

DOKUMENTATION OCH ANSVAR FÖR ICKE-FUNKTIONELLA KRAV

- *PRAXIS ELLER ETT NÖDVÄNDIGT ONT?*

Kandidatuppsats i Informatik

Louise Andersson
Ronja O'Kane
Margarita Pålsson

VT 2023:KANI21



HÖGSKOLAN
I BORÅS

Förord

Vi vill tacka vänner, verksamhetskontakter och familj som har hjälpt oss att göra denna studie möjlig genom att skicka vår enkätundersökning till alla de kunde tänka sig ingå i vår målgrupp för studien. Vi vill även tacka alla som tagit sig tid att svara på vår enkät. Sist vill vi tacka kursansvariga Anders Hjalmarsson Jordanius och Antonios Tsertsidis samt vår handledare Mats Jadeskär för de intressanta diskussionerna och dess tillhörande metaforer. Denna uppsats är förhoppningsvis vad ett strålkastarljus är i ett mörkt rum.

“It's really complex to make something simple.”
– Jack Dorsey, 2011

Svensk titel: Dokumentation och ansvar för icke-funktionella krav - Praxis eller nödvändigt ont?

Engelsk titel: Documentation and Responsibility for Non-Functional Requirements - Practice or Necessary Evil?

Utgivningsår: 2023

Författare: Louise Andersson, Ronja O’Kane och Margarita Pålsson

Handledare: Mats Jadeskär

Abstract

This study examines non-functional requirements (NFRs), also known as quality requirements, in Swedish software development projects. Requirements engineering is considered the most crucial part of software development, which is still a challenge in the majority of projects today. In particular, the NFRs are the most challenging seeing as there are ambiguities concerning their definition. This in turn leads to lack of documentation and, ultimately, their deprioritization. To address these issues, this study aimed to answer how well NFRs are documented in Swedish software development projects and who can be responsible for ensuring the quality of these NFRs. A mixed-method was conducted with both a qualitative literature review of previous research and a quantitative study. The data collection in the quantitative study was conducted using a web-based survey with 24 questions. The survey was sent to 72 software development companies in Sweden. A total of 43 responses were collected, and a univariate analysis was performed. The survey indicated that respondents have a positive attitude towards documenting. Additionally, most respondents lack comprehensive knowledge about NFRs and clear roles for the NFRs are missing. Standardization, better guidelines, and clear roles are needed to ensure the quality of NFRs, which are currently lacking. A solution or software architect is best suited to ensure the quality of NFRs.

This study, written in Swedish, can benefit students, researchers, and professionals as it highlights the importance of NFRs and suggests improvements for existing deficiencies.

Keywords: Agile software development, Documentation, Documentation techniques, Documentation methods, Non-functional requirements, Responsibility, Requirements engineering, Roles, Software development, Traditional software development, Quality requirements, Quality assurance

Sammanfattning

Denna studie problematiserar icke-funktionella krav (IF krav), även benämnt kvalitetskrav, i svenska mjukvaruutvecklingsprojekt. Kravhantering anses vara den mest avgörande delen i mjukvaruutveckling för att lyckas, vilket än idag är en utmaning i majoriteten av projekten. Särskilt svårt är det att lyckas med IF krav, då tvetydigheter leder till brist på dokumentation och slutligen bortprioriteras de. För att adressera dessa problem syftade denna studie att besvara hur väl de IF kraven dokumenteras inom svenska mjukvaruutvecklingsprojekt samt vem som kan ansvara för att kvalitetssäkra de IF kraven. En mixed-method har genomförts med både en kvalitativ litteraturstudie av tidigare forskning samt en kvantitativ studie. Datainsamlingen i den kvantitativa studien har skett med webbaserad enkätundersökning med 24 frågor. Enkäten skickades ut till 72 företag i Sverige vars huvudsyssla var mjukvaruutveckling. Sammanlagt samlades 43 svar in och en univariat analys genomfördes. Enkätundersökningen indikerade att respondenterna är positivt inställda till dokumentation av IF krav. De flesta saknar dessutom gedigen kunskap om IF krav. Vidare saknas tydliga ansvarsroller för dokumentationen av IF krav. En lösnings- eller mjukvaruarkitekt är bäst lämpad för att kvalitetssäkra de IF kraven.

Denna studie är skriven på svenska och kan främja studenter, forskare och yrkesverksamma då den nyanserar vikten av IF krav och hur existerande brister kan förbättras.

Nyckelord: Agil mjukvaruutveckling, Ansvar, Dokumentation, Dokumentationsmetoder, Dokumentationstekniker, Icke-funktionella krav, Kravhantering, Kvalitetskrav, Kvalitetssäkring, Mjukvaruutveckling, Roller, Traditionell mjukvaruutveckling

