. Ingen möjlighet att standardisera det. Det skulle jag vilja ha. Jag skulle vilja ha en mall, excel-mall eller vad som helst, som säger att när du gör någonting så ska du dokumentera. För det gör man ju exempelvis, om jag nu gör en ändring i dokumentet för webbservicetjänsten, längst bak så gör man versionshantering. Man gör en sida sits som säger version 1 - skapat av *namn*. Sen nästa version 1.1, ändrade *namn* detta datum och då gjorde han det här. Detta gör man för att kunden ska veta vad som har hänt. Då samtidigt så är ju det en intern systemdokumentation då den är tillgänglig intern också.	ISD+U, ISD+K, ISD+T
53	LH	Skulle du säga att du skulle dokumentera oftare när det sker mindre kodändringar eller dokumenterar du oftare om det är en större kodändring, hur prioriterar du då?	ISD
54	R1	Den större naturligtvis.	ISD
55	LH	Så de är viktigare?	ISD
56	R1	Javisst.	ISD
57	LH	Ja, okej. Mmm.	
58	R1	Det är ju så. När man gör ett system så jobbar man med version och revision. Alltså, 1.0 det är version ett. 1.1 är fortfarande version, men revision 1 och så vidare. Det finns de ju de som har builds, att man gör nytt system, och den är build, den är inte lika väsentlig att systemdokumentera som en version eller revision. Det finns ju både i C# och Visual studio, där finns det automatiska dokumentationsverktyg. Man gör vissa kommentarer, snesträck och grejer i koden, som när man sen kompilerar och bygger så skriver den ut de sakerna man har skrivit. Men det är ju inte något strukturerat på något annat sätt än att man kan få ut vad man har skrivit och det är inte strukturerat på hur man skriver, utan vad man har skrivit. Det vet jag att både *namn* och *namn* i Danmark på QMS gör det. Det ser bra ut tycker jag. Där går jag in faktiskt ofta och tittar.	ISD+T, ISD+U, ISD+K
59	LH	Men vad bra. Använder du systemdokumentation även om den är gammal och inte uppdaterad så ofta?	ISD+U

60	R1	Jag brukar titta på den och se vad som är intentionerna och hur gammal är den, vad är det för version, revision som står i dokumentet till skillnad vad som står i kod. Är det en väldigt stor skillnad, då läser jag egentligen bara ingressen.	ISD+U, ISD+K
61	MH	Ja okej	
62	LH	På vilket sätt är den användbar då? Alltså om den är utdaterad och kanske inte riktigt det du vill ha? Kan du ändå använda den på något sätt?	ISD+U, ISD+K
63	R1	Ja, det kan jag för det finns ju information i den vem som har gjort det. Och de kan man ju också fråga.	ISD+U
64	LH	Ja okej, det var ju smart. Men skulle du säga att ju mer uppdaterad den är desto bättre och mer användbar är den?	ISD+U, ISD+K
65	R1	Ja absolut. Självklart. Och jag menar är den färsk så är det ju jättebra att läsa.	ISD+U
66	MH	Ja	
67	LH	Skulle du säga att skillnaden positivt är så pass stor att det är värt att lägga ner den tiden på att hålla dokumentationen uppdaterad?	ISD+U
68	R1	Absolut. Men kunden måste betala den tiden.	ISD+U
69	MH	Ja okej.	
70	R1	Eftersom kunden inte gör det och eftersom dokumentationsverktyget finns i typ Visual studio och Eclipse, att försöka kommentera sin kod och sen få ut den kommentaren i ett dokument. Det är en systemdokumentation i sig.	ISD
71	MH	Ja	
72	LH	Visst är det så. Hur skulle du säga att den här icke-uppdaterade dokumentationen påverkar delandet av kunskap?	ISD+U+ KD
73	R1	Att den inte är uppdaterad menar du?	ISD+U
74	LH	Ja precis.	
75	R1	Det är ju jättedåligt. Det är det ju faktiskt. Nu jobbar många systemutvecklare, inklusive mig själv, jobbar agilt, dvs i sprints. De är sprintarna måste dokumenteras. Det är ju en form av systemdokumentation det också ju. Så att de som jobbar i sprints och med agil utveckling, de har en fördel med att ha bra systemdokumentation. Men de som inte gör det, som mesta tiden som jag har jobbat, så fanns det ju inte något sånt. Det fanns inte agilt. Man jobbade på och levererade någonting som man tyckte va bra. Den dokumentationen, den blev ju gammal om det är så att man gick in och rättade buggar exempelvis. Och inte skrev ner vad man har gjort. Men i en sprint, den måste man dokumentera allting man gör.	ISD+U+ KD
76	LH	Men när du säger att det påverkade delandet av kunskapen dåligt, hur menar du då?	ISD+U+ KD
77	R1	Ja alltså den informationen som finns i dokumentationen är inte up-to-date så metoder och skärmbilder och allting sånt som kan finnas, de stämmer inte. Då blir man förvirrad. Är det verkligen samma system? Det är också såhär att det kan ju finnas många versioner av systemdokumentation i samma mapp, och kan man inte se utifrån på filnamn och såna saker	ISD+U+ KD, ISD+T

		vilken version det är, då kan man ofta ta fel, det har hänt mig. Det är otydligt.	
78	LH	Men den här förvirringen som du talar om, som kan ske när det inte är up-to-date, vad kan den leda till?	ISD+U+ KO
79	R1	Naturligtvis kan det ju leda till att man inte har den kunskapen om systemet som man borde ha. Också när kunden beställer nytt, så pratar ju de inte alls i termer om metoder, kod och databaser utan de säger att den här funktionen ska ändras och fungera på det här sättet, och finns inte funktionen beskriven i ett dokument, att den här funktionen som kunden pratar om, den motsvaras i de här koderna, eller den här databastabellerna som har de här relationerna. Då är man rätt så illa ute om man inte har jobbat med systemet innan.	ISD+U+ KO
80	LH	Ja, det kan man tänka sig. Men tycker du att den här dokumentationen som då inte är up-to-date påverkar systemutvecklingen och underhållsarbetet? Hur då skulle du säga i så fall?	ISD+U +KO
81	R1	Absolut, det måste bara en up-to-date dokumentation för att man överhuvudtaget ska kunna förändra och gå in och utveckla sitt system. Det är ju också så precis som jag säger, det är inte alltid man har en grupp som jobbar i en sprint, eller man jobbar på det agila sättet, utan plötsligt så får man till sig att – du det här gamla systemet i Visual Basic, det måste du ändra och de betalar en miljon för ändringarna. Finns det då ingen dokumentation, då tar det nästan en miljon att lära sig det här systemet. Men finns det dokumentation då läser man den först.	ISD+U +KO
82	LH	Ja. Men anser du att den här icke up-to-date dokumentationen alltid leder till negativitet?	ISD+U +KO
83	R1	Ja, oftast gör den väl det. Men det kan ju också vara en positiv del i det här att man får ta kontakt med människor som har gjort det här, skrivit det här. Fråga, hörrudu, har du inte uppdaterat den här? Ja och vad har du då gjort i så fall? Sätta en liten sporre i hjulet på dem om att de kanske borde uppdatera sin dokumentation. Men som sagt så är det ofta så att de inte har gjort det där för att de inte har fått den tiden.	ISD+U +KO
84	MH	Nej precis. Nu tänkte vi gå vidare till en annan egenskap som handlar om kvalitet av den interna systemdokumentationen. Vi undrar vad du anser motsvarar hög dokumentationskvalitet?	ISD+K
85	R1	Vilken nivå är vi på då? Är det koder och eller funktioner?	ISD+K
86	LH	Det är alla typer skulle man kunna säga, både dokumenten i text, kommentarer i koden och sådär.	ISD+K
87	R1	Jag skulle vilja ha strukturerad dokumentation. Den strukturen skall vara bestämd vetenskapligt och inte av någon egoistisk systemarkitekt som mig själv, utan det finns ju universitet och högskolor som har utarbetat strukturer på hur systemdokumentation ska se ut. Jag vet att det finns, vad heter de. I USA har jag själv gått sådana kurser, de heter någonting. Den strukturen är ju utifrån provad erfarenheter. Jag tycker att man ska följa den strukturen. Gör man inte det, då är man lite	ISD+K

		illa ute när det gäller att kunna dela med sig, för då kanske man skriver precis som en novell och det är inte meningen.	
88	MH	Så dina förväntningar på systemdokumentation är att den ska vara strukturerad. Anser du att ni följer det?	ISD+K
89	R1	Något så när men inte alls egentligen till fullast. Man gör ett litet dokument som tar upp det man gör, men det skulle också finnas inte bara att man jobbar i Word eller Excel, det borde finnas system för det webbsystem där man går in och är knutet till det här systemet. I vissa fall så gör man det. De här riktigt stora systemen de har, de har rätt så bra dokumentation.	ISD+K, ISD+T
90	MH	Ah okej. Vi har tagit upp några kvalitetsattribut som vi har hittat från litteratur som utvecklaren ser vara viktigt. Vi undrar hur viktiga du anser dessa är. Den första är korrekthet. Hur viktigt tycker du att det är med rätt översättningar, kommentarer är med systemdokumentation?	ISD+K
91	R1	Att det är korrekt förutsätter man. Annars finns det ingen anledning att läsa, om det är fel. Men utgångspunkten är att det ska vara korrekt. Dessutom är det såhär att det inte kan vara skrivit på modersmål i en sådan här organisation som vår. Jag har fått systemdokumentation på finska, och det är inte alltid lätt. Databastabellernas kolumnnamn heter också något på finska.	ISD+K
92	LH	Jaha oj	
93	R1	Och det är inte alls bra. Utan jobbar man med systemutveckling så ska det enligt praxis vara på engelska. Så att korrekthet för mig det är att det är skrivit på engelska.	ISD+K
94	MH	Okej. Hur viktigt är det då med läsbarheten? Stör du dig på felstavningar, grammatik, namnkonventioner?	ISD+K
95	R1	Nej, det gör jag inte. I den här världen så är systemdokumentation amerikanskan. Amerikanska är ju inte speciellt korrekt. De skriver lite hur de vill och säger hur de vill.	ISD+K
96	MH	Ja	
97	R1	Hade vi haft brittisk engelska hade det varit en annan sak, men nu är det amerikansk engelska.	ISD+K
98	MH	Ja. Nästa handlar då om struktur, den har du ju pratat ganska mycket om och sagt att det är något av det viktigaste.	ISD+K
99	R1	Ja	
100	MH	Då går vi till användbarhet. Hur viktigt är det att, ja	ISD+K
101	LH	Alltså hur det går att använda sig, att kvaliteten är så pass hög att den är användbar, kan man väl säga	ISD+K
102	MH	Att det uppgår målet med systemdokumentationen.	ISD+K
103	R1	Ja det är ju jätteviktigt. Man måste kunna använda den. Man måste även kunna lägga till saker. Där har jag även läst, jag drar mig till minnes nu, systemdokumentation som har skrivits strukturerat en gång i tiden, men där, vem det nu är, har skrivit icke-strukturerat uppdatering. Alltså vi har lagt till lalalala och det är inte alls bra för då får man inte alls något sammanhang. Är det strukturerat så ska man fortsätta på den strukturen och inte avvika från den.	ISD+K, ISD+U

104	MH	Nej precis. Den sista är då, hur anser du att det måste vara fullständigt, eller går det att använda dokument som är halvt skrivna i princip.	ISD+K
105	R1	Ja, om man kan veta vad halvt skrivna betyder. Alltså det är ju rätt svårt, det är subjektivt. Man dokumenterar ju inte varje kod rad, det går inte att göra. Om det är inter systemdokumentation som en annan systemutvecklare ska förstå, då dokumenterar man metoder och kanske också input och output. Men man dokumenterar inte varje kod rad för om man går in i koden så är den idag oftast väldigt välinstruerad i sig själv. Den enda koden som jag har programmerat i, som jag slutade med ganska snabbt, det är C++. C++ är gjort för att det ska gå snabbt och skriva, men har bokstavsförkortningar, variabelnamn med endast enkla namn, funktioner heter a, b, c och likande. Den är gjord för att man snabbt ska kunna skriva kod och metoder, men inte för att läsas. Men i C# och java och sådana, där är det ju väldigt, där ger man variabelnamn och metodnamn så man förstå vad de ska göra, ex get, set osv.	ISD+K
106	MH	Skulle du säga, när du själv skriver systemdokumentation, tar du hänsyn till de här kvalitetsattributen? Exempelvis, skriver du på engelska.	ISD+K
107	R1	Ja, det gör jag faktiskt. Och databasens kolumner och tabeller är skrivning på engelska. Däremot så är det så att ibland om jag vet at den här ska bara användas i Sverige, eller när jag gör något privat och då skriver jag på svenska. Men jag skriver fortfarande på databasen på engelska.	ISD+K
108	LH	Jaha, okej	
109	R1	Jag har en egen struktur som gör att jag fattar den.	ISD+K
110	MH	Ja. Hur ofta tycker du att den systemdokumentation som ni har på ditt arbete, hur ofta har den all nödvändig information som krävs för att du ska förstå systemet, underhålla systemet och använda systemet?	ISD+K
111	R1	Det är väldigt olika faktiskt. När man jobbar agilt med sprintar och sammansatta grupper, där är det ju systemarkitektens uppgift att hålla ihop de här dokumenten. Är det en bra dokumentation så blir det också bra sprintar. Är det ett litet system och det inte finns bra dokumentation, så gör inte det så mycket egentligen för man kan alltid snackas vid med den som har jobbat med det. Men det är ju klart, nu är jag ju inte så aktiv i mina kodbibliotek på *företaget* och jag vet inte om någon är inne och tittar på det. Men de som har gjort det och haft frågor, de har ju hört av sig. Än så länge kommer jag ihåg, jag är inte dement än.	ISD+K
112	LH	Det är ju bra	
113	MH	Hur tycker du då att intern systemdokumentation med dålig kvalitet påverkar systemutveckling eller underhållningsarbetet?	ISD+K+ KO
114	R1	Det borde ju påverka dåligt men det beror på programmerar och hur pass clever den här människan är. Om man kan se igenom dokumentationen och säga att det här stämmer ju inte.	ISD+K+ KO

		Den här dokumentationen kan jag ju inte använda. På något sätt så kan man ju säkert använda sig av, eller göra något i systemet utan att titta i dokumentationen, men det är ju väldigt mycket lättare om det finns dokumentation.	
115	MH	Då har vi den sista och tredje egenskapen som är tillgänglighet. Då undrar vi om det är lätt för dig att få tillgång till den interna systemdokumentationen när du behöver den?	ISD+T
116	R1	Ja, det är det faktiskt. Vi har ju då ett system som heter Teamer som ska gå upp i något annat. Det är egentligen ett file system med sharepoint system från Microsoft där man lägger filer i sina olika projekt.	ISD+T
117	LH	Okej.	
118	R1	Alla har inte alltid tillgång till alla, utan när man får ett uppdrag så får man tillgång till det här. Det fungerar väl bra faktiskt.	ISD+T
119	LH	Men hur går du tillväga om du är en av dem som inte har tillgång till det. Alltså som du sa, man blir tillagd i projektet när du ska ha projektet, men om du nu av någon anledning inte får tillgång, hur lätt är det för dig att exempelvis ringa någon, för att kunna få tillgång?	ISD+T
120	R1	På Teamer, när man kommer till försättssidan, den har alla tillgång till. På den försättssidan står det till exempel QMS, där finns en mapp som heter QMS och där bredvid står det också vem som är ansvarig. *namn på kollega*. Kommer man inte in där får man kontakta den här personen. Förhoppningsvis finns den här personen kvar i organisationen. Jag har stått på några sådana mappar där de har sagt, hur fasan gör vi nu då. Jag är ju inte kvar, men jag har ju faktiskt haft kvar mina administrättigheter. Jag var ju ansvarig för QMS men jag har gett *namn på kollega* alla rättigheter, jag kommer nästan inte in där nu.	ISD+T
121	LH	Nej så kan det ju vara	
122	R1	Nej hos oss så tycker jag att det är jättebra	
123	LH	Okej, vad bra	
124	MH	Då kan vi slopa nästa fråga	
125	LH	Ja men då möter det dina, eller det är så du vill ha det liksom	ISD+T
126	R1	Ja när det gäller dokument. För att egentligen som jag sa innan, så skulle jag ha velat ha ett system, jag hade velat ha ett webbsystem där jag kunde söka och få information online. Också kunna ändra, strukturera online. Precis som det finns för projektarbetet och sådana saker. Men jag tror att det finns sådana system, det tror jag nog. Men de är nog väldigt dyra.	ISD+T
127	MH	Vi har ju varit inne på detta innan, men sparas endast den senaste versionen av systemdokumentationen eller även äldre versioner?	ISD+T
128	R1	Det är äldre också	ISD+T
129	MH	Är det lätt att veta relevant, senast uppdaterad?	ISD+T
130	R1	Om man läser filnamnet, där står version 2 istället för 1.	ISD+T
131	MH	Hur tycker du att tillgängligheten av systemdokumentation bidrar till delning av kunskap?	ISD+T+ KD