Innehållsförteckning

1 Inledning	7
1.1 Bakgrund	7
1.2 Forskningsöversikt	8
1.2.1 Definition av icke-funktionella krav	8
1.2.2 Kvalitetssäkring i kravhanteringsprocessen	8
1.2.3 Systemutvecklingslivscykeln	9
1.2.3.1 Traditionell kravhantering	9
1.2.3.2 Agil kravhantering	9
1.3 Problemdiskussion	10
1.4 Problemformulering	12
1.5 Syfte och forskningsfrågor	13
1.6 Målgrupp	13
1.7 Avgränsning	13
2 Metod	14
2.1 Forskningsstrategi	14
2.2 Kvantitativ metod med kvalitativt perspektiv	15
2.3 Litteraturstudie	16
2.3.1 Artikelsamling	16
2.3.2 Valda Artiklar	17
2.3.3 Presentation av valda artiklar	19
2.3.4 Motiv till valda artiklar	20
2.3.5 Litteraturanalys	22
2.3.6 Teman	22
2.4 Kvantitativ undersökning	24
2.4.1 Urval av respondenter	24
2.4.2 Respondenterna	25
2.4.3 Databaserad datainsamling	32
2.4.4 Enkätundersökning	33
2.4.5 Webbaserad enkät	35
2.4.6 Dataanalys	36
2.5 Metodreflektion	36
2.5.1 Reliabilitet	36
2.5.2 Validitet	38
2.5.3 Etiska ställningstaganden	40
3 Teoretiskt ramverk	41
3.1 Utmaningar med att identifiera IF krav	41
3.1.1 (Tyst) Kunskap om IF krav	42
3.2 Dokumentationens betydelse för IF krav	42
3.2.1 Tillvägagångssätt för att dokumentera IF krav	44
3.3 Faktorer som påverkar kvalitetssäkring av IF krav	47
3.3.1 Säkerställning av kvalitet enligt svensk standard (SS-EN ISO 9001:2015)	48

3.4	Ansvarsroller för att kvalitetssäkra IF krav	49
3.5	Sammanfattning av det teoretiska ramverket	50
4	Resultat	51
4.1	Univariat analys	51
5	Analys och diskussion	62
5.1	Utmaningar med att identifiera IF krav	62
5.1.1	(Tyst) kunskap om IF krav	63
5.2	Dokumentationens betydelse för IF krav	65
5.2.1	Tillvägagångssätt för att dokumentera IF krav	66
5.2.2	Kombination av dokument	68
5.3	Faktorer som påverkar kvalitetssäkring av IF krav	69
5.3.1	Ansvar som faller mellan stolarna	70
5.3.2	Både kunskap och entusiasm är nödvändiga	70
5.3.3	Kunden påverkar kvaliteten	71
5.4	Ansvarsroller för att kvalitetssäkra IF krav	72
5.4.1	Senioritet	73
5.4.2	Respondenterna som kunde identifiera IF krav	73
5.4.3	Kommunikation	74
5.5	Sammanfattning av analys och diskussion	75
5.6	Sammanfattning av nya upptäckter	77
6	Slutsatser	79
6.1	Generalisering	81
6.2	Framtida forskning	82
	Källförteckning	84
	Bilaga 1 - Enkätfrågor	89

1 Inledning

I följande kapitel presenteras uppsatsens ämne. Vidare presenteras forskningsöversikt, problemformulering, uppsatsens syfte samt de frågeställningar som studien ämnar besvara. Avslutningsvis presenteras studiens målgrupp samt de avgränsningar som har gjorts.

1.1 Bakgrund

Enligt Chaos-rapporten som publiceras av The Standish Group (2015) är det bara 29 % av alla mjukvaruutvecklingsprojekt som lyckas. Särskilt svårt är det att lyckas med stora projekt där endast 2 % av de största projekten har lyckats. Utav de allra minsta mjukvaruutvecklingsprojekten är det 62 % som lyckas, vilket påvisar att även i de bästa fallen är det idag en väsentlig andel av mjukvaruutvecklingsprojekt som inte lyckas. Detta är inte bara en enorm ekonomisk kostnad för skattebetalarna enligt Riksrevisionens rapport (2011) då kostnaderna för misslyckade statliga projekt uppgår till 200 miljoner kronor per år, utan även ett alarmerande resursslöseri. Kocak (2013) menar att informationsteknologi (IT) kan bidra till hållbarhet genom att använda mindre resurser vid utveckling, eftersom IT idag står för cirka 2 % av världens CO2 emissioner (lika mycket som flygindustrin).

Inom mjukvaruutveckling är människans kunskap det mest kritiska verktyget enligt Wohlin, Šmite och Moe (2015), men inte på grund av att det är komplex kod som ska produceras, utan på grund av arbetet inför produktionen där kravhantering (*eng: requirements engineering*) påbörjas. På grund av att det är många olika typer av personer från olika verksamhetsområden och med olika kompetenser som ska förstå och interagera med varandra, är mjukvaruutveckling till stor grad beroende av ledarskap, det vill säga de ansvariga personerna.

Kravhantering kan definieras som processen att specificera kundens förväntningar och behov, som görs för att kunna skapa det som efterfrågas (Shafiq et al. 2018). Enligt Browne och Ramesh (2002) är det ett vedertaget faktum att kravhantering är den mest kritiska fasen av ett mjukvaruutvecklingsprojekt och argumenterar att det finns mycket bevis inom forskningen som föreslår att kravhantering både kan och borde förbättras. Enligt Riksrevisionens rapport (2011) var de mest förekommande svaren på varför statliga IT-projekt misslyckats i Sverige; oklara mål och krav, brister i riskhantering, brister i kalkylering och tidsplanering samt brister i metodval och metodanvändning.

Krav delas konventionellt upp i två kategorier: icke-funktionella krav (IF krav), även benämnt kvalitetskrav (*eng: quality requirements*) och funktionella krav (FK). En vanlig och översiktlig förklaring av skillnaderna mellan dem är att ett FK beskriver *vad* ett system ska göra medan ett IF krav beskriver *hur* systemet ska göra något (Eckhardt, Vogelsand & Fernández 2016). Således är de FK ofta något specifikt som systemet ska göra och ett IF krav är ett krav på hur kvaliteten ska utformas. Enligt Domah och Mitropoulos (2015) räcker det idag inte längre att utveckla mjukvara som endast gör vad den ska, den måste produceras med hänsyn till de IF krav som behövs för att säkerställa att produkten uppnår en viss kvalitet och kundens förväntningar.

Behutiye et al. (2022) konstaterar det som många tidigare nämnt men inte fastställt: Utmaningarna med att dokumentera IF krav ligger i att det råder begränsad medvetenhet,

tillvägagångssätt och artefakter kring att dokumentera IF krav. För att lyckas med IF krav är dokumentationen en väsentlig del vilket leder oss in på vårt forskningsområde.

1.2 Forskningsöversikt

I följande avsnitt presenteras definition av IF krav, kravhanteringsprocessen enligt traditionell och agil metodik, samt olika utmaningar gällande dokumentation av IF krav utifrån tidigare forskning i problemdiskussionen.

1.2.1 Definition av icke-funktionella krav

Både forskare och yrkesverksamma menar att namnet icke-funktionella krav (*eng: non-functional requirements*) är vilseledande eftersom det indikerar att något "icke" ska göras till skillnad från termen kvalitetskrav som är en alternativ term för samma fenomen (Ali, Khalil, Ahmad, Parveen & Zaman 2022; BA Video Series 2020). Glinz (2007) argumenterar att kvalitetsattribut är ett mer passande begrepp eftersom IF krav är krav som beskriver kvalitetsattribut på mjukvaran och förkortar även begreppet till enbart "attribut". Termen bör dock inte förvirras med krav som är av hög kvalitet. Det vill säga adekvata, entydiga, konsekventa och verifierbara (Glinz 2007). I denna studie kommer begreppet IF krav användas, eftersom det är den term som används i BABOK (Business Analysis Body of Knowledge) och anses vara den mest förekommande termen (AltexSoft 2022; Eckhardt, Vogelsand & Fernandez 2016). En tydlig och konsekvent definition av vad som klassas som IF krav saknas än idag, även om många ansatser att definiera fenomenet på ett tydligt sätt gjorts inom forskningen. En vanligt förekommande men inofficiell beskrivning är att IF krav är "*the -ilities*"; *portability, accuracy, readability, security, maintainability, usability och reliability*, bland andra (Glinz 2007). Bukky Campbell, senior Business Analyst inom cybersäkerhet och arkitektur på Sky, menar att det delvis är svårt att sätta exakta ord på vad IF krav är då det ständigt kommer nya IF krav, i och med att tekniken utvecklas och likaså våra förväntningar på kvaliteten av mjukvarusystemet som används (BA Video Series 2020).

Behutiye, Karhapää, Costal, Oivo och Franch (2017) fastställer att de IF kraven definieras utifrån två olika nivåer: generiska och detaljerade. De generiska är mer överskådliga och avser funktionalitet som berör hela systemet, exempelvis "Systemet måste vara användbart" eller att "Systemet måste se till att en viss funktion kan nås inom tre musklick". De mer detaljerade IF kraven är mer specifika och kan exempelvis vara "Svarstiden får inte överstiga 0.25 sekunder i 90 % av fallen".

I denna studie används definitionen av IF krav som beskriver *hur* ett system ska göra något med fokus på kvalitet (Eckhardt, Vogelsand & Fernández 2016), det vill säga alla "the -ilities" enligt Glinz 2007.

1.2.2 Kvalitetssäkring i kravhanteringsprocessen

Eriksson (2008) beskriver att följande processer finns i kravhanteringsprocessen: analysera och strukturera krav, prioritering av krav, dokumentation av krav, kvalitetssäkring, förvaltning och validering av krav.

Att kvalitetssäkra handlar om att säkerställa att de krav som är realiserade når upp till det som de förväntades göra. Detta är en process som pågår kontinuerligt genom hela arbetet. Ett sätt att säkra kvaliteten är att leverera delar av systemet i etapper, det vill säga genom

agila/iterativa processer eftersom feedback kan fås av kunden och förändringar/förbättringar kan göras. Kvalitet måste byggas in från början för att kunna säkerställa att systemet gör rätt från början och kunden får det som önskas (Eriksson 2008).

1.2.3 Systemutvecklingslivscykeln

Enligt Ruparelia (2010) följs en Systemutvecklingslivscykeln (SDLC) vid utveckling av mjukvara och beskriver det enligt följande: "A life cycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance." (ss.8) I SDLC ingår faserna (i kronologisk ordning): genomförbarhetsstudie för att se om projektet bör genomföras, kravanalys för att hantera kraven, design för att designa systemet, implementering där kodningen sker, testning för att validera att programvaran fungerar som den ska i alla skeden och sist driftsättning och underhåll av mjukvaran (ibid).