132	R1	Det är ganska bra faktiskt. Man kan ju då gå in och se vem som har gjort det och fråga, hur gjorde du den här grejen. Jag har ju ett dokument som beskriver det kan jag säga då, då har de ju tillgång till det. I och med att det finns online på Teamer så är det ju tillgängligt för dem. Så det tycker jag funkar jättebra.	ISD+T
133	LH	Bra	
134	MH	Super. Om det finns begränsad tillgänglighet, hur tycker du då det påverkar systemutvecklingen eller underhållningsarbetet?	ISD+T+ KO
135	R1	Har man inte tillgång till något så har man inte kunskap. Så det är klart att en sådan tillgänglighet måste finnas. Om en systemutvecklare inte får tillgång till rätt dokumentation, då ska inte den utvecklaren jobba med systemet. Det är ju upp till ledaren, om det nu är arkitekten eller projektledaren eller säljaren i det här fallet då. Eller i många fall ska jag säga. Den är ansvarig för säljdokumentationen. Gå in och titta här och se hur lång tid du behöver estimerar, kan han säga till mig. Om man inte får det, då kan man inte jobba med det. Så är det, punkt slut.	ISD+T+ KO
136	MH	Ja	
137	R1	Det har också fungerat jättebra måste jag säga	ISD+T+ KO
138	LH	Vad bra, Vi har några avslutande frågor också	
139	MH	Vi har i princip inga fler frågor kring vårt arbete med undrar om det finns något som du hade velat lägga till som du kommit på under intervjuens gång, som kanske hade varit viktigt för oss att veta.	
140	R1	Det som jag tänker på det är ju just det som jag inledde med, det finns en stor brist i systemdokumentationen. Jag tror när ni intervjuar fler så kommer de också säga det. Vi borde nog dokumentera mer än vad vi gör, och har gjort. Det är ju lite synd, men oftast tror jag att det beror på två saker. Det beror på att det inte finns någon tradition och det beror också på att det inte finns någon tid. I vårt fall är tid pengar, det vill säga att någon måste betala den här tiden och det är ju kunden i slutändan. Så säljaren som säljer in det här har en jättestor påverkan på naturligtvis vad kunden köper. Om den här säljarpersonen kan förmedla till kunden att förutom projektledning och utveckling så måste vi ha dokumentationstid också. Jag tror att många säljare som jag har haft har försätt det, men jag vet att det finns säljare som inte har det.	SD, K, KO
141	LH	Ja.	
142	R1	Så brist på tid, det är ju säkert någonting som ni kommer att få höra med.	ISD+KO
143	LH	Ja, det känns så.	
144	R1	Det är inte kalendertid egentligen, det är projekttid. När jag estimerar, när en systemutvecklare estimerar, då estimerar man inte dokumentationstid. Det är inte det man tänker på. Man tänker på sittande kod, programmering. Så att, där har ju jag	ISD+KO



		mina säljare *namn* och *namn*. De har ju varit jätteduktiga och lagt på procentdelar på estimeringen som jag har gjort.	
145	MH	Sista frågan, är det okej om vi skulle ha några andra frågor kring ämnet eller om vi vill att du ska förtydliga något du har sagt.	
146	R1	Det tycker jag, det kan ni gott få göra.	
147	LH	Vad snällt.	
148	MH	Det var allt för oss då	
149	LH	Jag kom faktiskt på en sak, vi glömde i början att fråga en mycket inledande fråga. Vad gör företaget du jobbar på?	
150	R1	Ja just det, vi gör massa saker.	
151	LH	Lite kort går bra	
152	R1	Inom IT så är det, eller den delen jag jobbar med är healthcare. Hälso- och sjukvårdssystem inom kommuner och regioner, inte landsting längre. Inom hälso- och sjukvård så är det inte diagnostiska system som ex röntgen utan administrativa system. Typ journalsystem.	
153	LH	Och då också, vad är din roll, yrkes roll?	
154	R1	Min sista roll var systemarkitekt. Se till att man får in all information från säljarna som säljer systemen, med det upphandlingen. Se till att utvecklingarna förstår vad de ska göra och även skriva dokumentation.	
155	LH	Ja, okej. Kanon ju. Tack så jättemycket, det här gick ju jättebra.	
156	R1	Tack själva det här var ju jätteintressant.	
157	MH	Ja tack så jättemycket.	

## Bilaga C – Respondent 2, transkribering

R2: Respondent 2

LH: Linnéa Hansson

Rad	Person	Meningsenhet	Kod
1	LH	Jag tänkte då börja med att fråga om det är så att du skulle vilja vara anonym? Ditt namn kommer inte vara med, men om företaget ska vara anonymt eller om det är okej att vi har med det?	
2	R2	Alltihop får gärna vara helt anonymt.	
3	LH	Okej, bra.	
4	LH	Bara lite snabbt, vad håller företaget på med?	
5	R2	Vi är ett gigantiskt företag, men vi bygger IT-system i en mängd olika branscher, allt ifrån finans till oljeindustri till hälsosjukvård till skogsindustri till myndigheter och kommuner. Ja, vi har kunder egentligen genom alla möjliga branscher.	
6	LH	Ja, nice. Kanon ju.	
7	LH	Vad är då din arbetsroll? Eller din yrkestitel?	
8	R2	Systemutvecklare.	
9	LH	Ja, kanon. Okej, men det var nog det. Du sa att du ville inleda lite själv, så kan du köra igång nu.	
10	R2	Ja, om man pratar om det här området med dokumentation så ska det egentligen styras i ett företag av företagets kvalitetssystem.	ISD
11	LH	Mmm.	
12	R2	Och kvalitetssystemet definierar olika processer som företaget arbetar i då.	
13	LH	Okej.	
14	R2	Om man säger en IT-avdelning kan t.ex. vara berörd av processer för dokumenthantering eller dokumentkontroll och naturligtvis en utvecklingsprocess. Och den här första processen, den som handlar om dokumentkontroll, den handlar då om hur man skapar dokument, identifierar dokument, ett dokument livscykel och hur man vet att ett dokument är giltigt. Och har en viss kvalitet, om det är godkänt eller inte. Dokument måste också fasa ut när versionerna blir för gamla. Den andra utvecklingsprocess som man då har en idé om vilken metodik man vill använda när man utvecklar en produkt eller ett system.	ISD, ISD+U
15	LH	Mm, okej.	
16	R2	Så kan man också säga att det brukar också finnas en processbeskrivning för hur man underhåller gamla system. Egentligen så är det den viktigaste, för att ett IT-system lever oftast i decennier. Om dem överhuvudtaget används då också. Men om man säger då, eftersom det då också finns *går inte att uppfatta*. Om man då tar i utvecklingsprocessen så brukar det ju då finnas beroende på hur man arbetar med, men det kan ju	ISD, ISD+K

		då till exempel finnas någon form av användarkrav och det kan finnas tekniska krav. Dessa två använder man då dels för att försöka förtydliga vad är det vi ska implementera och sen också så finns det ju den andra sidan när man är klar då. Att dem tekniska kraven dem verifierar man genom test eller granskning, visar att man har uppfyllt dem och användarkraven dem validerar man då man visar att systemet har uppfyllt vad det ska göra då. Så det är väl kanske någon form av grund i det hela då. Och sedan så tar man ju fram andra kontrakt då runt systemet, så som programkod och sånt. Men ska man betrakta dem som dokument, det ses väl som en diskussionsfråga. Sen så finns det ju olika metodiker också för hur man bedriver en utvecklingsprocess som kan ha mer eller mindre olika dokument kopplade till sig. Detta ska ju då helst vara beskrivet i kvalitetssystemet hur man gör.	
17	LH	Mmm.	
18	R2	Och sen om man går över på när man har levererat sin första *inte uppfattbart* också så arbetar man ju ofta med någon form av change request, som sedan leder till att man ska uppdatera både dokumentation och programkod.	ISD
19	LH	Ja.	
20	R2	Och sen så, först verifieras och sen validerar man att man har gjort detta då. Och sen förutom detta så finns det då projektledare, processer då för hur dem ska skriva sina planer och göra sina budgetar. Sådana saker.	ISD
21	LH	Ja, okej.	
22	R2	Så det är egentligen som jag ser ramverket runt detta med systemdokumentation att det inte ska vara ett fritt tänkande, utan det ska utgå från ett kvalitetssystem som bestämmer vilken typ av dokument tar man fram och hur hanterar vi dem här dokumentens livscykel. Hur identifierar vi dem, hur hanterar vi deras livscykel.	ISD, ISD+K
23	LH	Mmm. Det var ju jättebra input. Jättebra.	
24	R2	Innan jag var systemutvecklare var jag projektledare, bland annat på medicintekniska företag.	
25	LH	Ja okej.	
26	R2	Jag har arbetat där och medicinteknik dem är väldigt hård med det här med dokumentationen att man ska kunna spåra och följa allting. Det här var lightversionen av hur man gör då, men det ska inte vara den enskilda personer som tycker eller tänker om detta utan man ska göra på ett och samma sätt.	ISD+K
27	LH	Ja.	
28	R2	I IT-världen så är man aldrig, eller extremt sällan, så är man lika strikt som man är inom medicinteknik. Därför att i medicinteknik så regleras det här av lagstiftning och standard som man är tvungen att följa.	ISD+K
29	LH	Ja, men precis.	
30	R2	Det är likadant inom finans, det är likadant inom transportindustrin, till exempel bilindustrin har sådana regelverk, flygindustrin har sådana här regelverk.	ISD+K

31	LH	Nä, jag har förstått det nu när vi har arbetet med det här att det är osämja inom IT-världen med detta. Man vet inte riktigt hur man ska göra och det är därför det då är så intressant att göra den här undersökningen för att se vad folk faktiskt tycker.	
32	R2	Ja, man behöver inte ha någon osämja, men man behöver bestämma sig vilken typ av kvalitetssystem ska man ha, vad är det man vill uppnå. Osämja uppstår när man inte har bestämt det, för då finns det alltid någon som tycker vi ska göra lite mer och göra si eller så, så finns det någon annan som tycker att det är helt meningslöst att skriva dokumentation. Vad man ska läsa i koden det är den som är verkligheten.	ISD+K
33	LH	Ja, men precis, exakt så har vi också...	
34	R2	Ja, jo...	
35	LH	Förlåt, sa du någonting.	
36	R2	Ja, nä, jag tänkte bara när jag läser era frågor att jag ville ge den här bakgrunden för annars skulle mina svar vara så konstiga på en del av frågorna.	
37	LH	Jaha, okej, ja men det var verkligen jättebra inledande. Det gav mig även tankeställare och förståelse, så det ska bli intressant och se vad du säger fortsatt.	
38	LH	Tänkte att vi kör igång lite med frågor.	
39	R2	Ja, ta dina frågor nu så ska jag svara på dem utefter bästa förmåga.	
40	LH	Kanon. Men är det så att du aktivt använder dig av intern systemdokumentation i ditt vardagliga arbete?	ISD
41	R2	Ja och nej, vi följer ju inget kvalitetssystem, så vi tillhör den här sorten som har ad-hoc dokumentation.	ISD, ISD+K
42	LH	Okej.	
43	R2	Officiellt utåt så följer vi ett kvalitetssystem. Vi säger till våra kunder att vi följer ett kvalitetssystem. I praktiken så gör vi det inte.	ISD, ISD+K
44	LH	Okej.	
45	LH	Men till vilken grad är det så att du använder dig av det eller kanske tidigare i din jobberfarenhet har du använd dig av förut eller så?	ISD
46	R2	Ja, jag har arbetat i olika branscher och på en del har det varit, i medicinteknik har det nog varit mest dokumentation och telekommunikation och elektronik näst mest.	ISD
47	LH	Mmm.	
48	R2	Om det är IT-projekt så brukar de alltid gå lite hejsan hoppsan.	
49	LH	Men när du jobbar nu så är det ingenting du gör jätteaktivt?	ISD
50	R2	Jo, vi har utvecklardokumentation som styrd av vad vi behöver göra för att vi ska kunna kommunicera inom teamet.	ISD+K D
51	LH	Mm, ah okej.	
52	R2	Så att den dokumentationen, om de andra utvecklarna frågar, borde du inte ha skrivit ner hur man gör på det här stället, så skriver vi ner det och håller det i en Wiki.	ISD

53	LH	Ah okej. Men anser du att dem här dokumenten eller dem här som du skriver ner i Wiki att de förenklar ditt arbete på något sätt?	ISD
54	R2	Ja, det förenklar kommunikationen inom gruppen eftersom vi slipper gå och fråga varandra om vi har skrivit ner någonting då så sparar vi tid genom att vi inte sitter och svara muntligt på frågor som går att slå upp då.	ISD+K D
55	LH	Precis. Vad tycker du då att den här interna systemdokumentationen ska uppnå för kriterier för att vara användbar för dig?	ISD
56	R2	Ja, alltså, det beror ju liksom egentligen, det är ju egentligen ett företagsbeslut vilken nivå av kvalitetssäkring man vill lägga sig på. Och så eftersom av oss utvecklare krävs inte något annat än att vi ska hålla ordning på vad vi själva gör så är det den nivån vi lägger oss på. Så att det är liksom ett managementbeslut att vi ska använda ett kvalitetssystem.	ISD, ISD+K
57	LH	Okej.	
58	LH	Förlitar du dig på den dokumentation som finns? Att din kompanjon har dokumenterat exakt som det är?	ISD
59	R2	Ja, det är också lite olika. Det beror på, ibland är det en instruktion hur gör jag när jag logga in mig på den och den kundens VPN och så. Jag litar på att andra har skrivit rätt, om det inte stämmer så får man gå och fråga vilket är ganska enkelt. Men sen börjar vi närma oss riktig kvalitetssäkring när vi skriver specifikationer, som vi ibland behöver göra för att beskriva ett tjänstekontrakt eller API, då måste vi vara väldigt exakta och då granskar vi dokumenten flera personer, så att vi vet att det stämmer. Om vi skriver något som inte stämmer blir det bara så jobbigt för då kommer någon annan implementera någonting som inte fungerar och så kommer vi få hålla på och rätta i ett senare led.	ISD, ISD+K, ISD+K O
60	LH	Mm, okej.	
61	R2	Fel kostar mer och mer ju längre fram i processen man kommer för att rätta dem. Så det är bättre att göra rätt från början. Så där är vi väldigt exakta och håller hög nivå.	ISD+K O
62	LH	Okej, så bra.	
63	LH	Jag tänker vi ska ta detta lite mer in på kunskap, för att dokumentera är ju ändå ett sätt att hantera kunskap inom företaget. Så jag tänker, hur fungerar kunskapsdelningen inom ditt företag eller ditt team?	KD
64	R2	*Tyst*	
65	LH	Är det så att ni skriver ner för att sprida kunskap, eller är det snarare att ni försöker sprida vidare genom muntliga konversationer?	KD
66	R2	Jag skulle väl säga att 70% av kunskapsöverföringen är väl muntlig, att man ber någon annan visa en hur gör jag det här. Jag har det här problemet att lösa, vet du hur man löser det och sen så är det någon annan som visar en hur man ska göra med just det. En person kanske säger att vi kanske skulle kunna skriva ner det här så att det flyter över lite i andra... Efterhand	KD, ISD+K D

		så har vi hållit på att bygga upp en bättre kunskapsmassa när det gäller hur man hanterar olika komponenter och ja, det är liksom alla möjliga saker, från inlogningar på annat VPN, till hur man bygger och konfigurerar ett visst system eller en del av ett system. Det kan ju va olika saker.	
67	LH	Yes, okej.	
68	LH	Sen när det då är någon kunskapsbesittande person som ska närma sig att lämna företaget, det kan va pension eller byta jobb eller något i den stilen, hur gör ni då för att behålla den här kunskapen?	KD
69	R2	Mmm. Det är ett problem och då har jag ju sett i vår grupp att det finns flera personer som är 60+ och alla vi är ju ett problem med den kunskapen vi sitter på. Men där brukar vi ju sköta det genom överlämningsprojekt. Vi har ju också det problemet med att många av oss är experter på just ett visst smalt område, som ingen annan kan. Så det är definitivt ett problem.	KD
70	LH	Mmm, ja.	
71	LH	Är det så att du ofta arbetar med underhåll av redan befintliga system?	
72	R2	Ja, i princip så är det ju i hela vårt liv att vi arbetar med existerande system som vi håller på att göra förbättringar på.	
73	LH	Okej. Hur brukar du då gå till väga för att hitta den kunskap som krävs för att du ska kunna utföra det arbete till fullo?	KD
74	R2	Ja, jag är så lyckligt lottad att jag nästan enbart arbetar med system som jag antingen själv har byggt eller varit med och byggt.	KD
75	LH	Okej.	
76	R2	Men, om det, för dem som arbetar med dem här *namn på system* baserade systemen då när man är inne i dem, så kanske om man ska göra en liten ändring, som kanske tar två timmar, det kräver kanske 3 - 4 dagar att jag sitter och försöker förstå hur programmet fungerar och ber om hjälp från någon av dem andra för att få en förklaring. Till slut gör man då ändringen, sen lägger jag några dagar till på att testa det.	KD
77	LH	Mmm, okej.	
78	R2	Så det är lite olika. Här finns det bara och det är att gå in och läsa programkoden.	ISD
79	LH	Okej, tror du det hade förenklats om du då redan hade lite dokumentation på detta sen innan? Eller tycker du att det är bättre att gå in och läsa koden?	ISD+K D
80	R2	Till att börja med hade det underlättat om de inte hade spaghetti-programmerat, så att man kunde läsa koden. Ja, det hade ju varit bra om det hade funnits designdokumentation, som hade beskrivit vad systemet överhuvudtaget gör, så att det inte bara var en muntlig förklaring från en kollega.	ISD+K D
81	LH	Ja men precis. Kanon.	
82	LH	Då tänker jag vi ska gå vidare till ett annat steg. Det är så att vi har hittat tre stycken egenskaper, kan man säga, problemområden, kring den interna systemdokumentationen,	ISD+U