SDLC är en standardiserad metod för att organisera och styra mjukvaruutvecklingen och kan beskrivas som ett ramverk eller en övergripande process. Processen i sig kan genomföras på olika sätt, beroende på vilken modell som följs. Metoden beskriver således vad som görs medan valet av modell beskriver hur det görs (Ruparelia 2010). De två ansatserna som denna studien beskriver är traditionella kravhantering och agila kravhantering. Se avsnitt *1.2.3.1 Traditionell kravhantering* och avsnitt *1.2.3.2 Agil kravhantering* för att förstå två vanliga ansatser. Det finns dock fler tillvägagångssätt och modeller för att utveckla mjukvara, men i denna studie har enbart de traditionella och agila ansatserna beskrivits på ett övergripande sätt. Det finns skillnader mellan de två tillvägagångssätten som skildrar sig i fasen kravanalys som beskrivs nedan (ibid).

1.2.3.1 Traditionell kravhantering

Inom den traditionella kravhantering fastställs kraven i ett tidigt skede som sedan inte är ändringsbara. Detta medför svårigheter vid potentiella problem och förändringar som uppkommer under projektets gång då det är svårt att återgå till den föregående avslutade fasen om uppdatering och/eller förändring av krav skulle ske (Dennis, Wixom & Roth 2010). En annan nackdel med traditionell kravhantering är att det inte fungerar i stora och komplexa projekt då mycket, inklusive kraven, är okänt i början av sådana projekt. En fördel med den traditionella kravhanteringen är att det är lättförståeligt och innehåller tydlig och gedigen dokumentation, samt är systematisk och sekventiell (Shalloway, Beaver & Trott 2010).

1.2.3.2 Agil kravhantering

Som svar till den traditionella metoden publicerades i början av 2000-talet det "Agila Manifestet" som förespråkade en nyare metod för mjukvaruutvecklingsprojekt: ett iterativt arbetssätt (Agila manifestet 2001). Det agila manifestet förespråkar korta iterationer av alla faser av mjukvaruutvecklingen under en sekvens, varpå sekvensen görs om flera gånger. Synonymt med det agila tänket är snabba leveranser med kontinuerlig påbyggnad (Sharma, Sarkar & Gupta 2012). Bjarnason, Unterkalmsteiner, Borg och Engström (2016) beskriver att ansikte-mot-ansikte kommunikation prioriteras över skriven dokumentation, vilket kan intygas av Inayat, Salim, Marczak, Daneva och Shamshirband (2014) som hänvisar till att det förekommer dokumentation i mindre utsträckning enligt den agila mjukvaruutvecklingen. Inayat et al. (2014) beskriver att kunden specificerar sina önskningar i de frekventa mötena som förekommer mellan kunden och mjukvaruutvecklingsteamet, vilket även stärker

relationskapandet dessemellan. Önsknningar och krav prioriteras kontinuerligt av kunden innan de implementeras, det vill säga att kodas in i ett system, i nästa iteration av den agila mjukvaruutvecklingen.

Ochodek och Kopczyński (2018) beskriver att det finns vissa praktiker inom den agila kravhanteringen som verksamheter bör anamma för att lyckas med sin kravhantering. Det som prioriteras är att ha en delad vision av det som ska skapas, att ha kunden tillgänglig för frågor och samarbete, att organisera dagliga möten, att visa kunden vad som har implementeras via en demo, få kontinuerlig feedback och verifikation samt att göra kraven lättillgängliga.

1.3 Problemdiskussion

“Research has repeatedly shown that high-quality requirements are essential for the success of development projects” skriver Montgomery, Fucci, Bourafa, Scholz och Maalej (2022, s. 183) och bekräftar det som andra forskare konstaterar är ett vedertaget faktum: kravhantering med kvalitet är det kritiska för att lyckas med ett mjukvaruutvecklingsprojekt (Browne & Ramesh 2002; Behutiye et al. 2017).

Faserna i kravhanteringsprocessen är enligt Eriksson (2008) analysera och strukturera krav, prioritering av krav, dokumentation av krav, kvalitetssäkring, förvaltning och validering av krav. Faserna är desamma i både den traditionella och den agila kravhanteringen. Det som skiljer dessa åt är enligt Ochodek och Kopczyński (2018) att den agila kravhantering är en iterativ och människocentrerad process där en delad vision och dokumentation av krav beskrivs på ett lättförståeligt sätt. Kravhantering enligt traditionella projekt ser annorlunda ut (Elshandidy & Mazen 2013). Kraven är fördefinierade och nedskrivna i en kravspecifikation innan utvecklingen börjar. Kunden vet vad den vill ha och utvecklarna förstår på ett korrekt och klart sett vad systemet ska göra. Detta anses viktigt eftersom det finns en vilja att undvika omfattande förändringar som i slutändan kan bli kostsamma. I den traditionella kravhanteringen behövs en försäkran att kravspecifikationen är uppnådd. Detta innebär att om det finns krav som inte har blivit implementerade eller ändringar som har gjorts så behöver dessa förstås och förklaras. Medan i den agila kravhanteringen sker ändringar kontinuerligt genom hela utvecklingsprocessen och därav finns det mindre behov av dokumentation (Elshandidy & Mazen 2013).