		som vi har hittat via litteratur och andra studier. Och den första av dem är då uppdatering av dokumentationen.	
83	LH	Hur viktigt är det för dig att uppdatera den interna dokumentationen som finns?	ISD+U
84	LH	Vi har nämligen hittat mycket om att det är ett problem att den inte uppdateras och det finns mycket utdaterad dokumentation.	ISD+U
85	R2	Ja. Beklagligt är det nog så att det är viktigare för den jag ska lämna över information till att jag har gjort det här jobbet än vad det är för mig att göra det. Men, man säger då om man har ett kvalitetssystem som man följer där man ska göra det och där leveransen inte godkänns om du inte gjort det så är det metodiken för att det görs. Därför att man inte får leverera om man inte gjort alla steg i sin kvalitetssäkring. Och det där är sättet som man använder för att se till att detta blir gjort, så det ska inte bygga på någon frivillighet från respektive utvecklare.	ISD+U, ISD+K
86	LH	Känner du i din arbetsgrupp att man uppdaterar systemdokumentationen? Är det kanske ett beslut ni tagit?	ISD+U
87	R2	Alltså, vi uppdaterar ju den information som vi tycker att vi själva har mest nytta av. Den är ju ingen som tvingar oss att göra det och därför prioriteras testning och omedelbara behov istället.	ISD+U
88	LH	Hur ofta skulle du säga att det händer att du har suttit i ett sådant arbete där du har uppdaterat en systemdokumentation?	ISD+U
89	R2	På den här arbetsplatsen eller på andra arbetsplatser?	
90	LH	Generellt.	
91	R2	På den här arbetsplatsen, ja, vi gör ju det varje månad som minimum.	ISD+U
92	LH	Okej.	
93	LH	Skulle du säga att man uppdaterar och dokumenterar annorlunda beroende på storleken av ändringen i koden? I fall det är en mindre kodändring eller en större kodändring, gör man olika?	ISD+U
94	R2	Jag skulle säga att det är omvänt professionellt förhållande. Ju större ändringen är desto mindre benägen är man att ändra i dokumentationen.	ISD+U
95	LH	Varför tror du så?	ISD+U
96	R2	Därför då blir det två stora jobb som man staplar på varandra och då... jaa. Förstår du vad jag menar?	ISD+U
97	LH	Ja, absolut, det gör jag.	
98	R2	Om det är frivilligt för utvecklaren att göra någonting och man kanske har ett visst antal timmar på sig att göra någonting, då kommer utvecklaren alltid att prioritera att koden blir färdig och blir rätt. Finns det sen inte timmar över eller det inte är obligatoriskt så finns det en stor risk för att dem stora dokumentändringarna inte blir gjorda. Om det däremot är en liten ändring så är det bara att göra den lilla ändringen här och den lilla ändringen där och sen har man gjort det.	ISD+U
99	LH	Ah precis. Det låter rimligt.	

100	LH	Men, finns det någonting som skulle få dig mer motiverad till att göra dem större ändringarna eller är det mycket det här med tiden som sätter stopp?	ISD+U
101	R2	Nä, jag kan säga att jag väl inte är sådär moraliskt bättre än vad någon annan är så att är det ingen som tvingar mig göra det så gör jag inte det. Men det ska inte vara en fråga om den enskilda personens val utan det ska vara en industriell process som ser till att det händer.	ISD+U
102	LH	Ja, okej.	
103	E2	Det ska inte vara så att X är duktig på att dokumentera, någonting sådant finns inte, utan antingen så är detta ett processtänk som vi gör eller om vi inte har det processtänket så blir det inte gjort.	ISD
104	LH	Ja, okej.	
105	R2	Men i IT-världen är det oftast, det är liksom reducerat till att X är en jätteduktig utvecklare och jättebra och ordentlig person och hon är så duktig och hon gör det och det och det, titta på henne hur bra hon gör. Men, om du på ett medicintekniskt företag så är där liksom inte frågan om X är flitig och Y är lat utan det är snarare att bägge ska göra det och bägge ska granskas. Slarvar Y får han en bakläxa tills det är tillräckligt bra.	ISD
106	LH	Precis. Det handlar inte om individuellt tyckande.	
107	R2	Precis, det är inget individuellt tyckande. Det är lite det budskapet jag kör som en röd tråd i svaren.	
108	R2	Antingen har man en kvalitetsstämpel, som säger att såhär ska vi göra och alla följer det eller så är det upp till varje person och då blir det också efter vad varje person tycker och tänker om det.	ISD+K
109	LH	Men om du skulle stöta på en dokumentation som du ser är gammal, skulle du använda dig av den ändå?	ISD+U
110	R2	Nä, det är fullt möjligt att jag skulle försöka ta reda på vad systemet egentligen gör och stämma av om dokumentationen stämmer och uppdatera den innan jag går vidare och gör en ändring. Det finns en sannolikhet för att jag personligen skulle göra så.	ISD+U
111	LH	Kanon. Skulle du säga att en dokumentation som inte är uppdaterad och är lite gammal påverkar delandet av kunskap på något sätt inom gruppen?	ISD+U +KD
112	R2	Ja. Jag vet inte riktigt vad jag ska säga. Så blir det ju naturligtvis. Och det finns väldigt mycket personberoende inom IT-världen. Att det finns individer som håller sina kunder och sina företag som gisslan. Därför att dem sitter på viss kunskap som dem inte vill dela med sig av.	ISD+U +KD
113	LH	Ah, då är det klart det påverkar om man inte vill skriva ner det i dokumentationen eller dela med sig, som du säger.	
114	R2	Speciellt inom konsultvärlden finns det en typ av låntidskonsult som tar 1800 kronor i timmen eller liknande och aldrig delar med sig av den kunskap dem har och det är bara dem som kan ändra på ett visst system och dem kämpar stenhårt för att dessa	KD



		system inte ska bytas ut så att de ska kunna fortsätta fakturera sina 1800 kronor i timmen.	
115	LH	Mmm. Precis. Lite synd verkligen, men så är det ju.	
116	R2	Speciell variant av onda och dåliga människor. Dem vill köra fina bilar.	
117	LH	Men, den här inte uppdaterade dokumentationen ser du att den påverkar systemutvecklingen eller underhållsarbetet eller hela systemet på något sätt? Att den då är utdaterad, gammal.	ISD+U +KO
118	R2	Det beror ju på vem det är som ska göra ändringen. Är det en person som ändå kan systemet och vet hur den fungerar så påverkar det inte ett dugg för dem kan göra dem här ändringarna utan att göra det. Däremot så kan det ju vara så att kombinationen av illa skriven kod och obefintlig dokumentation gör det omöjligt för någon annan eller någon ny person att kunna börja arbeta med just det systemet.	ISD+U +KO
119	LH	Hmm, mm. Det var nog det vi hade på uppdatering. Tänker att vi går vidare till kvalitet, vilket ju är någonting som du ändå har berört, som då också är en av dem här egenskaperna/problemområdena inom den interna systemdokumentationen.	ISD+K
120	LH	Vad skulle du säga motsvarar hög dokumentationskvalitet?	ISD+K
121	R2	Hmm, ja det är ju egentligen bara först att vi börjar med faktorerna, att det är skrivet på ett grammatiskt, korrekt och rättstavat språk. Att det på något vis är lättläst och pedagogiskt, att det förhåller sig till saken, att det är exakt och korrekt. Det finns ju liksom olika typer av dokument, om man tar en traditionell kravspecifikation som är nedbruten i kärn och massmeningar så är det ju inte någon skönlitterär roman utan där skriver man att the lowest value shall be och så sätter man något värde och så får det inte vara lägre än det värdet för att värdena skall accepteras. Teknisk korrekthet är en av de viktigaste när det gäller teknisk dokumentation, att det stämmer överens. Och sen att man har en kvalitetssäkringsprocess där dokumentet är granskat, versionhanteras och sedan kanske framförallt att det lagras på ett sätt så at man kan hitta det, att det finns någon form av index eller innehållsförteckning eller sökfunktion som gör att man kan hitta dokumentet.	ISD+K, ISD+T
122	LH	Ah, okej. Skulle du säga att detta även är dina förväntningar på dokumentationen?	ISD+K
123	R2	Ja, verkligen.	
124	LH	Uppfylls detta i ditt dagliga arbete?	ISD+K
125	R2	Nej, verkligen inte. Vi saknar unik identifikation på dokumenten. Vi har inget dokumenthanteringssystem där vi kan leta rätt på ett dokument och veta om det är godkänt eller inte.	ISD+K, ISD+T
126	LH	Vi har kollat på några olika kvalitetsattribut, vilka du har nämnt några redan. Exempelvis läsbarhet så tyckte du det skulle vara rätt stavningar etcetera Vi har några attribut vi tagit fram som många tycker är viktiga och vill kolla lite med dig om hur du ser på det?	ISD+K

127	LH	Den första är korrekthet. Det pratade du ju lite om. Det handlar om översättningar, kommentarer, att innehållet ska vara korrekt helt enkelt. Det tycker du är viktigt?	ISD+K
128	R2	Absolut.	ISD+K
129	LH	Även läsbarhet. Det nämnde du ju med grammatik och stavningar, kan även vara namnkonventioner och lite sådant.	ISD+K
130	R2	Ja, namnkonventioner då är vi kanske lite mer över på kod, men det kan förekomma i dokumentationen också.	ISD+K
131	LH	Ja, precis. Sen kommer det här med struktur, standarder för hur dokumentationen ska vara utformad. Tycker du det är viktigt?	ISD+K
132	R2	Ja, det beror lite på vad man menar med struktur.	ISD+K
133	LH	Såklart. Mm, möjligtvis att dokumenten ska vara uppbyggda på samma sätt.	ISD+K
134	R2	Om man har ett kvalitetssystem som finns det ju vissa formalia med hur ett acceptabelt dokument ska se ut och att det ska finnas en introduktion, att man skriver syftet med målgruppen, vem ska läsa det. Att man har med en ändringshistorik. Att man har med en unik identifierare och att det klart och tydligt framgår vem det är som har granskat och hur denna har granskat dokumentet. Vem har godkänt det. Det finns några hänvisningar till ett granskningsprotokoll, att man har granskningsmöten.	ISD+K
135	LH	Sen har vi fullständighet, det talar lite om att dokumenten är färdigställda och innehåller det nödvändigaste. Det finns nämligen tillfällen där dokumentationen kan vara halva nästa, att man har påbörjat, men inte riktigt färdigställt det. Anser du det viktigt?	ISD+K
136	R2	Ja, men då är dokumentet inte färdigt, då är det snarare ett draft.	ISD+K
137	LH	Yes. Skulle du säga att du tar hänsyn till dessa attribut när du dokumenterar?	ISD+K
138	R2	Alltså, eftersom vi inte tillämpar ett kvalitetssystem och vi mest just i min grupp använder Wiki för att dokumentera, så befinner det sig i ett ständigt draft-tillstånd. Det är också en metodik att man inte har distinkta punkter där man svär att nu är det godkänt utan att man har ett ständigt levande dokument och det är vissa som hävdar att det är bättre att denna modellen med att ha strikta punkter där man säger att nu har vi granskat och nu har vi godkänt och att versionen är den korrekta, att det finns en grupp med lite mer fria själar som tycker att den typen av dokumentationen är felvänt och felaktigt att göra på det viset, därför att den är föråldrat samma dag som man godkänt den. Det viktiga är då att man uppdaterar och håller sin dokumentation levande och skriver ner vad det är som gäller för varje dag och för varje gång man ändrar. Att man inte behöver ha någon formell granskning utan att det liksom, man skriver för sina kompisar i teamet. Ja. Det kan finnas vissa poänger med det resonemanget också. Vi lutar ju mer åt det andra hållet än åt det första hållet.	ISD+K, ISD+U

139	LH	Ja. Hur tycker du att intern systemdokumentation med lite dålig kvalitet kan påverka systemutveckling, underhållsarbetet och systemet i sig?	ISD+K +KO
140	R2	Det beror på. Lite som samma svar som innan. Det beror ju på då om det är någon som redan kan sitt system framlänges och baklänges. Vissa delar på system där jag kommer ihåg koden mer eller mindre i huvudet då kan jag göra en ny tilläggsdesign i huvudet, jag gjorde en sådan förra veckan till en ny funktion, men behövde inte skriva ner någonting. Då sa jag bara till *namn* att jag ska lägga till en sån och en sån och sen ska jag ändra där och där, vilket jag kunde säga muntligt. Och då sa hen det blir säkert bra, jag får titta på det när det är färdigt. Och hen likadan också, så ibland eftersom det är hen och jag som arbetar med frontend, ibland så lägger hen till saker och så får jag säga vänta vänta jag hänger inte med vad har du gjort nu, nu får du berätta för mig vilka ändringar det är. Så det beror ju på. Men problemet uppstår ju framförallt när man byter människor i teamet, någon blir sjukskriven eller går i pension eller byter jobb eller något liknande och ska några nya människor ta sig in i det.	ISD+K +KO
141	LH	Ja men precis, det kan jag verkligen tänka mig.	
142	R2	Det blir en lång startsträcka för dem som kommer nya.	ISD+K +KO
143	LH	Kanon. Vi går vidare till det sista stycket tillgänglighet. Vilket du också redan varit lite inne på då du nämnde att ni inte har något dokumenthanteringssystem, men skulle du ändå säga att det är lätt att få tillgång till den dokumentation som ni har när du behöver den?	ISD+T
144	R2	Nä, om det inte finns på Wikin och jag ska försöka leta rätt på ett Excel eller Word-dokument, det kan till och med vara så illa att jag varit en ekorre och gått och gömt mina nötter någonstans. Så det är sådär okej i vilken mapp ligger detta? Och det finns inget index för att leta rätt på det hela, det finns inget unik ID på detta dokument som jag kan använda för att söka på, utan det är helt upp till min egen demenshjärna var jag la det någonstans. Det är ju ingens fel, men det är en kombination av hur det är. Vi har haft detta uppe till diskussion i hela teamet, i hela den större gruppen. Då är det några som sagt att vi gör en ny mappstruktur på den och den servern också, men när man gör sådana system så blir den struktur bli bra för en sak men dåligt för allt annat. Man behöver ha något dokumenthanteringssystem, där man dels kan ha ID på dokumentet dels tagga dem med olika nyckelord, så att man kan söka på både titel och innehåll och på dem här nyckelorden för att kunna hitta dokument.	ISD+T
145	LH	Ah, det låter ju verkligen smidigt.	
146	R2	Det finns sådana system.	ISD+T
147	LH	Ja, jag har sett det, men jag har förstått att det är ganska dyrt.	ISD+T
148	R2	Ja.	ISD+T
149	LH	Ah.	

150	LH	Sparar ni endast den senaste versionen av dokument eller finns det även äldre versioner sparade?	ISD+T
151	R2	Det är ju så att vi har gått över till att använda Teams för att spara våra dokument och den bygger på sharepoint och där kan man gå tillbaka och hitta en historik på sina dokument. Vad man absolut inte ska göra är att använda, vilket ju görs väldigt ofta väldigt ofta, är att man inte ska kalla sina dokument för namn streck 1.0 eller namn och streck 2.0, eller något sådant, för då har man ofta en hel härva med dokument som ligger där då. Man kan säga att jag inte är helt hundra procent klar på hur jag kan leta på historiken, men det ska gå att göra.	ISD+T
152	LH	Okej. Hmm, nu kommer ju lite liknande, vi avslutar ju varje del med lite liknande frågor, jag kan tänka mig att du kanske har lite samma svar igen, men då är ifall den begränsade tillgängligheten kan påverka systemutvecklingen, underhållsarbetet och då även systemet i sig? Är det så att du tycker att det är samma, lite likt innehåll eller är där något annat du skulle vilja lyfta?	ISD+T +KO
153	R2	Nä, jag kommer svara samma sak, den som redan kan den kommer inte ha några problem, medan den som är ny och som inte kan området kommer att ha problem.	ISD+T +KO
154	LH	Yes, det låter rimligt. I så fall var vi faktiskt klara och jag skulle bara vilja fråga om det är någonting du skulle vilja tillägga, som du har kommit att tänka på eller om du känner dig nöjd?	
155	R2	Nä, jag tycker ni har tänkt igenom området med era frågor. Och sen hoppas att jag får igenom synpunkten att där är två vägar att gå, antingen har man ett formellt kvalitetssystem som säger att detta är en del av vår process och som en del av vår process tar vi fram den och den dokumentationen och att dokumentationen ska uppfylla dem och dem kriterierna. De ska även vara godkända och detta är liksom en del av vår leverans, så beskriver inte vart går det utan vi gör alltihop, det är liksom en del av allt vi gör och det ingår i våra offerter. Och det andra sättet är då att man lämnar det till varje persons goda vilja, hur mycket dem har lust att göra.	ISD+K
156	LH	Ja.	
157		Det ni kanske också får om ni intervjuat andra är ju att det finns olika åsikter, att någon säger det ska göras så, men så saknas det. Det är min huvudsynpunkt.	
158	LH	Ja, men det tycker jag verkligen har kommit fram med det du sagt.	
169	LH	Är det okej om vi skulle kunna kontakta dig om vi har fler frågor eller om vi vill förtydliga någonting du sagt i efterhand?	
160	R2	Ja.	
161	LH	Men, då tackar vi så mycket för detta!	

## Bilaga D – Respondent 3, transkribering

R3: Respondent 3

LH: Linnéa Hansson

MH: Mathilda Hezsö

Rad	Person	Meningsenhet	Kod
1	MH	Vi vill börja med att säga att både företagsnamn och ditt namn kommer att vara anonymt.	
2	LH	För att sätta igång startar vi med några introduktionsfrågor.	
3	LH	Vad gör företaget som du jobbar på?	
4	R3	*Företaget namn* är inom två områden i Sverige kan jag tycka. Det är ju integration och analytics. Sen finns det en del andra specialavdelningar också, men det är de två huvudområdena enligt mig.	
5	LH	Vad är din roll på företaget och vad arbetar du med?	
6	R3	Jag jobbar på integrationsavdelningen som QA manager, det vill säga kvalitetssäkring och test.	
7	LH	När vi pratade lite på mejl så sa du att du inte var säker på om du kunde svara på alla frågorna ur ett test/QA-perspektiv. Vi vill ändå säga att om du kan svara på de ur ett systemutvecklarperspektiv, om du har jobbat med det, så är vi intresserade av alla möjligt perspektiv. Men annars så är det bara att passa på frågan om du känner att du inte kan svara. Känns det okej?	
8	R3	Ja, det blir fel känner jag om jag ska svara ur ett systemutvecklarperspektiv då jag inte utvecklat på 10+ år. Men jag jobbar med utvecklare så jag kan svara från hur jag uppfattar deras perspektiv.	
9	LH	Du kan göra hur du känner. Känner du ändå att du inte riktigt kan svara så är det helt okej, då bara skippar vi den frågan.	
10	MH	Vi tänkte första gå in lite på intern systemdokumentation och undrar om du aktivt använder dig av intern systemdokumentation i ditt vardagliga arbete?	ISD
11	R3	Nej	ISD
12	MH	Vad ska intern systemdokumentation uppnå för kriterier för att den skulle kunna vara användbar för dig?	ISD
13	R3	Aktuell	ISD
14	LH	När du säger aktuell, vad menar du då?	ISD
15	R3	Ja det är ju inte helt ovanligt att systemdokumentation blir föråldrad för att man, speciellt i this day and age som är inom utvecklarområdet där allting går så väldigt fort. Jag kan tänka mig att många dokumenterar i koden och att tyvärr finns det de som gör ändringar och dels då inte testar alternativt inte uppdaterar sin kod, kommentarerna i koden.	ISD+U
16	LH	Vi har förstått det också	ISD
17	MH	Hur skulle du säga att man förlitar sig på den interna systemdokumentationen som ni har?	ISD

18	R3	Jag får svara lite generellt eftersom jag inte är detaljinsatt. Det är väldigt personligt faktiskt tycker jag. Har jag märkt. Nu är jag så gammal och säger efter 18 år i branschen. Om man förenklar det lite och drar det på någon skala, seniora är mognare och vanliga med dokumentation. Kan man nu också dra någon skala att: visionär, skjuta från höften, kreativ – mindre dokumentation.	ISD
19	MH	Hur tycker du då att kunskapsdelning fungerar inom ert företag? Delar du helst kunskap via dokumentation eller muntliga konversationer?	ISD+KD
20	R3	Dokumentation tycker jag fungerar, speciellt nu när ni säger den frågan, när vi har förvaltningsåtaganden. Dock blir det ju KB – knowledge based – så fort, ja, i utveckling eller i vår utveckling, produktion så löser man det, så skriver man en KB på, där händer det och såhär fixar man det för att nästa person som råkar på något likande, jaha. Då fungerar dokumentation.	ISD
21	LH	Vad skönt	
22	MH	Hur gör ni när kunskapsbesittande personer närmar sig att lämna företaget?	KD
23	MH	Försvinner den kunskapen eller försöker man bevara den på något sätt?	KD
24	R3	Alltså, ja och nej. Dels så naturligtvis så omsköts *inte hörbart* de här KBs och man har ju också någon överlämningsperiod och då vanligtvis så är det några som sitter bredvid varandra i några dagar. Men då skrapar du ju liksom x antal procent av den kunskapen. Så när någon slutar så är det ju den personen, förmågan och drivet, attityden, det är den man tappar.	ISD+KD
25	MH	Arbetar du med underhållsarbete av system?	ISD
26	R3	Nej	ISD
27	LH	Okej, men genom litteratur och andra studier så har vi hittat tre egenskaper, problemområden inom intern systemdokumentation och vi har tänkt ställa lite mer specifika frågor kring dessa.	
28	R3	Du, får jag fråga där, det slog mig, när ni säger intern systemdokumentation, vad menar ni då? För man har ju produktbolag som *stort bolag*, de har intern systemdokumentation. Men vi som konsultbolag har ju aldrig intern systemdokumentation för vi gör ju allt åt kunden.	ISD
29	LH	Ja vi menar den dokumentation som utvecklare skriver för andra utvecklare. Så det kan vara att om jag har byggt ett system så vill jag skriva ner att dessa funktioner gör det här, dessa metoder gör det här, för att i ett senare led ska någon annan jag, alternativt samma person kunna gå in i systemet igen när det behöver göras ändringar, underhåll, uppdateringar, för att veta hur olika funktioner fungerar och hur koden är skriven och fungerar. Det kan också	ISD

		handla om datamodeller eller liknande. Det är sådan dokumentation som vi åsyftar	
30	R3	Okej då förstår jag	
31	LH	Men hur viktigt skulle du säga att det är att uppdatera dokumentationen?	ISD+U
32	R3	Det vore jättebra om det var så. Det är svårt att gradera. Men såklart så tappar man ju tid varje gång någon ny kommer in och inte har fått den där intervjun med den som lämnar. Sen kan det va att folk bara byter projekt, då är den kollegan kvar. Så i det fallet som en person slutar, eller att kunden väljer en annan leverantör och de som lämnar inte har dokumenterat, då är det mycket värt.	ISD+U
33	LH	Ja, såklart	
34	LH	Nu är det ju kanske inte så att du uppdaterar dokumentation så ofta, men hur ofta i genomsnitt skulle du säga att man uppdaterar dokumentationen i ditt team, avdelning eller företaget i det stora?	ISD+U
35	R3	Jag kan inte svara på det	
36	LH	Nej, vi förstår. det är säkert individuellt för många	ISD+U
37	R3	Ja, och vad för typ av leverans. Förvaltning så går det antagligen bra.	ISD+U
38	LH	Tror du det skulle spela någon större skillnad på om kodändringarna är större respektive mindre? Alltså hur benägen man blir att uppdatera dokumenten efter en sådan ändring i koden?	ISD+U
39	R3	Spontant skulle jag säga att det är lättare att uppdatera småändringar för då behöver man inte skriva så mycket.	ISD+U
40	LH	Nej, precis	
41	LH	Skulle du säga att du skulle använda dig av dokumentation även om den är gammal?	ISD+U
42	R3	Ja för att förstå the bigger picture.	ISD+U
43	LH	Jaa. Men tror du ändå att det hade varit mer optimalt om den faktiskt var uppdaterad?	ISD+U
44	R3	Ja	ISD+U
45	LH	Skulle du säga att den här skillnaden mellan inte uppdaterad dokumentation och faktiskt uppdaterad dokumentation till den aktuella statusen, att det här arbetet anses värt att utföra? Det kan vara att det tidsmässigt eller kostnadmässigt. Eller tror du att man hellre använder sig av icke-uppdaterad dokumentation i så fall?	ISD+U
46	R3	Jag får gå tillbaka till tidigare svar. Om det är situationer där det är den som har utvecklat och skrivit, om någon anledning kopplas bort, försvinner. Då blir det ju definitivt viktigare än om personerna är kvar inom samma team, företag.	ISD+U+KD
47	LH	Om vi sedan kopplar det här till kunskap, det du är inne på också. På vilket sätt skulle du säga att den här icke-uppdaterade dokumentationen påverkar till delandet av kunskap? Tar det dig in på samma spår?	ISD+U+KD

48	R3	Ja, å andra sidan så tvingas ju då den som ska göra något eller ta över, att lära sig via att ta kontakt med den som har gjort det. Men inom vår bransch, konsultbransch, så är det ju tyvärr vanligt att folk byter uppdrag, projekt. Man byter oftare än för 10 – 15 år sedan. Allting är mer rörligt. Så ja, det är alltid en inkörningsperiod.	ISD+U+KD
49	LH	Ja, så är det ju	
50	LH	Hur tror du att dokumentation som inte är uppdaterad påverkar systemutveckling och underhållsarbetet eller alternativt hela systemet i slutändan?	ISD+U+KO
51	R3	Jag skulle säga såhär. Ju större system och ju djupare kärn-ITn du kommer, desto värre är det ju.	ISD+U+KO
52	MH	Ja okej	
53	R3	Om någon inte har dokumenterat en del av user interface och långt upp i den hierarkin, ja visst du kan förlora pengar och del av en dålig webb. Men det är en annan sak om det är hardcore, långt nere i värdekedjan. Då är det så att det säkert kan rulla på så länge det är vackert väder, men så fort the shit hits the fan och det blir fel i backend-systemen, då är det ju riktigt illa. Framförallt är det illa när man inte dokumenterar vilka rutiner som används. Liksom det här certifikatet måste uppdateras minst en gång i halvåret. Den här två-faktoriseringssaukoriseringen måste bli vart tredje kvartal. Den här databasen måste patchas, eller inte patchas för då kommer det här hända.	ISD+KO
54	LH	Ja, men det makes sense.	
55	MH	Då har vi en annan egenskap som handlar om kvalitet, kvaliteten på dokumentationen och hur det påverkar system och så.	ISD+K
56	MH	Vi undrar lite, om du kan svara på detta, anser motsvara hög dokumentationskvalitet?	ISD+K
57	R3	Hög dokumentationskvalitet är dels fullständighet men utgår också nog ifrån hur pass väl den som har skrivit den har skrivit den och tänkt på sin kollega som är på samma nivå. Oftast är det där man, eller jag skriver såhär för det fattar man ju. Men det beror ju på vilken nivå, vilken typ av människa det är som ska ta över det arbete. Då dokumenterar man dokumentation på en rimlig motsvarande nivå för att en nu arbiträr kollega ska kunna ta över det.	ISD+K+KD
58	MH	Vad har du för förväntningar på kvaliteten?	ISD+K
59	R3	Ja, jag skulle säga höga men det är inte det.	ISD+K
60	MH	Ja okej. Vi har några kvalitetsattribut där det ofta sviktar. Vi undrar hur du tycker om följande attribut. Vi börjar med korrekthet – översättningar, kommentarer, exempel. Hur viktigt är det att dessa är bra för att kunna använda sig utav den?	ISD+K
61	MH	Är det viktigt att det är på engelska exempelvis?	ISD+K
62	R3	När jag kodade sist var det regel att man skulle ha kommentarer på engelska redan då. Det märker jag nu när	ISD+K



		vi har informationsmejl, informationsmöten på företaget. Vi kan nästan aldrig har det på svenska längre.	
63	LH	Det blir ju så	
64	R3	Det är så mycket olika nationaliteter och vi jobbar med off-sure företag. Så engelska det vara. Korrekt ska det naturligtvis vara, då kommer man in på personer som jag själv som vill att det ska vara korrekt, det blir ju ofta långt. Det är inte heller bra, då blir det så lång text. Det här orkar utvecklare inte läsa och testat sig fram själv och använder stackoverflow. Detta gör att allt tar längre tid eftersom inlärningsperioden blir längre.	ISD+K
65	MH	Läsbarhet, bryr du dig om stavningar, grammatik?	ISD+K
66	R3	Där tycker jag att utvecklare förstår om några, vikten av inventering och läsbarhet. Man pratar mycket om läsbarkod så då borde man kunna ha krav på läsbara kommentarer och dokumentation.	ISD+K
67	MH	Ja, hur viktigt är det med strukturen på ett sådant dokument? Har n in någon standard på hur dokument ska vara utformad?	ISD+K
68	R3	På förvaltningsuppdragen använder vi dom programvaror som är vanligast, DevOps, Jira, och då blir det som styr formen.	ISD+K
69	MH	All right, spårbarhet. Hur enkelt är det att hitta vem som skrivit dokumentationen, vem som har gjort ändringar och när ändringarna har gjorts?	ISD+K
70	R3	Det är viktigt, men det slår mig att vi kommer in på hela det här med vattenfall vs scrum. Vi har gått från stora komplexa system till att vi vill jobba i Microsoft services och att team ska vara ansvariga för allt. Det blir svårt i olika organisationer. Det handlar mer och mer om organisation när jag tänker efter. Går det till stora organisationer med många människor, då finns det en roll inom ITIL, vad heter det CM – configuration management.	ISD+K
71	LH	Precis	
72	R3	Det finns en heltid där man bara ska se till att det blir ordning och reda på saker och ting. Sen har vi de autonoma rapid scrum-teamen. Jag vet knapps om de dokumenterar alls. Så ja. Jag är ledsen att jag inte svarar ja eller nej på era frågor	ISD+K
73	MH	Det gör absolut ingenting	
74	R3	Det beror så mycket på vilken kontext	ISD
75	LH	Ja, såklart.	
76	MH	Vad tycker du om fullständighet? Hur viktigt är det att faktiskt ha ett helt färdigställt dokument?	ISD+K
77	R3	Jag skulle säga att trenden inte är dokument i form av word-filer. De är i form av knowledge based-artiklar och andra uppdelningar i de här verktygen oftast. Vi hade ett uttryck för det någon gång, något i stil med så fullständigt som möjligt.	ISD+K
78	LH	Ja	

79	MH	Hur ofta innehåller intern systemdokumentation all information som krävs för att förstå, underhålla och använda ett system?	ISD+K
80	R3	Sällan. Men jag tycker samtidigt att de system, programvaror jag själv använder i och med google-generationen, så tycker jag att det har blivit väldigt mycket enklare att använda system och hitta information i det jag behöver. De verktyg jag vill använda.	ISD+K
81	LH	Vad bra	
82	MH	Så då anser du att det är viktigt att dokumentationen alltid är fullständig med tanke på att det är kostsamt? Med tanke på det du säger, du söker så mycket i andra verktyg?	ISD+K
83	R3	Ja, jag tänker att jag får också koppla upp med annat svar. Dokumentationen ska vara så pass fullständig och tillräcklig att en viss mottagare, en viss typ av mottagare ska förstå. Det är ingen idé att du och jag skriver fullständig dokumentation, om vi pratar fullständighet som min mor ska förstå – då får vi skriva ihjäl oss. Och det är klart att om en senior arkitekt utvecklare skulle skriva något som ni som studenter utan vidare arbetslivserfarenheter ska ta över, då får den personen skriva ganska mycket.	ISD+K
84	MH	Ja	
85	R3	Så ja, det är ett intressant område som ni har valt. Jag tänker väl att ja, man måste sätta vad det är för förväntningar på dokumentationen, vem är mottagare?	ISD
86	MH	Hur bidrar kvalitetsnivån till delandet av kunskap? Om kvaliteten är dåligt exempel, bidrar det till försämrad kunskapsdelning?	ISD+K+KD
87	R3	Ja	ISD+K+KD
88	MH	Hur tycker du då att intern systemdokumentation med dålig kvalitet eventuellt påverkar systemutveckling, underhållsarbetet och systemet?	ISD+K+KO
89	R3	Det påverkar definitivt underhållningsarbetet och incidenthantering.	ISD+K+KO
90	LH	Kanon, då går vi vidare till den sista egenskapen som handlar om tillgänglighet. Skulle du säga att det är enkelt att få tillgång till systemdokumentation när den behövs?	ISD+T
91	R3	Ja. Om vi håller oss till när mottagaren är en utvecklare. Men jag skulle ha jättesvårt att hitta rätt. Då måste jag kanske ha ett konto någonstans. Sen ska jag ha det här Git kanske, vart finns det.	ISD+T
92	LH	Men den som behöver ha tillgång till den, har förmodligen enkelt till det?	ISD+T
93	R3	Ja	ISD+T
94	LH	Vad bra.	
95	LH	Var och hur förvaras den här interna systemdokumentationen?	ISD+T

96	R3	Det finns huvudsakligen två ställen. Knowledge basis i confluence eller som kod i versionshanteringslösningar som vi har. Git, DevOps.	ISD+T
97	LH	Hur skulle du önska att den var förvarad?	ISD+T
98	R3	Teams använder vi. Så som den är förvarad i dag, jo det systemet är nog bra.	ISD+T
99	MH	Så bra.	
100	LH	Är det endast den senaste versionen av dokumenten som sparas eller är det även möjligt att hitta äldre versioner?	ISD+T
101	R3	Ja man kan hitta äldre också.	ISD+T
102	LH	Är det lätt att hitta vilken som är senaste, den som ska användas?	ISD+T
103	R3	Ja, det är enkelt.	ISD+T
104	LH	Så bra.	
105	LH	Om vi tar det in på kunskapsdelning. Bidrar tillgängligheten till delandet av kunskap på något sätt?	ISD+T+KD
106	R3	Japp	ISD+T+KD
107	LH	Hur då skulle du säga?	ISD+T+KD
108	R3	Vi vet vart vi ska hitta det. Det finns ett begränsat antal platser att leta på. Så det är väl själva orden som avgör.	ISD+T
109	LH	Tror du att begränsad tillgänglighet av de här dokumenten påverkar som tidigare nämnt, systemutvecklingen, underhållsarbetet eller systemet i det hela?	ISD+T+KO
110	R3	Ja om den vore begränsad så absolut.	ISD+T+KO
111	LH	Då var vi egentligen färdiga med de frågorna som vi haft men vi vill ändå fråga om det är något du skulle vilja lägga till?	
112	R3	Glöm inte ta med i ert arbete organisationstänket och mottagarnivå. Det är väldigt olika.	ISD
113	LH	Jag kan verkligen föreställa mig det. Är det okej om vi kontaktar dig med ytterligare frågor?	
114	R3	Ja, det går bra.	
115	LH	Vad snällt. Vill du även att vi skickar transkriberingen till dig efteråt?	
116	R3	Ni kan skicka den så läser jag om jag har tid.	
117	LH	Det blir jättebra.	
118	MH	Det var allt vi hade.	
119	LH	Tack så jättemycket för att du ville vara med	
120	R3	Ja, tack själva.	