Elshandidy och Mazen (2013) framför att IF krav är en utmaning i både agila och traditionell kravhantering eftersom det finns svårigheter att integrera ofullständiga IF krav i de olika faserna av mjukvaruutvecklingen. Ramesh, Cao och Baskerville (2010) beskriver att den traditionella kravhanteringen innehåller en mer formell och detaljerad dokumentation än den agila som mer förlitar sig på verbal kommunikation. Den traditionella är mer strukturerad och kan därför uppfattas enligt Ramesh, Cao och Baskerville (2010) som ett mer pålitligt alternativ än den agila kravhanteringen.

Flera forskare argumenterar att IF krav bortprioriteras av FK i praktiken (Karhapää et al. 2021; Alsaqaf, Daneva & Wieringa 2019; Behutiye et al. 2017) och området har därför blivit ett aktuellt och högst relevant forskningsområde som, trots att mycket referegranskad forskning finns, har kunskapsluckor. Alsaqaf, Daneva och Wieringa (2019) har fastslagit 15 utmaningar som finns med att implementera IF krav i agila mjukvaruutvecklingsprojekt, baserat på systematiska litteraturundersökningar och en utforskande fallstudie. Dessa är bland annat minimal dokumentation, för få möjligheter att revidera IF krav med kunden, kraven är

inte tillräckligt tydligt definierade, misskommunikation av kraven som leder till individuella uppfattningar och gissningar, med flera.

Alsaqaf, Daneva och Wieringa (2019) beskriver att det även finns utmaningar med traditionella mjukvaruutvecklingsprojekt såsom sen upptäckt och insamling av IF krav, implementation av IF krav, brist på synlighet av IF krav samt arkitektoniska utmaningar. Alsaqaf, Daneva och Wieringa (2019) framför att det råder mer fördefinierade tillvägagångssätt för hur kunskapsöverföring och dokumentation ska hanteras i den traditionella kravhanteringen. Genom att använda standarder, checklistor och en fördefinierad terminologi för definition av IF krav kan det underlätta för nya teammedlemmar att komma in i arbetet. Medan en stor andel av forskningen är fokuserad på utmaningar med IF krav i agila projekt, menar Alsaqaf, Daneva och Wieringa (2019) att det inte finns tillräckligt mycket forskning som framhäver och konkretiserar just hur *dokumentationen* av IF krav bör hanteras, utan att nyansera skillnaderna mellan traditionella som agila projekt, för att lyckas kommunicera dem från kund till kod.

En missvisande indikation som ter sig i agila projekt är att dokumentationen ska vara minimal, vilket ofta missuppfattas som att dokumentationen är onödig eller knappt behövs (Behutiye et al. 2020a). Dokumentationen är relaterad till en av de mest förekommande utmaningarna med IF krav, nämligen att IF krav är svårdefinierat och därför inte dokumenteras tillräckligt väl, vilket slutligen leder till att IF krav bortprioriteras vid implementationen (Alsaqaf, Daneva & Wieringa 2019). Behutiye, Seppänen, Rodríguez och Oivo (2020b) beskriver att utebliven och otillräcklig dokumentation av IF krav bidrar till teknisk skuld och långsiktigt högre underhållningskostnader. Behutiye et al. (2020a) påtalar att illa hantering av IF krav kan resultera i bristfälliga system som kräver revideringar. Teknisk skuld innebär exempelvis problem med försummade krav och ofullständiga artefakter på grund av påskyndande av leveranser till kund. Detta stärker det som Ali et al. (2022) hävdar: tidigare forskning har påvisat att ospecificerade och uteblivna IF krav leder till merkostnader. Martins (2019) påpekar att det råder brist på uppmärksamhet till de IF kraven inom den agil mjukvaruutveckling just eftersom de är dåligt definierade. Behutiye et al. (2020a) menar att även om det finns viss forskning som undersöker dokumentation av IF krav, finns det fortfarande kunskapsklyftor att fylla inom det området.

Flera författare menar att kunskap om IF krav är så kallad tyst kunskap, det vill säga kunskap som är svår att uttrycka och förmedla samt att man förlitar sig på den tysta kunskapen när det kommer till IF krav (Chazette, Brunotte & Speith 2022; Alsaqaf, Daneva & Wieringa 2019; Behutiye et al. 2017). Detta gör det svårt att konkretisera den tillgängliga kunskapen om fenomenet, vilket kan förklara varför det saknas standarder för att dokumentera dem. Behutiye et al. (2022) menar att genom att förstå yrkesverksammas uppfattning av att dokumentera IF krav idag, kan det beskriva deras förhållningssätt till dokumentation av IF krav. Uppfattningen av IF krav påverkar hur IF krav hanteras (Behutiye et al. 2020b). I vissa fall är det kundens val att bortprioritera IF krav, vilket också grundar sig i okunskap om vikten av IF krav i slutprodukten (Alsaqaf, Daneva & Wieringa 2019; Behutiye et al. 2017).

1.4 Problemformulering

Forskning inom kravhantering och de FK och IF kraven i mjukvaruutveckling har vuxit fram sedan 1970-talet och ökat i sökfrequens på Google Books Ngrams Viewer fram till cirka år 2011, då plötsligt färre referegranskade artiklar publicerades än tidigare. IF krav har å andra sidan endast fortsatt att öka i sökfrequens sedan år 1974, men är idag ändå det område som har avsevärt minst relaterade träffar i Google Ngrams Viewer (u.å.). Google Ngram är en sökmotor som visar statistik över antal publicerade böcker relaterade till de angivna sökorden. Google Ngram står för 4 % av alla världens publicerade böcker i digitalt format (Michel et al. 2011).

Enligt tidigare forskning finns det inget allmänt accepterat tillvägagångssätt för att samla in, dokumentera och analysera IF krav. Dagens mest förekommande kunskap om IF krav som används är den så kallade tysta kunskapen. Det sägs att utvecklarna är de som besitter den tysta kunskapen som är både svår att definiera och föra vidare då den är baserad på utvecklarnas erfarenhet (Eckhardt, Vogelsand & Fernández 2016). Detta gör det svårt att förstå den kunskap som idag förekommer om IF krav och för att kunna dokumentera de IF kraven (Chazette, Brunotte & Speith 2022).

Mycket av tidigare forskningen kring IF krav är kvalitativ och har undersökt specifika fall på djupet. Genom tidigare kvalitativa forskning påvisades olika mönster och koncept kring utebliven eller felaktig dokumentation av IF krav som förekommer framför allt i agila projekt. Utmaningarna med dokumentation av IF krav förekommer även inom de traditionella projekt, enligt tidigare kvalitativa studier (Alsaqaf, Daneva & Wieringa 2019). Det har visat sig att inom tidigare forskningen finns det kunskapsklyftor kring dokumentation av IF krav såsom tysta kunskap, bristfällig definition av IF krav och bristfällig kunskap om IF krav (Behutiye et al. 2020a). Dessa klyftor har blivit en motiverande faktor för att undersöka dokumentationen av IF krav med en kvantitativ metod för att kunna ta fram generaliserbar kunskap och kunna uppmärksamma olika faktorer som behöver förbättras eller ändras kring dokumentation av IF krav och dess användning inom svenska mjukvaruutvecklingsprojekt. Enligt tidigare studier finns det mycket utrymme för mer forskning inom IF krav och flera källor bekräftar att det inte finns någon tydlig standard att samla in, dokumentera och verifiera IF krav (Behutiye et al. 2022). På grund av den tysta kunskapen som IF krav baseras på, saknas det dessutom forskning av vem, eller vilka kompetenser, som krävs för att dokumentera och kvalitetssäkra de IF kraven.

Därför kommer denna studie att uppmärksamma olika faktorer som finns kring dokumentation av IF krav inom svenska mjukvaruutvecklingsprojekt för att kunna bidra till en mer statistisk generaliserbar kunskap om IF krav.

Studien avser även att undersöka hur väl dokumentationen av IF krav går till och vem som kan vara ansvarig för att kvalitetssäkra de IF krav inom mjukvaruutvecklingsprojekt. För att kunna genomföra en sådan undersökning har information samlats in kring olika dokumentationsmetoder av IF krav och dess funktionalitet. Vidare beskriver studien vem som kan ha ansvarsrollen för IF krav och dess dokumentation för att kunna avgöra vilken kompetens borde en ansvarig besitta för att kvalitetssäkra de IF kraven.

1.5 Syfte och forskningsfrågor

Syftet med denna studie är att undersöka dokumentationen av IF krav inom svenska mjukvaruutvecklingsprojekt, påvisa vikten av dokumentation av IF krav samt uppmärksamma kompetensbehovet för vem som kan och är mest lämpad för att ansvara och kvalitetssäkra de IF kraven. Detta leder fram till följande huvudfråga (FF1) och delfråga (FF2):

- FF1: Hur väl dokumenteras icke-funktionella krav inom svenska mjukvaruutvecklingsprojekt?
- FF2: Vem kan ansvara för att kvalitetssäkra icke-funktionella krav i mjukvaruutvecklingsprojekt?

1.6 Målgrupp

Uppsatsen riktar sig till de som arbetar inom svenska mjukvaruutvecklingsprojekt, framför allt kravanalytiker, Business Analyst, projektledare, verksamhetsanalytiker, kravadministratörer, systemägare, systemutvecklare, produktägare, arkitekter och beställare som framförallt dokumenterar både FK och IF krav. Vidare riktar sig denna studie till studenter inom data- och informationsvetenskap som är intresserade av kravhantering, IF krav samt tillvägagångssätt för att dokumentera IF krav.

1.7 Avgränsning

Det finns tidigare studier som påvisar vilka utmaningar som finns med IF krav och att det kan vara den avgörande orsaken till att ett projekt misslyckas. Denna studie begränsas till vem som är ansvarig för att kvalitetssäkra de IF kraven och hur väl de IF krav dokumenteras idag utifrån forskning och empiri. Empiri som samlades under denna studie täcker en väldigt bred grupp av respondenter som arbetar inom mjukvaruutvecklingsprojekt med dokumentation av IF krav inom båda mindre, stora och medelstora företag inom Sverige.