## Bilaga E– Respondent 4, transkribering

R4: Respondent 4

LH: Linnéa Hansson

MH: Mathilda Hezsö

Rad	Person	Meningsenhet	Kod
1	LH	Varsågod, du kan börja.	
2	R4	Okej, intern systemdokumentation det används inom applikationsförvaltning när ett företag förvaltar ett system, en applikation.	ISD
3	MH	Mm.	
4	R4	Den bör innehålla den övergripande affärslogiken, samt hur man hanterar supportärenden bäst. Då använder man dokumentation om man behöver gå in, vad är de vanligaste felen, men även när man ska göra förändringar i systemet, så kan man ha det som ett underlag för att kunna ta fram vad är det som behöver förändras i systemet, då är detta viktigt input.	ISD
5	LH	Det låter väldigt likt det som vi också tänker oss. Så bra.	
6	MH	Vi har några introduktionsfrågor. Vi undrar vad ditt företag gör?	
7	R4	Företaget, jag jobbar främst med sjukvårdssystem, alltså inte journalsystem utan stödsystem runt omkring som heter *namn*, vilket är ett patientadministrativt stödsystem.	
8	MH	Vad är din arbetsroll/yrkestitel?	
9	R4	Jag är senior systemutvecklare tror jag det står på min titel på företaget. Men jag är allt ifrån att jag kan va en arkitekt till att jag tar hand om utveckling och support.	
10	MH	Det är dina arbetsuppgifter då?	
11	R4	Ja.	
12	LH	Kanon. Då tänker vi att vi leder in frågorna på den interna systemdokumentationen.	ISD
13	LH	Använder du dig aktivt av intern systemdokumentation i ditt vardagliga arbete?	ISD
14	R4	Det beror på vad ni menar med intern systemdokumentation. Vi är ganska slarviga här på företaget med det. Så är det ju när man kommer in i en existerande förvaltning. Då kan man inte komma in och säga såhär kan vi inte jobba, man får komma in lite ödmjukt och anpassa sig till den standard som finns. Och det jag har introducerat hos oss i alla fall för ett sätt att dokumentera är att vi har satt in en intern Wiki-sida inom avdelningen där vi försöker lägga upp dokumentation på en övergripande nivå.	ISD, ISD+T
15	LH	Okej. Skulle du säga att den här typen av dokumentation förenklar ditt arbete?	ISD
16	R4	Det förenklar, ja det gör det.	ISD
17	LH	På vilket sätt?	ISD
18	R4	...Och även för andra.	ISD

19	R4	Bland annat när nya resurser kommer in. För att inte ha någon dokumentation överhuvudtaget, utan databasen och källkoden är dokumentationen, det är jättevanligt på äldre system. Så lyfter detta sig lite, att där är lite mer kött på benet, där är lite mer kunskap på en högre nivå	ISD
20	LH	Tycker du att den här interna systemdokumentationen behöver uppnå för kriterier för att vara användbar för dig?	ISD
21	R4	När jag kommer in så är jag inte intresserad av vad källkoden gör på lägsta nivå, jag vill förstå övergripande, lite helikopterperspektiv. Alltså nyttan, vad är det den gör på en hög nivå.	ISD
22	LH	Okej. Förlitar du dig på den dokumentationen som redan finns när du ska använda dig av den?	ISD
23	R4	Det får jag ju göra, så att säga, jag har ju inget val där.	ISD
24	MH	Mm.	
25	LH	Vi har ju även riktad in oss på kunskapsdelning och hur fungerar kunskapsdelning inom ert företag?	KD
26	R4	Den kunde vara bättre kan jag bara säga. Den enda jag säger är att när jag började här så fanns det ingen dokumentation överhuvudtaget.	ISD+KD
27	LH	Okej.	
28	R4	När jag började då fick jag sätta mig vid sidan om och börja så smått, titta på koden och börja med någon mindre sak. Och sen så är det bara att kavla upp skjortärmarna och blir skitig och gräva ner sig i koden och vara hands on. Förstår jag inte får jag fråga en kollega.	KD
29	LH	Skulle du säga att det är så kunskapsdelning fungerar hos er. Att man frågar kollegor?	KD
30	R4	Ja, tyvärr. Det är så tyvärr nog. Jag har ändå jobbat sen -88 och jag kan säga dig att alla dem ställena jag varit på har jag aldrig sett någon bra systemdokumentation.	KD+ISD
31	MH	Nä okej.	
32	LH	Men personligen, delar du helst kunskap genom muntliga dokumentationer eller dokumentation?	KD+ISD
33	R4	Jag försöker dokumentera på en övergripande nivå, det ska liksom inte vara tusen sidor utan jag kanske försöker få ner på ett tjugo-, trettiotal sidor på en övergripande nivå, alltså vad gör systemet som jag tycker är viktigast.	ISD+KD
34	LH	Ja.	
35	R4	Och gärna med någon arkitekturbild. För jag tror att en systemskiss säger så mycket mer än när man bara skriver en massa löptext, jag tror man ska försöka ha lite serietänk, typ som kalle anka, som en serietidning, mycket bilder och skisser och inte en massa löptext.	ISD+K
36	LH	Det blir tydligare då.	
37	R4	Det är nog lättare att komma in i det.	ISD
38	LH	Hur hanterar ni det när en kunskapsbesittande person närma sig lämna företaget?	KD

39	R4	Då ber vi den försöka dokumentera på Wiki-sidan som vi har fått upp.	ISD+KD
40	LH	Okej.	
41	LH	Är det så att du arbetar med underhållsarbete av olika system?	
42	R4	Ja.	
43	LH	För att hitta den kunskap som krävs för att kunna utföra arbete hur går du till väga då?	KD
44	R4	Reverse engineering.	KD
45	LH	Skulle du säga att det finns dokumentation som du kan använda dig av?	ISD+KD
46	R4	Nej.	ISD+KD
47	LH	Vi har också fokuserat in oss på tre egenskaper som det ofta är problem med inom intern systemdokumentation. En av dem handlar om uppdatering. Så då vill jag fråga hur viktigt det är för dig att uppdatera den dokumentation som ni använder dig av?	ISD+U
48	R4	Det är jätteviktigt annars kan jag kasta den direkt.	ISD+U
49	LH	Varför tänker du det?	ISD+U
50	R4	Ja, men om jag ändrar i systemet och dokumentationen inte matchar då har du absolut ingen nytta av systemdokumentationen på sikt. Det blir för stor skillnad mellan verkligheten och kartan, eller hur man nu säger det.	ISD+U
51	LH	Hur ofta händer det att du uppdaterar den interna systemdokumentationen?	ISD+U
52	R4	Jag försöker göra det där jag har tagit fram dokumentationen och jag känner till den. Återigen på en övergripande nivå, jag går in och till exempel om vi lägger till någon ny kolumn i en databastabell då försöker jag beskriva vad det fältet gör i den tabellen. Vi försöker ha det så, det är en del av den interna dokumentationen. Jag tycker det är jätteviktigt så att man förstår informationen i databasen, samt är där någon övergripande text om systemet.	ISD+U
53	LH	Skulle du säga att du dokumenterar annorlunda beroende på storleken av ändringen i koden? Om det är en mindre eller större ändring?	ISD+U
54	R4	Ja, det tycker jag.	ISD+U
55	LH	Hur då?	ISD+U
56	R4	Det är tack vare att vi har en övergripande dokumentation, vi går inte ner och dokumenterar varje enskild kod-rad utan jag vill ju hålla på en hög nivå, så är det bara så att man gör en extra parameter, som bara är en kosmetisk sak och inte ändrar systemets huvudfunktion, då bryr jag mig inte om att dokumentera. Det är mer när man ändrar en viktig del i systemet som jag tycker det är viktigt att ändra det.	ISD+U
57	LH	Mm.	
58	R4	Men allting beror på den dokumentation man har från början, för det är jättesvårt att dokumentera i efterhand. Säg systemet byggdes för tio år sedan, det är jättedyrt att dokumentera, det	ISD+U

		finns inte dem pengarna. Det ska göras när utvecklingen görs, efter det så finns det inte ekonomiska möjligheter.	
59	LH	Mm. Men då blir det helt enkelt att dem större, viktigare ändringar är det du prioriterar?	ISD+U
60	R4	Ja, precis.	ISD+U
61	LH	När du stöter på en dokumentation som är lite gammal och inte uppdaterad skulle du använda dig av den?	ISD+U
62	R4	Det är som jag sa, jag har inget val. För att komma in i det så är det ju bättre än ingenting.	ISD+U
63	LH	På vilket sätt skulle du använda dig av den då?	ISD+U
64	R4	Om det är ett helt nytt system jag kommer in i så får jag kanske förståelse för systemet. Vad är systemet och vad vill jag uppnå med systemet, så att jag får en förståelse, för om jag inte förstår det så kan jag ju inte gå in och ändra i koden.	ISD+U
65	MH	Men du anser ändå att ju mer uppdaterad ju bättre är det?	ISD+U
66	R4	Jaja, så är det.	ISD+U
67	LH	Mm. Skulle du säga att skillnaden är så pass stor och värd att det är värt att lägga ner den här tiden på det här uppdateringsarbetet?	ISD+U
68	R4	Det är värt det. Tänk dig om du går in och ändrar bara någon gång om året så kommer du tillbaka och jobbar med 10 andra system, man blandar ihop saker med åren om man inte gör ändringar så mycket.	ISD+U
69	LH	Det låter väldigt rimligt.	
70	LH	Hur tycker du att den här icke-uppdaterade dokumentationen påverkar delandet av kunskap?	ISD+U+ KD
71	R4	Det blir svårare för nya kollegor att komma in. Det tar ju längre tid och kostar desto mer pengar.	ISD+U+ KD
72	LH	Yes. Hur tycker du att den icke-uppdaterade dokumentationen kan påverka själva systemutvecklingen, underhållsarbetet eller systemen?	ISD+U+ KO
73	R4	Jättemycket. Till sist vågar du inte gå in och ändra i systemet för du vet inte vad det är som sker bakom skynket, hur det är tänkt att fungera.	ISD+U+ KO
74	LH	Ja, precis.	
75	R4	Jag har sett sådan källkod där vi inte vågar gå in och ändrar och då har du tappat kontrollen på systemet.	ISD+U+ KO
76	LH	Blir det ofta fel skulle du säga?	ISD+U+ KO
77	R4	Oja, har du inte bra förståelse och bra dokumentation för att gå in och ändra så ökar detta sannolikheten för fel. Du tänker inte på alla testfall. Så testfall är jätteviktigt att ha en systemdokumentation också.	ISD+U+ KO
78	LH	Kanon.	
79	MH	Då har vi vårt andra attribut som är kvalitet, alltså kvalitet av systemdokumentation. Då undrar vi vad du anser motsvarar hög dokumentationskvalitet?	ISD+K
80	R4	Det är ju, jag kom lite in på det, det är ju faktiskt att ha bra testdata och bra testfall. Även att ha en arkitekturbild, som	ISD+K

		visar systemsamband, flöde och hur det är tänkt att fungera på en hög nivå.	
81	MH	Då skulle du säga att detta är dem förväntningar du har på er systemdokumentation?	ISD+K
82	R4	Ja, ja.	ISD+K
83	MH	Tycker du att dess förväntningarna uppfylls i den systemdokumentation ni använder i företaget?	ISD+K
84	R4	Nej, vi har varken testfall eller arkitekturbild för alla system som vi använder. Det tar tid att ta fram alla dessa saker och någon måste betala för det. Det är svårt att betala för något som har körts i flera år och som skulle ha gjorts i samband med utvecklingsarbetet.	ISD+K
85	MH	Och då har vi hittat en par kvalitetsattribut, som vi gärna hade velat veta hur viktiga du tycker dem är i din systemdokumentation. Till exempel så är den första korrekthet. Hur viktig är det att dokumenten är översatta rätt eller att kommentarer etcetera är korrekt?	ISD+K
86	R4	Det är ju jätteviktigt, ju mer det är rätt, desto snabbare kommer du in i det och ju villigare blir det att göra underhållet.	ISD+K
87	LH	Okej.	
88	MH	Läsbarhet. Hur viktigt är det med rätt stavning, grammatik och språkuttryck?	ISD+K
89	R4	Ja, stavningen inte så mycket, men mer namnkonventioner.	ISD+K
90	MH	Ja, att det ska vara liksom rätt och på engelska då eller hur tänker du?	ISD+K
91	R4	Ja, men att man även benämner olika saker i ett system så att man inte olika kallar det för olika saker. Man har samma namn överallt, det är nog det viktigaste för läsbarheten	ISD+K
92	MH	Struktur, hur viktigt är det att det finns en standard för hur dokumentationen ska vara utformad?	ISD+K
93	R4	Det är också jätteviktig, ju bättre standard du har då vet du var du ska gå in och söka information och uppdatera information.	ISD+K
94	MH	Mm. Spårbarhet. Hur viktigt är det att se vem som skrivit dokumentet och gjort ändringar?	ISD+K
95	R4	Det är också viktigt att förvalta om det kommer in något fel så brukar den som skrivit det få rätta det för den personen är bäst lämpad. Och det har vi ju när man checkar källkod i något versionshanteringssystem. Det är ju där du har spårbarheten.	ISD+K
96	MH	Fullständighet. Det finns ju systemdokument som inte är färdig skрина eller som kanske inte innehåller allt nödvändigt innehåll. Kan du använda dig av den ändå?	ISD+K
97	R4	Det är som jag sa innan, jag är inte intresserad att gå in och beskriva i ett visst program i koden, man kan ju faktiskt läsa källkoden, man kan använda den ändå. Men jag vill ha en översiktbeskrivning, det är viktigaste än att du går in och beskriver enskilda metoder.	ISD+K
98	MH	Skulle du säga att du tar hänsyn till dessa kvalitetsattribut när du själv skriver systemdokumentation?	ISD+K



99	R4	Jag försöker, man anpassar sig till den nivå som finns när man kommer till ett äldre system. För man kan inte trampa på som en elefant, såhär ska det vara, då lär man inte komma så långt.	ISD+K
100	MH	Mm. Hur ofta innehåller systemdokumentation all den information som du behöver för att förstå, underhålla och använda ett system?	ISD+K
101	R4	Jag kan säga att det finns förbättringspotential. Ofta finns inte allt det nödvändiga utan man måste kavla upp skjortärmarna och grota ner sig innan man kan börja på riktigt.	ISD+K
102	MH	Okej.	
103	R4	Anser du det viktigt att dokumentationen alltid är fullständig med tanke på att det är så pass kostsamt?	ISD+K
104	R4	Det är en balansakt mellan att ta fram dokumentationen kontra när du väl behöver göra någonting, så det gäller att hitta någon balans där. För du kan inte lägga ner hur mycket tid som helst för att göra den perfekta dokumentationen. Men visst, ju bättre du gör det desto mer tjänar du på det i framtiden.	ISD+K
105	MH	Ja, precis.	ISD+K
106	MH	Mm, hur tycker då att en hög eller låg kvalitetsnivå bidrar till delning av kunskap inom organisationen?	ISD+K+ KD
107	R4	Hur menar du då?	ISD+K+ KD
108	MH	Jag tänker om det är dålig kvalitet på ett dokument använder ni det ändå...	ISD+K+ KD
109	LH	Sätter det någon begränsning i hur väl kunskapen sprids inom företaget eller inte?	ISD+K+ KD
110	R4	Som sagt, ingen dokumentation, då vill du ha dokumentation. Finns det ingenting så är en dålig dokumentation bättre för att sprida kunskap.	ISD+K+ KD
111	MH	Hur tycker du då att dålig kvalitet påverkar utveckling, underhållsarbete och själva systemet?	ISD+K+ KO
112	R4	Allting tar mycket längre tid, plus att du ökar sannolikheten för fel.	ISD+K KO
113	LH	Kanon.	
114	LH	Då har vi den sista egenskapen kvar som handlar om tillgänglighet. Är det lätt att få tillgång till den interna systemdokumentationen när du behöver använda den?	ISD+T
115	R4	Ja, det tycker jag tack vare att vi inte har den i massa Word dokument utan att vi lagt den i en Wiki.	ISD+T
116	LH	Skulle precis säga det, var och hur är systemdokumentationen förvarad, men den är på Wiki?	ISD+T
117	R4	Ja, på Wiki och så har vi versionshantering också, där man kan lägga lite.	ISD+T
118	LH	Mm, så det är som en typ av dokumenthanteringssystem?	ISD+T
119	R4	Ja, jag gillar Wiki för du har inte en massa Office dokument, du skriver det på sidor, det är lättare att läsa, det försvinner inte och man kan söka på informationen.	ISD+T
120	LH	Kanon.	