Vi har fokuserat på att mäta hur olika tillvägagångssätt att dokumentera påverkar hur väl IF krav dokumenteras genom att jämföra dess karaktär och utformning. Därmed har vi inte separerat dokumentationstekniker från dokumentationsmetoder eller verktyg, utan analyserat jämlikt utifrån vad de erbjuder. Det gemensamma begreppet är tillvägagångssättet.

Det är viktigt att notera att denna studie inte jämfört effektiviteten mellan traditionella och agila perspektiv av SDLC. Inom både traditionella och agila perspektiv är kravhanteringsfasen en av de mest betydelsefulla inom SDLC (Ochodek & Kopczyński 2018).

2 Metod

I följande kapitel presenteras studiens forskningsstrategi i syfte att besvara studiens forskningsfrågor. Metoderna som används för att genomföra studien kommer att beskrivas och ge läsaren insikt i studiens litteraturstudie och den kvantitativa undersökningen. Kapitlet avslutas med reflektion över vald metod.

2.1 Forskningsstrategi

Forskaren står inför en pragmatisk inställning där pragmatiken ligger i att det är omöjligt att välja den bästa metoden för att relatera teori och verkligheten till varandra. Den problemlösande process som forskaren utför kontinuerligt gör forskningen till en kombination av induktion och deduktion (Jacobsen 2017).

Induktion är en av de tre forskningsstrategierna, utöver deduktion och abduktion. I alla tre strategier arbetar forskaren med att relatera teori och empiri till varandra. Induktion innebär att forskaren studerar forskningsobjektet utan att först djupdyka i tidigare teorier. Därefter, utifrån den insamlade informationen, kommer forskaren att formulera sin egen teori om verkligheten. Deduktion innebär att forskaren drar sina slutsatser om enskilda fall utifrån befintliga teorier och principer. Vid det deduktiva arbetet är det teorin som bestämmer vilken empiri som ska samlas in och hur den ska tolkas och relateras till den befintliga teorin (Patel & Davidson 2019). Abduktion innebär en kombination av induktion och deduktion (Jacobsen 2017) och innebär att det enskilda fallet studeras utifrån en tidigare vedertagen teori, induktivt, och därefter prövas teori på nya fall för att sedan utveckla teori för att bli mer generell teori, deduktion (Patel & Davidson 2019).

Att arbeta strikt induktiv eller deduktiv anses vara problematiskt eftersom det är omöjligt att enbart förhålla sig till teori enligt Jacobsen (2017) och dessutom kan forskaren fastna i hög grad om forskaren inte är tillräckligt vidsynt, enligt Patel och Davidson (2019). Utgångspunkten till att abduktion valdes för denna studie är att allt vetenskapligt tänkande börjar med observation (Jacobsen 2017). Först studerades befintliga teorier om IF krav och dess dokumentation för att härleda dessa befintliga utmaningar och dra slutsatser kring problematiken med dokumentation av IF krav inom mjukvaruutvecklingsprojekt. Sedan utfördes en undersökning med frågeformulär utifrån tidigare forskning och allmänna teorier. Ett enskilt fall har studerats, det vill säga dokumentation av IF krav inom svenska mjukvaruutvecklingsprojekt, därmed har en mer generell kunskap utvecklats där kompetensbehovet för att kvalitetssäkra IF krav beskrivits. Ambitionen med en undersökning av dokumentation av IF krav var att visa kompetensbehovet för att kvalitetssäkra IF krav och dess dokumentation inom svenska mjukvaruutvecklingsprojekt istället för utveckling av en mer generell teori eller slutsatser om enskilda fall. Sålunda kunde inte denna studie genomföras med rent deduktiva eller induktiva arbetssätt. Därför är abduktion det bästa arbetssättet som anses vara passande studiens syfte och frågeställningar.

Med en deduktiv ansats finns det risk att studien endast förhåller sig till teorin om dokumentationen av IF krav, vilket skulle kunna leda till att frågorna blir för teoretiska och svåra att besvara. Vid en induktiv ansats finns det risk att studien inte känner till förhållanden mellan olika parametrar inom dokumentationen av IF krav och dess problematik kring fördelade ansvarsroller, och missar därför att fånga upp viktiga detaljer vid

enkätundersökningen (Jacobsen 2017). Risken med att arbeta induktivt innebär då att forskaren oftast omedveten väljer studieobjekt utifrån tidigare erfarenheter och kan formulera teori som utesluter andra alternativa tolkningar. Dessutom kan det abduktiva arbetssättet verifieras i deduktiva fasen (Patel & Davidson 2019). Med hänsyn till dessa risker har det abduktiva arbetssättet använts. Genom att ha en abduktiv ansats syftar denna studie till att bredda förståelse av IF krav, samtidigt som att visa det aktuella problemet med dokumentation av IF krav inom svenska mjukvaruutvecklingsprojekt.

2.2 Kvantitativ metod med kvalitativt perspektiv

En mixed methods-ansats har tillämpats där kvalitativ och kvantitativ metodik har kombinerats för att uppnå en fördjupad och omfattande förståelse av vårt forskningsområde. Mixed method är en metod som kräver mer tid då två olika typer av data ska samlas in och analyseras i en och samma studie, det vill säga både kvalitativa och kvantitativa data. Genom att kombinera dessa två metoder kan mer värde och förståelse skapas (McKim 2017; Cronholm & Hjalmarsson 2021) vilket även Eliasson (2013) hänvisar till att syftet med denna kombinationsmetod är att täcka olika perspektiv och infallsvinklar genom att samla in och analysera olika typer av information.