121	R4	Hur önskar du i en perfekt värld att den va förvarad för att bringa mest nytta?	ISD+T
122	LH	Jag skulle vilja följa en modell som heter TOGAF.	ISD+T
123	R4	Okej, vill du förklara det.	ISD+T
124	R4	Det är ett Enterprise... Det är en metodik för att hålla ihop Enterprise architecture, på en hög nivå, där du både tittar på informationsmängden, vilken infrastruktur du har, vilka roller i verksamheten som du använder dig som administratör, är det externa användare etcetera Plus att den håller reda på verksamheten inom företaget, liksom är det HR-avdelning, plus inköpsavdelning, plus utvecklingsavdelning.	ISD, ISD+T, ISD+K
125	LH	Jaha, vad intressant. Det har jag inte hört om innan. Låter kanon.	
126	R4	Ja, det är det. Jag har satt *ohörbart* på TOGAF och har man den dokumentationen, för då tar man den övergripande nivån och där finns ett sätt hur man kan modellerna detta med alla dem parametrarna jag sa, med informationsmängd, infrastruktur, roller och organisationer i ett set.	ISD+T
127	MH	Ja, okej.	
128	R4	Plus att bygga upp ett flöde. Så när du vill göra en förändring så har du ju as is, som det ser ut idag, sen modellerar du ditt slutmål, hur vill du att det ska se på alla dessa fyra parametrarna. Sen gör du en GAPanalys och då får du ut underlag om vilka arbetspaket man behöver göra för att komma dit. Så har man det som en input för utvecklingsarbetet.	ISD+T, ISD+K
129	LH	Hm, okej spännande. Det får vi kika på.	
130	LH	Men i nuläget sparas endast den senaste versionen av dokumentet eller kan man även få tillgång till de äldre versionerna?	ISD+T
131	R4	På Wiki är det versionshantering på det automatiskt.	ISD+T
132	LH	Är det lätt att veta vilken som är den senaste/mest relevanta?	ISD+T
133	R4	Ja, du ser bara det senaste, det är ju egentligen det som är av intresse.	ISD+T
134	LH	Okej, så du ser bara det senaste. Ja, okej.	ISD+T
135	R4	Ja, det andra är ju bara historia, det kan ju vara intressant för att se varför gjorde vi så, då kan man gå bak, men annars har man ingen anledning att gå bakåt.	ISD+T
136	MH	Okej, ser man även vem det är som skrivit den där?	ISD+T
137	R4	Ja.	ISD+T
138	MH	Okej, kanon.	
139	LH	Hur skulle du säga att tillgängligheten bidrar till delning av kunskap?	ISD+T+ KD
140	R4	Det underlättar väsentligt. Jag införde ju detta för fem år sedan, jag fick en budget för att jag drev den utveckling. Och i börja sa folk vad ska vi med det till. Men nu i efterhand inser dem nyttan.	ISD+T+ KD
141	LH	Är det så att det blir lättare att ta del av kunskap ju lättare det är att få tag på dokumenten?	ISD+T+ KD

142	R4	Jaja, då har du någonting du själv kan söka på, du behöver inte ringa eller prata med personen som kan systemet, utan du kan gräva lite själv också. Så, visst hjälper detta att sprida kunskap. Kompetensöverföring.	ISD+T+ KD
143	LH	Kanon ju.	
144	LH	Okej, men hur tycker du då att begränsad tillgänglighet inom intern systemdokumentation påverkar systemutveckling, underhållsarbetet eller systemet?	ISD+T+ KO
145	R4	Det är betydligt, det är mycket negativt.	ISD+T+ KO
146	LH	På vilket sätt då?	ISD+T+ KO
147	R4	Ja, det är som vi sa innan, det tar mycket längre tid för att du måste skapa dig en förståelse hur ett äldre system fungerar. Du ökar sannolikheter för fel när du inte får tag i dokument och då inte förstår felet helt ut när du ska sätta in nya saker. Och ett sätt att minimera dem riskerna är att man behöver intern systemdokumentation för testfall och testning. Då kan man återanvända när man gjort en ändring, så när man utvecklar för första gången så har man testfall och testdata och där fungerar det. Sen gör man en ändring. Sen ska man göra om dem här gamla testerna för att säkerställa att gammal funktionalitet fungerar som innan.	ISD+T+ KO
148	LH	Okej, kanon.	
149	MH	Mm.	
150	R4	Hoppas ni har blivit klokare.	
151	LH	Ja, det tycker jag absolut. Det var slutet på våra frågor, men vi vill fråga om det finns något du vil tillägga som du kommit på under intervjuens gång.	
152	R4	Nä, jag har nog nämnt dem flesta punkterna jag tycker är viktiga. Det är det här med att följa metodik och dokumentera det. TOGAF, tillexempel och att även implementera testfall med testdata.	
153	LH	Yes, amen kanon.	
154	MH	Vi undrar också om det är så att det är okej att vi kontaktar dig om vi har några fler frågor?	
155	R4	Inga problem.	
156	LH	Tack.	
157	LH	Är det också så att du vill att vi skickar transkriberingen till dig så du kan kolla igenom den?	
158	R4	Ja, ni kan skicka den. Om ni tycker det hjälper er så kan jag kolla igenom den.	
159	LH	Yes, vi skickar den. Du kan även få ta del av uppsatsen sen när den är helt klar om du hade velat det.	
160	R4	Ja, det hade varit kul att se.	
161	LH	Ja, men då tackar vi för oss. Tack så jättemycket för att du ställde upp.	
162	MH	Ja, tack.	

## Bilaga F – Respondent 5, transkribering

R5: Respondent 5

MH: Mathilda Hezsö

LH: Linnéa Hansson

Rad	Person	Meningsenhet	Kod
1	MH	Vad gör ditt företag?	
2	R5	*Företaget* gör övervakningskameror, men min roll i det är att jag sitter på IT-avdelningen, så jag jobbar med interna system.	
3	MH	Och vad är din yrkestitel?	
4	R5	Min yrkestitel är att jag är Product Owner.	
5	MH	Okej, skulle du kunna beskriva dina arbetsuppgifter lite övergripande?	
6	R5	Jag äger en roadmap och 14 resurser som jag sen ger ut arbete till för mina system som jag äger. Jag är systemägare och äger min roadmap för dem system jag äger.	
7	LH	Wow vad häftigt.	
8	LH	Men då tar vi oss in lite på systemdokumentation, är det någonting du aktivt använder i ditt vardagliga arbete?	ISD
9	R5	Ja, dagligt och dagligt, men jag använder det ofta.	ISD
10	LH	Okej. Tycker du att det förenklar ditt arbete?	ISD
11	R5	Ja, men det gör det. Jag kan skicka vidare det till intressenter, konsumenter eller producenter, folk som producerar information och folk som konsumerar min information, mina system. På så sätt slipper jag förklara allting så brukar jag skicka vidare det eller ha en to-do lista för att gå väga.	ISD
12	MH	Super.	
13	LH	Vilka kriterier ska den här interna systemdokumentationen uppnå för att den ska vara användbar för dig?	ISD
14	R5	Den ska kunna vara lätt att uppdatera, den ska vara lättillgänglig och den ska vara lättläst.	ISD+U, ISD+T, ISD+K
15	MH	Okej.	
16	LH	På vilket sätt då, eller det tar vi oss in på senare i och för sig. Hur väl förlitar du dig på den systemdokumentationen som du använder och som redan är skriven?	ISD
17	R5	Av andra eller som jag skrivit själv?	ISD
18	LH	Framförallt då av andra.	ISD
19	R5	Den dokumentation som andra skrivit brukar jag följa. Överallt inom IT-världen så rör sig saker så himla snabbt, så jag ser det mer som en riktlinje än att jag ser det som någonting som absolut måste vara på ett speciellt sätt. Jag kan läsa dem och dra lite egna slutsatser, eller jag kan läsa det och få ett hum om hur jag ska göra saker, men jag vet om att det säkert innehåller fel ibland. Jag är lätt skeptisk ibland.	ISD
20	LH	Ja, men det blir nog lätt så när det förändras så fort.	ISD
21	R5	Ja det går så himla snabbt, så man kan räkna med att inte alla hinner uppdatera sin egen systemdokumentation	ISD+U

22	LH	Men hur fungerar kunskapsdelning på ert företag?	KD
23	R5	Vi har haft sådana.... Innan Corona så var det att alla skulle vara på jobbet hela tiden, vi har det som standard, vi arbetar väldigt lite hemifrån, förutom nu då. Då har mycket av överlämning och informationsspridning det har gått från person till person istället för genom dokumentation. Dokumentation har mer varit ett sätt när det ska vara folk man inte träffar så ofta eller om det är någonting man måste återupprepa många gånger, då brukar vi göra dokumentation.	KD, ISD+KD
24	LH	Okej, men om ni har någon person inom företaget som sitter på mycket kunskap, hur gör ni när den börjar närma sig lämna företaget?	KD
25	R5	Ja, oftast så brukar vi istället för dokumentation, så låter vi den personen hoppa in på andra uppgifter, som kanske inte är lika viktiga och sen så ser vi var det smäller.	KD
26	LH	Oj.	
27	R5	Ja, så man ser var den personen har varit alldeles för viktig och var det har varit svårt. Men vi har ju såklart dokumentation, men vissa utvecklare kanske inte riktigt brinner för dokumentation.	KD, ISD
28	LH	Nä, det har vi förstått.	
29	R5	Det är inte så lätt att hålla den uppdaterad och så handlar det också om att när det väl finns människor som har jobbat väldigt, väldigt länge på *företaget* så kanske de sitter på så himla mycket information att folk ändå inte kommer hitta den information de letar efter i systemdokumentation. Man får ha en avvägning där. Det brukar vara lättare om personer får sätta sig in i problemen själva, så att det har ett hum om var de ska börja leta. Vår kod är till stor del systemdokumentation också.	ISD, KD
30	LH	Jaa.	
31	R5	Så för oss är vår koddokumentation också.	ISD
32	LH	Jag förstår det som att du inte håller på med så mycket kodning och underhållsarbete just nu, men är det någonting som du gjort mycket innan?	KD
33	R5	Ja det är det. Jag är civilingenjör i datateknik, och har varit utvecklare i 10 år.	
34	LH	Mm, men hur gjorde du då för att hitta den kunskap som krävdes för att utföra det här underhållsarbete ordentligt?	KD
35	R5	Jag letade i kodbasen.	KD
36	LH	Okej.	
37	R5	Det var oftast det som var enklast. Jag har jobbat på så många olika ställen. Och *företaget* säger att det inte är så bra på systemdokumentation, *annat företag* dem sa att dem var väldigt bra på dokumentation, men man drunknade i all systemdokumentation på det företaget.	KD, ISD, ISD+KD
38	LH	Oj, okej.	
39	R5	Och på *företaget* är det snarare att det finns lite dokumentation, men det finns liksom de viktigaste sakerna som man har svårt att ta ut ifrån koden, och sen när jag har jobbat på andra it-ställen, på *annat företag*, *annat företag*,	ISD, ISD+KD

		och när jag jobbat som konsult innan jag började på *företaget*, då fanns det ingen systemdokumentation utan då fick man kolla i koden.	
40	LH	Nej, precis, vi har förstår det på lite andra utvecklare, dem säger lite likadant att det har kommit lite den senare tiden.	
41	R5	Mm, desto större företag desto mer dokumentation, så kan man sammanfatta det.	ISD
42	MH	Ja okej.	
43	LH	Vi har ju tagit fram tre egenskaper eller problemområden inom intern systemdokumentation som det ofta förekommer lite svårigheter kring. Så vi tänkte ställa några frågor om dem. Den första är uppdatering.	ISD+U
44	LH	Hur viktigt tycker du det är att man uppdaterar intern systemdokumentation?	ISD+U
45	R5	Det tycker jag är superviktig eftersom om man skriver ner någonting och inte kollar igenom det då och då, då kan man inte lita på det. Men hur ofta det behöver ske... Jag kan ge ett exempel på hur vi brukar göra. Jag tycker att dokumentation är viktigt men det måste vara på ett pragmatiskt sätt. Det får vara russinen i kakan man dokumenterar, alltså det som är absolut viktigast. Vi har sådana här dokumentationsdagar i mitt team, där vi sätter oss och dokumenterar. Vi sätter oss ner och diskuterar vad som saknas och vilka som är de viktigaste delarna och sen sätter vi oss ner och dokumenterar. Vi går även igenom den gamla dokumentationen. Det har vi ungefär en gång i halvåret. Sen tillhör det ju i uppgiften att när man skriver en ny funktionalitet så ska man fundera om det behövs dokumentation för det.	ISD+U
46	MH	Mm.	
47	LH	Av vilken anledning dokumenterar ni?	ISD
48	R5	Vi dokumenterar ofta när det är klurig funktionalitet, som man ofta glömmer och som kräver att man måste gå in i koden och lägga en halvdag för att förstå.	ISD
49	LH	Ja, okej.	
50	R5	Jag brukar fråga mina utvecklare, om och om igen och vid olika tillfällen och vi fortfarande måste gå in i koden för att dubbelkolla då är det bara lättare att skriva ner det istället.	ISD
51	LH	Ja, det låter rimligt.	
52	LH	Skulle du säga att dokumentationsarbetet ter sig annorlunda beroende på storlek av ändring i koden?	ISD+U
53	R5	Nej. Det är snarare svårigheten av funktionaliteten.	ISD+U
54	LH	Så det spelar ingen roll om det är större eller mindre kodändringar?	ISD0U
55	R5	Nä, det tycker jag inte. Det kan vara jätte-straightforward ändringar men som påverkar på massa olika ställen.	ISD+U
56	LH	Okej, förstår.	
57	LH	Skulle du använda dig av en dokumentation som är gammal och som inte är uppdaterad till statusen i koden?	'ISD+U

58	R5	Jag skulle nog då hellre fråga en utvecklare som arbetar med det.	ISD+U
59	LH	Varför?	ISD+U
60	R5	För att jag skulle vara för orolig att det är för stora skillnader mellan hur det fungerar mot hur det är dokumenterat. Att det är någon som har skrivit om det för mycket, men det händer väldigt sällan. Där finns ju alltid ett grovt korn av sanning.	ISD+U
61	LH	Ja, det är klart.	
62	LH	Så du skulle ändå säga att det finns en del användbarhet även om den är lite gammal?	ISD+U
63	R5	Ja, det är väldigt sällan den inte ger någonting alls.	ISD+U
64	LH	Så bra.	
65	LH	Hur skulle du säga att en icke-uppdaterad dokumentation påverkar delning av kunskap?	ISD+U+ KD
66	R5	Svårare att dela ut det till andra aktörer, för då måste jag stå för informationen som står där.	ISD+U+ KD
67	LH	Yes, och hur tycker du den icke-uppdaterade dokumentationen påverkar utvecklingen, underhållsarbete och systemet i slutändan?	ISD+U+ KO
68	R5	Att vi drar slutsatser som är felaktiga.	ISD+U+ KO
69	LH	Vad skulle det kunna leda till?	ISD+U+ KO
70	R5	Det skulle kunna leda till att vi tar fel beslut och att vi tar fel funktionsändringar, att vi utgår från att det fungerar på ett sätt men så är det på ett annat sätt. Istället för att kolla i koden så har man läst dokumentationen och så utgår man för att så det är. För att koden är svår just där så utgår man från dokumentationen istället. Det kan vara ett problem.	ISD+U+ KO
71	LH	Ja, kan det leda till något speciellt i sig?	ISD+U+ KO
72	R5	Det kan leda till stora problem för våra konsumenter och våra system. Alltså så att det blir kostnader och att andra system som är beroende av vårt system inte fungerar. Det skulle kunna bli svåra problem.	ISD+U+ KO
73	LH	Yes, tack.	
74	MH	Då tänkte vi gå vidare till nästa egenskap som handlar om kvalitet. Vi undrar lite vad du anser motsvararhög dokumentationskvalitet?	ISD+K
75	R5	Vad som jag tycker är hög dokumentationskvalitet är att den är lätt att ta till sig, lätt att förstå och tydlig.	ISD+K
76	LH	Mm.	
77	R5	Men om jag får säga andra saker så tycker jag att bilder är så himla stor skillnad. Det är en bisak egentligen, men att göra den så tydlig som möjligt.	ISD+K
78	MH	Ja, och då hjälper ju bilder liksom.	ISD+K
79	R5	Exakt.	ISD+K
80	MH	Då skulle du säga att detta är dina förväntningar på kvaliteten?	ISD+K

81	R5	Ja, men det behöver inte vara i detalj allting men den måste vara lättillgänglig, det är mycket viktigare. Alltså att man kan hänvisa till detaljer på andra ställen men att den övergripande bilden är supertydlig och att man är tydlig med vilka detaljer man utelämnat.	ISD+K
82	MH	Okej. Tycker du att dina förväntningar uppfylls i den systemdokumentation du använder?	ISD+K
83	R5	Till stor del.	ISD+K
84	MH	Om den inte gör det vad är det då du tänker?	ISD+K
85	R5	Om vi tänker mitt team så är det ju jag som har ansvaret och ser till att det blir tydligt nog. När det gäller för andra så är det ju högt och låg, vissa gillar textstycken, som man får sitta och läsa i två timmar och sen förstå och det tycker inte jag är lättillgängligt. Det är ju upp till hur man är som person.	ISD+K
86	MH	Ja det har vi också förstått, att mycket av detta handlar om den enskilda individen, vad man tycker är bäst.	ISD+K
87	R5	Precis.	ISD+K
88	MH	Vi har några kvalitetsattribut, som vi hittat i litteratur som helt enkelt påverkar kvaliteten på dokumentationen och vi undrar hur viktiga du tycker dessa är.	ISD+K
89	MH	En handlar om korrekthet, alltså typ översättningar och kommentarer. Hur viktigt är det till exempel att allting följer engelska?	ISD+K
90	R5	För mig inte det viktigaste, men vi har ju till exempel engelsktalande i vårt team, så vi skriver allting på engelska. Vi har även någon person i vårt team som har svårt för stavning, och det fungerar ju jättebra i alla fall, så stavning och sådant är mindre problem.	ISD+K
91	MH	Ja, okej. Det var nästa fråga, läsbarhet. Det hade att göra med stavning och grammatik, men då tycker inte du att det spelar så stor roll?	ISD+K
92	R5	Alltså om det stör mig så ska jag kunna gå in och ändra men det är inget problem för hen som har problem med stavningen att skriva. Det är viktigare att han får ner det och sen får vi hjälpa till med stavning.	ISD+K
93	MH	Ja, det påverkar kanske inte egentligen så mycket användbarheten av den systemdokumentationen.	ISD+K
94	R5	Nej, dyslexi är ju liksom inget problem.	ISD+K
95	MH	Nä, precis.	
96	MH	Sen har vi struktur. Har ni någon form av standard för hur systemdokumentation ska vara utformad?	ISD+K
97	R5	Ja, det tycker jag är viktigt att man ska ha en övergripande bild. En viss struktur tycker jag är väldigt värdefull.	ISD+K
98	MH	Har ni någon struktur ni följer?	ISD+K
99	R5	Vi har en systemdokumentation template som vi följer.	ISD+K
101	MH	Okej.	
102	MH	Spårbarhet, hur viktigt är det att se vem som gjort ändringarna, är det gjordes et cetera.	ISD+K



103	R5	Jag tycker det är viktigt att man ska kunna se det men det är väldigt sällan jag faktiskt använder det.	ISD+K
104	MH	Ja, okej. Och sen fullständighet. När vi läst litteratur så har vi hittat ganska mycket om att utvecklare använder sig av ofullständiga dokument bara för att hitta det lilla nödvändiga innehållet. Men hur viktigt skulle du säga det r att det är helt färdigskrivet?	ISD+K
105	R5	Det tycker jag inte är så viktigt heller. Jag tycker att man ska få förståelse, hum och överblicken. Sen kan man gräva, då får man lättare koll på var det är man ska börja gräva någonstans.	ISD+K
106	MH	Okej.	
107	MH	Skulle du säga att du tar hänsyn till dessa attribut när du skriver systemdokumentation?	ISD+K
108	R5	Ja, vissa av dem tycker jag är viktigt så som strukturen och användarvänligheten av den.	ISD+K
109	MH	Okej. Hur ofta tycker du att systemdokumentation innehåller all den nödvändiga information som behövs för att förstå, underhålla och använda systemet?	ISD+K
110	R5	I mina egna system så innehåller dem totalt all information jag behöver, annars så kräver jag av mina kollegor att dem ska lägga till information. Så jag tycker jag har fullständig information. Men det är ofta jag kommer på något nytt som behövs dokumentera. För nu har jag frågat en kollega 10 gånger om detta, nu måste vi skriva ner det istället. Då kan jag bara skicka ut det nästa gång till dem som frågar om det. Jag upptäcker ofta funktionalitet som skulle behöva dokumenteras.	ISD+K
111	MH	Men det är fullständigt och så handlar det snarare om tillägg?	ISD+K
112	R5	Ja.	ISD+K
113	MH	Okej.	
114	MH	Skulle du säga att det är lönsamt att använda dokumentation som inte då kanske innehåller all den nödvändiga informationen?	ISD+K
115	R5	Vi har valt bort att använda viss dokumentation när den inte är fullständig, när den ska skickas ut till helt främmande människor som man inte känner. Så när den inte är fullständig när det är folk man inte känner och som man ska skicka ut till då brukar vi inte skicka ut den. Men det handlar lite om hur nära relation och hur nära man sitter.	ISD+K
116	MH	Mm, okej.	
117	MH	Hur tycker du då att en hög eller låg kvalitetsnivå bidrar till kunskapsdelning?	ISD+K+KD
118	R5	En låg kvalitetsnivå, jag tycker att den bidrar till viss kunskapsspridning fast kvaliteten är låg eftersom det är någon som har tyckt att den varit viktigt nog att skriva ner det ens, då vet man att här är nog någonting knasigt, det måste vi vara försiktiga med.	ISD+K+KD
119	MH	Mm.	
120	R5	Men det har lite att göra med vem som har skrivit det. Är det någon som har jobbat i vårt team 2 år plus, om det kommit en längre kvalitet av deras dokumentation då vet jag att det är	ISD+K+KD

		något viktig. Här har dem börjar på någon dokumentation för att det var något klurigt. Så det är mer en flagga till mig att jag ska titta på det lite noggrannare.	
121	MH	Ja.	
122	R5	Men är det någon helt ny som skrivit den, då är det en helt annan sak. Då kanske det är någon ny som kanske tror att oj det här var svårt. Då brukar jag kolla vem det är som skrivit den.	ISD+K+KD
123	MH	Okej.	
124	MH	Hur tycker du då att en låg kvalitetsnivå påverkar underhållsarbetet, utvecklingen och själva systemet?	ISD+K+KO
125	R5	Att vi drar dåliga slutsatser.	ISD+K+KO
126	MH	Okej, så lite samma som på föregående konsekvensfråga?	ISD+K+KO
127	R5	Ja.	
128	LH	Okej, då går vi över till sista egenskapen som handlar om tillgänglighet, vilket du också rörde vid lite innan. Men skulle du säga att det är lätt att få tillgång till den interna systemdokumentationen när du behöver den?	ISD+T
129	R5	Ja och nej. Vår systemdokumentation är väldigt lättillgänglig, men R&D, vi sitter ju inte på R&D, utan vi sitter på IT-organisation och R&D har låst ner det för att det inte ska vara tillgängligt överallt. Så det har lite med att göra med vilken organisation man tillhör.	ISD+T
130	LH	Okej.	
131	R5	R&D har ju nya saker på gång, dem kan ju inte visa upp det, för sales kan ju ibland bli lite överglada och vilja sälja det lite tidigt. Så dem måste stänga ner det, vilket dem också gjort för IT, och då blir det ju lite svårt att hitta viss IT.	ISD+T
132	LH	Hur går du till väga då?	ISD+T
133	R5	Jag hör av mig till dem som har siten och dem intressenter som hör till. Och ibland får jag tillgång och får ett utdrag.	ISD+T
134	LH	Okej.	
135	LH	Hur brukar er interna systemdokumentation förvaras?	ISD+T
136	R5	Förut hade vi en SharePoint installation och den har vi fortfarande kvar, men nu har vi Confluens.	ISD+T
137	LH	Vad är det?	ISD+T
138	R5	Confluence är *namn på företag*, har ni hört talats om Jira?	ISD+T
139	R5	Jira är ägt av samma företag, det är också *namn på företag*.	ISD+T
140	LH	Jaha, okej.	
141	LH	Men är det någon typ av dokumenthanteringssystem?	ISD+T
142	R5	Ja, det är det och det är väldigt lätt att använda. Det är som ett Wiki, men det funkar lite, lite bättre. Så vi har massa olika Wiki också, men Confluence är den som används över hela företaget.	ISD+T
143	LH	Okej.	
144	LH	Men i en drömvärld, hur önskar du att systemdokumentationen var förvarad?	ISD+T

145	R5	Jag gillar ju Confluence och jag gillar Jira så jag tycker det fungerar bra.	ISD+T
146	LH	Kanon.	
147	LH	Sparar ni endast den senaste uppdaterade versionen av varje dokument eller sparas även de äldre?	ISD+T
148	R5	Confluence har ett versionshanteringssystem, så vår SharePoint version och Confluence, båda två har versionshantering.	ISD+T
149	LH	Så då är det egentligen bara den senaste versionen som ni har tillgång till?	ISD+T
150	R5	Ja, vi har ju tillgång till alla andra också men vi kollar ju bara på den senaste.	ISD+T
151	LH	Ja, okej, men man kan gå tillbaka om man skulle vilja göra det?	ISD+T
152	R5	Ja, man kan också se vem som ändrat, var och hur.	ISD+T
153	LH	Okej.	
154	LH	Hur skulle du säga att tillgängligheten av den interns systemdokumentationen bidrar till delningen av kunskap?	ISD+T+ KD
155	R5	Om den hade varit mer tillgänglig, om jag hade kunnat läsa mer R&D grejer så hade jag ju kunnat utveckla ännu mer och kunnat hjälpa till ännu mer, så det stoppar mig ibland.	ISD+T+ KD
156	LH	Okej, men hur tycker du då att begränsad tillgänglighet, som till exempel R&D, med intern systemdokumentation påverkar utveckling, underhållsarbete och systemet?	ISD+T+ KO
157	R5	Det blir ju lite försenat ibland. Det går ju långsammare, det läggs till extra tid. Och det kommer ju leda till att projektet i sin tur blir mer kostsamt.	ISD+T+ KO
158	LH	Kanon, det var alla våra frågor, men är det möjligtvis någonting du hade velat lägga till som du kommit på eller så?	
159	R5	Nä, jag tror det var allt, det var väldigt heltäckande frågor.	
160	LH	Okej.	
161	LH	Vi undra också om det är okej att vi kontaktar dig om det kommer upp någon fråga eller om vi behöver att du förtydligar någonting.	
162	R5	Gör så, det är lättast att nå mig på mejl.	
163	LH	Super.	
164	LH	Vi undrar också om du vill ta del av transkriberingen, titta över så det är korrekt och ser bra ut?	
165	R5	Om ni vill så kan jag kolla igenom det, så att ni vet att det är någon som kontrollerat. Men jag ser gärna er uppsats sen också.	
166	LH	Det får du absolut. Det lovar vi.	
167	R5	Spännande.	
168	MH	Ja, men tack så jättemycket då!	
169	R5	Ingen fara.	
170	LH	Tack.	

## Bilaga G – Respondent 6, transkribering

R6: Respondent 6

MH: Mathilda Hezsö

LH: Linnéa Hansson

Rad	Person	Meningsenhet	Kod
1	MH	Först och främst så undrar vi lite vad ditt företag gör?	
2	R6	Alltså det är lite svårt att säga, jag jobbar ju på ett konsultbolag, så vi hjälper företag att jobba smartare med mjukvaruutveckling.	
3	MH	Ja.	
4	R6	Danskt företag som helt fokuserat på Continuous Delivery och Continuous Development. DevNoOps är vår riktiga grej, att det inte ska finnas DevOps team, utan utvecklaren ska själva kunna skriva sina krav... eller inte krav. Men alltså att gå från att man får krav till att man kör sin ändring i produktionen, det är målet liksom, så man ska inte göra hand-overs till andra. Så det är många av våra kunder i traditionella bolag som har varit i industrin jättelänge som upptäckt nu, som ungefär lastbilar, att deras businessmodell bygger på att göra motorer men allt annat är så nytt för dem.	
5	MH	Okej, så vad är då din yrkestitel?	
6	R6	Continuous Development Coach eller Software development consultant, lite vad som helst faktiskt.	
7	MH	Okej, arbetsuppgifter är?	
8	R6	Ja, det kan vara att... vi försöker häva att vi inte är traditionellt konsultbolag, vi säljer liksom inte CV. Som exempel, ett *företag* är våra kunder och då har vi ett avtal så att dem köper en service av oss där vi har ett kontrakt att vi ska jobba 1500 timmar i halvåret år dem. Beroende på vad uppgiften är, just nu håller dem på att införa Azure ganska brett, då betyder det att även om jag har varit där mycket så har kanske inte jag så bra på Azure, så då kommer någon annan och gör det. Man hyr en konsult, men man får ett team och det är vi som bestämmer vem som hjälper kunden bäst. Inte alla företag går med på detta, men i den bästa världen.	
9	MH	Okej.	
10	R6	Det handlar mycket om byggsystem och verktygkedjor.	
11	MH	Mm.	
12	LH	Vi tänker att vi tar oss in lite på den interna systemdokumentationen och undrar om det är någonting som du aktivt använder dig av i ditt vardagliga arbete?	ISD
13	R6	Alltså det är det ju. Eller bristen av... Det är svårt att säga, antingen så ser vi ju, att vi jobbar mot GitHub att dokumentationen finns i ReadMe filer, den är ganska teknisk. I den bästa världen så uppdaterar man dokumentationen i takt med att man gör ändringar, tyvärr är det ju inte alltid så. Ibland hittar man dokumentation i version 0.1 och mjukvara i version	ISD

		7, det finns liksom ingen koppling. I det andra fallet, jag vet inte om ni använd Atlasian, Jira?	
14	MH	Ja, vi vet vad Jira är.	
15	R6	Ja, då har dem en Wiki-sak som heter Confluens. Den är kopplad till en databas. Man kan göra ganska bra lösningar där man knyter ihop liksom krav till ändringar till *ohörbart*. Så ibland dem som lägger lite mer tid på det, dem kan göra ganska bra intern dokumentation. I dem fall där alla känner sig delaktiga i systemdokumentationen så kan detta fungera, att man faktiskt uppdaterar efter hand, men väldigt ofta ser vi att dem som är intresserade av systemdokumentation, dem kanske inte är så aktiva i projekten. Produktägare, arkitekter, som mer bestämmer hur man ska jobba, det är det som är dokumenterat, snarare än vad det blev eller hur man gjorde. Då kan man också se att systemdokumentation ibland så finns det en annan, det går i en annan loop, för krav på kod, tester, testfall så har man någon slags av review-förfarande. Inte alltid, det finns företag som skiljer på verifiera och validera. Man kan göra allt man kan göra automatiskt i verifiering och så fort man behöver människor som ska tycka saker då är det validering, då gör man det sen.	ISD
16	MH	Mm.	
17	R6	Systemdokumentation kan då ha andra stakeholder eller andra intressenter som inte längre kanske jobbar så aktivt med koden och därför inte riktigt har hängt med hur man jobbar med pull-request eller vad är Git. Då får man i olika takt, mjukvaruutvecklingen går i en annan takt än dokumentationen. Det är klassiskt sådant fall där underhåll av systemdokumentation, massa möten och massa kommentarer, men den är helt frikopplad från vad utvecklarna tycker i varligheten, det vill säga hur koden ser ut och vad som fallerar och när. Det är svårt ämne.	ISD
18	LH	Ja, det har vi förstått.	
19	MH	Mm.	
20	LH	Skulle du säga att den intern systemdokumentationen som ni har förenklar ditt arbete?	ISD
21	R6	Det är det som är problemet... om den är bra. Om man säger såhär, jag jobbade ju på *företag* i 10 år. Där fanns det en Wiki-sida som var en blandning av total katastrof, men det fanns också sidor som var fantastiska. Efter att ha varit där så länge så visste man ju om vad man kunde lita på. När jag kom till ett nytt företag och där har jag ingen aning... Dem har också en Wiki-lösning för delar av systemdokumentation och så har dem liksom någon slags dokumenthantering för designdokument. Och då kan man väl tänka sig att man kan lita på designdokumentet för dem är ändå avsikten är i alla fall att dem ska vara rätt, men jag har ingen möjlighet att lita på det som står på Wiki. Jag måste prova det som står på Wiki. Är detta sant och sen när man upptäcker att detta inte funkar då	ISD, ISD+K, ISD+T

		får man ju ”du kan ju uppdatera Wiki”. Så det gör ju livet lättare men man måste ändå veta vad man kan lita på.	
22	LH	Vad är det som avgör det, om du kan lita på det eller inte?	ISD
23	R6	Det är om det som står där är rätt. Alltså har det rätt kvalitet. Det är på samma sätt som vissa människor i många av dessa företagen... Det går så fort och chefen säger att nu ska vi ha *program* för det har man på annat företag. Då sitter där massa medelålders människor, som jag då, och säger att men jag var ju en stjärna på *program*, men det använder vi ju inte längre. Man blir liksom promotad till en viss nivå för hur det var då. Och det går ju fortare och fortare, så träffar man på fler människor som inte hänger med i utvecklingen och dem sköter ofta om sånt som systemdokumentation. Dem kan ju göra nytta, men man måste lära känna systemet lite och förstå vad det är vi gör. Då vet man ofta om detta är bra.	ISD, ISD+K
24	LH	Ja, okej.	
25	LH	Vi har också lagt lite fokus på kunskapsdelning och undrar lite hur det fungerar hos er?	KD
26	R6	Just nu är jag i ett projekt där det inte alls funkar.	KD
27	LH	Nä, okej.	
28	R6	Ja, men dem är väldigt traditionella, eller samtidigt inte. Ett danskt bolag som strävar efter någon slags svensk konsensus. Där är en chef och det är ganska outtalat att det är han som bestämmer, men han vill inte riktigt bestämma. I ett *företag* i Göteborg sker kunskapsdelning att allting som är svårt och gemensamt så sker kunskapsdelning genom att man försöker sig på user groups eller key user meetings och regelbundet pratar om vad som är nytt och vad som gått extra bra eller extra dåligt. Det blir ganska mycket verktyg och sådana typ, ska man ha vettig utveckling så måste man ju kunna samla in data så man kan ge sina data någon slags metrics, så dem inte går runt och stör verksamheten och ställer dumma frågor. Dem får dashboards så man kan se vad som händer och allting går framåt.	KD
29	LH	Mm.	
30	R6	Där vi ser att det lyckas så har man ofta interna samarbeten, grupper, kommunikation. Vi på vårt eget bolag vi kör video varje fredag. fredag klockan 12 är det någon som pratar om något. Vi är ju 3 i Malmö, men betydligt fler i Norge, Finland och Danmark.	KD
31	LH	Okej.	
32	LH	Om det är så att ni har någon som ska lämna företaget, hur gör ni då för att försöka bibehålla så mycket kunskap som möjligt?	KD
33	F6	Alltså det är väldigt... Man tänker sig att min kund dem permitterade en massa människor och det bestämdes på torsdagen, man blev informerad på fredag och man skulle gå hem på månad. Och så har dem i takt med det också avslutat alla konsultkontrakt. Vilket vi redan märkt. Detta gällde då Sverige, men Indien, Frankrike, USA, Korea, dem ska få jobba vidare. Så nu har vi fått meddelande från någon stackare i	KD, KD+ISD

		Indien som säger att hans byggserver har förvunnit. Men sådant försvinner ju inte, det är ju bara att hitta den och starta om den. Men den var faktiskt borta. Det var en virtuell maskin, som var skapad av två tidigare konsulter, finns ingen dokumentation, vi vet att dem har gjort det med ett script, man måste göra det programmerbart och spinna upp den här maskinen. Men vi vet inte var scripten finns, så det var bara sorry, det kan vi inte lösa. Så det finns ju bolag som gör detta väldigt dåligt, vi själva försöker att aldrig jobba ensamma, vi jobbar alltid parvis, minst parvis men gärna i par som ändrar sig så att vi sprider kunskap. Man ska kunna sluta om man vill.	
34	LH	Mm. Ja, okej.	
35	R6	Annars blir det ju sådana komplicerade överlämningar när man sagt upp sig. Vissa konsulter har ju inte det, men dem flesta har 3 månaders uppsägningstid och då om man säger upp sig så blir det lite möten med hand-overs för att hitta ansvarsområden. Och dem funkar ju inte. Om man jobbar väldigt traditionellt med väldigt tydliga roller, då kan man ju göra hand-over, då har man ju dokumenterat vad man gjort, men i verkligheten, dem som officiellt har rollen att vara något, så jobbar dem ju med väldigt många andra saker som är besläktade med det dem är tilldelade att göra. Dem faller ju ofta.	KD, KD+ISD
36	LH	Okej, men hur gör du för att hitta den kunskapen som behövs för att göra ett underhåll av system eller funktion?	KD
37	R6	Alltså jag brukar försöka läsa koden och köra dem tester som finns, om det finns. Sen tittar jag på dokumentation. Men det är mest för att jag då ska ha en liten chans att avgöra om dokumentationen är rimlig. Om det är något jag kan lita på. Om det funkar med den versionen som finns av systemet.	KD, KD+ISD
38	LH	Kanon.	
39	LH	Vi har ju som vi nämnde i början tagit fram tre stycken egenskaper eller problemområden där det ofta är lite problematiskt inom intern systemdokumentation och hade tänkt ställa lite mer specifika frågor kring dem. Då är den första uppdatering.	ISD+U
40	LH	Hur viktigt tycker du det är att den interna systemdokumentationen är uppdaterad?	ISD+U
41	R6	Alltså det är ju viktigt. Systemdokumentation som ska vara användbar måste vi kunna lita på. Om jag då ser att systemdokumentationen är från december förra året och nu är det maj, då är det klart att då är det svårt att lita på den. Det gör ju att den måste underhållas kontinuerligt. Behöver inte betyda att den är rätt, men då betyder det att någon har försökt och det är bättre.	ISD+U
42	LH	Okej, då vill du helst ha den uppdaterad för att kunna lita på den och använda den ordentligt?	ISD+U
43	R6	Ja.	ISD+U
44	LH	Kanon.	

45	LH	Hur ofta uppdaterar du i genomsnitt den interna systemdokumentationen?	ISD+U
46	R6	Det beror lite på. Det beror på litegrann om... Jag har ett antal tillfällen, vi har försökt säga att den tekniska beskrivningen har vi inte Git, text, inget trams och det följer samma regler för koden. När man uppdaterar koden uppdatera man dokumentationen. Sen kan man på andra ställen, där det är lättare att läsa, Confluens, Wiki, vad som helst, där kan man aggregera informationen och presentera den i mer läsbart format. Jag gillar ju Markdown, men det gör ju inte alla. Då är det lätt för mig att uppdatera och det gör det lätt för mig att se när den är uppdaterad och då uppdaterar jag den. Men om jag måste gå in i ett system och checka ut en Word-fil och uppdatera, då blir det inte. Det ligger då för långt bort. Det är för mycket waste.	ISD+U
47	LH	Okej.	
48	LH	Mm, skulle du säga att dokumentationsarbetet ser annorlunda ut beroende på storlek på kodändringarna?	ISD+U
49	R6	Ja, det gör den naturligtvis.	ISD+U
50	MH	Hur då? Vilket prioriteras?	ISD+U
51	R6	Nä, men nu är det ju... Nu hamlar vi i en fälla för om man gör för stora kodändringar då gör man ju fel. Om man har ett existerande system och så ska man göra nästa generation och så ska man lägga till en ny funktion, då är det ju ofta större features, då ska ju den dokumentationen vara inkrementell eftersom koden ska vara gjort inkrementell. Stora ändringar är inte nödvändigtvis bättre dokumenterade, men det är fler som vet att det händer, så då blir det nog i alla fall av. Sen är nog kvaliteten samma.	ISD+U
52	LH	Okej.	
53	LH	Är det så att om du stöter på dokumentation som är lite gammal, anser du att det finns användningsområde ändå?	ISD+U
54	R6	Nej, lite. Man kan kolla vem har skrivit det här, är hen kvar och då gå och fråga, då ersätter man systemdokumentation med personligt nätverk istället.	ISD+U, KD
55	LH	Okej.	
56	R6	Det är ju nästa problem. Då hoppar man över dokumentation att man vet vem det är man kan gå och prata med.	ISD+KD
57	LH	Ja, precis.	
58	LH	Sådan här icke-uppdaterad dokumentation hur anser du den påverkar delandet av kunskap?	ISD+U+ KD
59	R6	Alltså den skapar ju silos, eller öar. Det betyder ju att om man hitta gammal dokumentation som man inser att en person har skrivit som man känner, då kan man gå och fråga honom. Men om ni inte känner den personen då kommer ju den dokumentationen inte ge er samma värde. Då skapar man dem här informella nätverken och det betyder ju att vi som är dubbelt så gamla som ni och varit där längre har ett större nätverk och kan känna att fan vad jag är duktig. Jag tror ju att den borde tas bort. Det är som issues i backlogen som har legat	ISD+U+ KD



		där ett år, om ingen använt den då tar man bort den. Dokumentation borde vara likadant. Om den legat där ett år och ingen använt den, då bort med den.	
60	LH	Mm.	
61	LH	Sådan här icke-uppdaterad dokumentation hur tror du att det påverkar systemutveckling, underhållsarbetet och systemet?	ISD+U+ KO
62	R6	Det går långsammare och så blir det sämre.	ISD+U+ KO
63	LH	Allting blir sämre? Systemet, utvecklingsarbetet?	ISD+U+ KO
64	R6	Ja, det betyder ju att, då kommer det ju finnas en klick... Alltså jag ser det nu, jag såg det idag. Vi har ett system, kunden har traditionellt jobbat på ett sätt med inbyggda system, men blivit påtvingat något nytt och häftigt, som Git eller... Men utvecklarna har inte riktigt den erfarenheten av att jobba i sådana system, dem måste liksom uppdatera sina SUB moduler, bygga en doc-container. Det finns så många steg dem kan gå vilse. Om vi inte skrivit dokumentation som är både bra för nybörjarna och äldre experter...	ISD+U+ KO
65	LH	Mm.	
66	R6	Det finns mycket äldre systemutvecklare som anser att det är systemet som är dum i huvudet, inte jag själv. Därför behöver vi dokumentation som både är noob-friendly, så att man kan vi snabb anblick se att det är det här jag ska göra, om jag vet vad det betyder så kan jag gå och göra det. Om jag inte vet hur man gör så måste jag kunna expandera informationen. Problemet blir då oftast om man bara skriver textfiler har man så mycket löpande information, att dem som kan sluta sedan läsa. Man behöver ett dokument som kan vara användbart för den som arbetat länge och kan alltihop och den som kommer ny utifrån.	ISD+U+ KO
67	LH	Okej, ja.	
68	MH	Yes, sen har vi då nästa egenskap som handlar om kvalitet på den interna systemdokumentationen och vi undrar vad du anser motsvarar hög dokumentationskvalitet?	ISD+K
69	R6	Det är att den är lättillgänglig, tillförlitlig, lätt att använda och lätt att uppdatera.	ISD+K
70	MH	Är det dina förväntningar på kvaliteten?	ISD+K
71	R6	Nä, mina normala förväntningar är att den inte finns eller att det är irrelevant eller att det är... Nej, men det jag vill ha är att det finns ReadMe-filer kring alla komponenter. Jag läser inte designdokument eller dem övergripande spakarna oftast.	OSD+K
72	MH	Följer ni detta i ert företag, har ni ReadMe-filer på allting.	ISD+K
73	R6	Ja, ja det tycker jag vi har.	ISD+K
74	MH	Mm, okej.	
75	MH	Vi har ju hittat några kvalitetsattribut som påverkar systemdokumentationen och vi undrar hur viktiga du tycker dem olika är.	ISD+K

76	MH	Den första handlar om korrekthet, hur viktigt är det att allting är översatt korrekt, att informationen är korrekt, kommentarer et cetera?	ISD+K
77	R6	Korrekt som i tekniskt korrekt eller språkligt korrekt?	ISD+K
78	MH	Både och.	ISD+K
79	R6	Det är ju viktigt att det är tekniskt korrekt, det är inte lika viktigt att det är grammatiskt och swenglish fritt, vilket jag nog tyckte var viktigt när jag var yngre.	ISD+K
80	MH	Mm.	
81	R6	Läsbart och förståeligt är mindre viktigt, men tekniskt korrekt är mycket viktigare.	ISD+K
82	MH	Okej.	
83	MH	Är det viktigt med struktur? Har ni någon standard för hur dokumentationen ska se ut?	ISD+K
84	R6	Nä, det har vi inte. Eller det beror lite på. När systemen blir större så blir strukturen desto viktigare.	ISD+K
85	MH	Okej.	
86	MH	Fullständighet då. När vi har läst litteratur så har det kommit fram att det är mycket utvecklare som använder sig av ofullständiga dokument bara för att få ut det lilla innehållet. Tycker du det är okej eller borde det alltid vara helt fullständigt?	ISD+K
87	R6	Hmm, alltså det där är ju en svår fråga, för det betyder att jag måste skriva komplett dokumentation. I och med att det i väldigt många fall kanske borde vara så att man strävar efter komplett dokumentation, men det blir inte alltid så. I praktiken är det bättre att få lite än inget alls.	ISD+K
88	MH	Ja, det är ju rimligt.	
89	R6	Man kan ju även dokumentera vad det är som saknas, det är ju nu bättre.	ISD+K
90	MH	Skulle du säga att du tar hänsyn till dessa attribut när du skriver dokumentation?	ISD+K
91	R6	Nej, det ska jag ärligt säga att det gör jag inte.	ISD+K
92	MH	Okej, hur ofta tycker du då att den systemdokumentation innehåller all den nödvändiga information som krävs för att du ska förstå, underhålla och använda systemet?	ISD+K
93	R6	Alltså, nästan aldrig, skulle jag vilja säga, men lite grann om jag tittar tillbaka på dem företag jag varit på dem sista fem åren, så kan jag säga att startsträckan... Hur lång tid tar det för mig att få mitt passerkort och min dator till att jag faktiskt kan göra någonting i systemet... Den bolaget jag är på nu, där var det en, en och en halv vecka innan jag kunde checka ut, bygga, göra och förstå var saker fanns. Detta är ett ganska stort företag, men ett ganska litet team i företaget. Medan i andra företag kan det ta månader innan man ens vet var man ska vända sig. Det beror ju på att dokumentationen är för dålig.	ISD+K
94	MH	Ja, okej.	
95	LH	Mm.	

96	MH	Hur tycker du då att kvalitetsnivån bidrar till delning av kunskap?	ISD+K+KD
97	R6	Det är ju så att om dokumentationen är bra, så betyder det att kunskapen är delad och att det är fler som... För att få bra dokumentation så måste fler hjälpa till och vara involverade och jobba med detta. Så jag tror att det är den delande kunskapen som är förutsättningen för att få bra dokumentation. Om man har ett team som är bra på att dela kunskap, då har dem förutsättning för att skriva bra dokumentation.	ISD+K+KD
98	MH	Bra.	
99	MH	Hur tycker du då att intern systemdokumentation med dålig kvalitet påverkar systemutvecklingen och systemet?	ISD+K+KO
100	R6	Den skapar en massa waste. Först skapar det ju tekniskt skuld i den meningen att... på ett gammalt företag jag arbetade på så visade det sig på en produkt att man använder ett system som inte fungerade, för att någon hade missförstått dokumentationen, sen blev det alldeles för dyrt för att ändra det. Man hade för många projekt för att ändra det. Det gjorde man aldrig. När man väl har fått in teknisk skuld på en så fundamental nivå så kommer det ju vara där föralltid.	ISD+K+KO
101	MH	Ja, okej.	
102	R6	Så jag tror att då tar allting längre tid för att det blir sämre.	ISD+K+KO
103	MH	Absolut.	
104	LH	Vi har ju då vårt siste område som handlar om tillgänglighet. Hur lätt är det att få tillgång till den interna systemdokumentationen hos er?	ISD+T
105	R6	Det beror på... Om den ska vara användbar så måste den vara lätt tillgänglig. Det beror på vilket företag vi pratar om.	ISD+T
106	LH	Där du är just nu tillexempel? Är det lätt för dig att logga in och hitta dokumentationen eller är det i en pärm någonstans i ett skåp med en nyckel du inte har?	ISD+T
107	R6	Det beror på. Det finns Wiki som till synes är strukturerad, den är lättillgänglig. Sen finns det lite mer dokumentation som vi upptäckte inte är så lättillgängligt för där har folk lagt upp saker på lite olika sätt. Så det finns, men då måste man veta vem det var som skrev det dokumentet, för att veta om det ligger i OneDrive eller här eller där. Eller kanske i ett dokumenthanteringssystem som jag inte ens känner till.	ISD+T
108	LH	Hur önskar du den var försvarad då för att bringa mest nytta för dig?	ISD+T
109	R6	Jag önskar att den var... Vi har ju bestämt att flytta allt detta till Git och Markdown, så att man kan bygga websidor av detta. Så jag vill att den ska vara i Git, jag vill ha den i text, inga databaser, inga Word-dokument. Den ska vara lätt att nå och maskinläsbar, för då kan man sen göra om den till dem behov som finns.	ISD+T
110	LH	Okej, sparas endast den senaste uppdaterade versionen eller kan man få tag i dem äldre?	ISD+T

111	R6	Sparar man versionerna i Git, så finns ju alla versionerna. Använder man Word så brukar dem pusha för Word och se att kolla här finns historik. Men det är väldigt svårt att plocka fram. Det är en blandning. I bästa fall sparas versionerna, i värsta fall sparar vi en kopia av allting.	ISD+T
112	LH	Men är det lätt att veta vilken version som du ska använda?	ISD+T
113	R6	Nej, det tycker jag inte.	ISD+T
114	LH	Kan det uppstå en del förvirring där då?	ISD+T
115	R6	Ja, det gör det. Ett exempel är ett företag jag arbetade på där folk i Lund jobbade på version nu, men det fanns underkonsulter i Kina eller Indien som inte ens kände till att vi hade ändrat hur vi jobbat två tre gånger, så dem hade liksom en kopia av en gammal version. Så hur informationen flödar i stora bolag gör ju att en ny idé kan vara hur bra som helst att spara in tid och pengar, att jobba mycket smartare, men det kan ta månader innan alla känner till. Det har ju tagit livet av massa bra idéer.	ISD+T
116	LH	Så det leder till att idéer krossas?	ISD+T
117	R6	Ja, mycket av det som är det nytänkande och nyskapande sågas av folk som inte riktigt förstår och som inte kan bidra.	ISD+T
118	LH	Men sådan här begränsad tillgänglighet av den interna systemdokumentationen, tycker du att det påverkar systemutvecklingen, underhållsarbete och systemet?	ISD+T+ KO
119	R6	Alltså om teamet är tillräckligt litet så kommer systemet utvecklas ändå. Men om man ska skala eller tar in nya eller blir av med någon så ökar bussfaktorn, sannolikheten att man blir beroende av att ha teamet kvar blir större. Risken ökar, men om kvaliteten minskar eller ökar, det är svårt att säga. Utvecklingen och nyutveckling kanske går lika fort, men underhåll kommer ju ta längre tid för då är ju inte folk kvar även om dem i teorin borde va det.	ISD+T+ KO
120	MH	Okej, då var våra frågor slut. Tack så mycket för du ville delta.	
121	LH	Har du något mer att kanske tillägga?	
122	R6	Nej, jag tycker jag har sagt det mesta.	
123	LH	Okej, och om vi skulle vilja ha svar på någon fråga eller få något förklarat lite extra, kan vi kontakta dig då?	
124	R6	Ja, det går absolut bra.	
125	LH	Sista, vill du ha transkriberingen att kolla igenom så allting är korrekt?	
126	R6	Nej, jag litar på att ni gör det sanningsenligt. Men om ni vill att jag ska kolla igenom den så kan jag göra det.	
127	MH	Okej, då tackar vi så jättemycket. Ha en fortsatt trevlig dag.	
128	R6	Tack, detsamma.	