Vi började med en kvalitativ litteraturanlys eftersom teorin och litteraturen inom området främst har varit kvalitativt orienterad. En litteraturstudie har genomförts i form av läsning och tolkning av vetenskapliga artiklar för att öka kunskapen om IF krav och dess dokumentation samt för att uppnå en översiktlig förståelse för området inför enkätundersökning, då förberedelserna enligt Eliasson (2013) är viktiga vid kvantitativ undersökning. Enligt Jacobsen (2017) är den kvantitativa metoden mest lämplig när förkunskaper om fenomenet som undersöks redan finns, det vill säga förståelse för dokumentation, kvalitetssäkring och ansvarsroller för IF krav samt när problemställning är relativt klar.

Genom sekundärdata – data som har samlats in och analyserats av andra (Eliasson 2013) – i form av intervjuer och observationer som har utförts i tidigare forskning har vi förstått befintliga perspektiv, erfarenheter och åsikter kring IF krav på ett djupgående sätt. Kvalitativ metod anses vara den rätta metoden när det gäller att komma åt sammanhang som kräver förståelse (Eliasson 2013). Genom denna process har vi kunnat identifiera viktiga teman, mönster och nyanser som inte skulle ha varit möjliga att fånga med enbart kvantitativ metod, då kvantitativa studier har saknats och inte heller erbjuder samma djupgående förståelse.

Efter att ha genomfört den kvalitativa litteraturanalysen har vi använt den insamlade kunskapen som grund för att utforma vår webbaserade enkät. Vi har skapat en enkätundersökning och samlat in numerisk data från en större population. Numerisk data innebär information som samlas in i form av siffror (Jacobsen 2017). Genom att tillämpa statistiska metoder har vi kunnat kvantifiera resultaten och utforska samband och mönster på en mer generaliserbar nivå. Denna kvantitativa empiri har kompletterat den kvalitativa litteraturanalysen genom att ge oss möjlighet att dra mer övergripande slutsatser och göra jämförelser på en bredare skala, det vill säga jämförelser mellan olika litteraturstudien och enkätundersökningen. Enligt Jacobsen (2017) är den kvantitativa metoden mest lämplig när det finns förkunskaper om det undersökta fenomenet, det vill säga dokumentationen av IF krav, och när problemställningen är relativt tydlig. Vidare har olika problemområden identifierats genom enkätundersökningen, vilket har kompletterats med ytterligare litteraturundersökning.

Genom att använda mixed methods-konceptet har vi kunnat dra nytta av både kvalitativ och kvantitativ metodik för att få en mer mångfacetterad och holistisk förståelse av vårt forskningsområde. Genom att kombinera teorin och litteraturen inom kvalitativ forskning med den empiriska kvantitativa analysen har vi kunnat utforska forskningsfrågorna på ett mer komplett sätt och erhålla nya insikter i det fenomen vi studerat, särskilt eftersom det för närvarande inte finns tillräckligt med kvantitativa studier inom ämnet (Behutiye et al. 2020a).

2.3 Litteraturstudie

I följande avsnitt presenteras hur de använda artiklarna samlades in, presentation av artiklarna, litteraturanalysen som genomförts samt vilka teman som genererades till det teoretiska ramverket.

2.3.1 Artikelsamling

Inledningsvis genomfördes en grundlig genomgång av nuvarande forskning kring dokumentation och uppfattning av IF krav. Insamlingen påbörjades tidigt av relevanta artiklar genom att söka i *Senior Scholars Basket of Journals* vilket enligt Recker (2021) är de högst rankade tidskrifterna inom informatik. Under insamling av artiklar, har vi lagt vikt på att artikeln ska handla om dokumentationen och uppfattningen av IF krav inom antingen traditionella eller agila mjukvaruutvecklingsprojekt. Vi syftade till att granska hanteringen av IF krav i olika mjukvaruutvecklingsprojekt där både metoderna tillämpas. Vi ville granska dem likvärdigt och se dokumentationsteknikerna för sig, utan att relatera dem till resterande delar av mjukvaruutvecklingsprojektet som är antingen traditionellt eller agilt. Vi använde olika sökorden för att hitta relevanta artiklar. Sökorden inkluderades i fyra olika sökningar i varje tidskrift:

- “Documentation” + “Non-functional requirements”
- “Documentation” + “Quality requirements”
- “Documentation” + “Extra functional requirement”
- “Documentation” + "Requirements engineering"

Endast en artikel relaterad till IF krav och dokumentation med sökordet “Documentation” + "Requirements engineering" valdes ur denna iteration (se Tabell 1).

Vid en andra iteration syftade vi att öka resultat chanserna och använde Högskolan i Borås bibliotekstjänst Primo som tillhandahåller fler databaser och tidskrifter. Inkluderingskriterier var att alla artiklar och konferensrapporter skulle vara i fulltext och expertgranskade. Årtalen begränsades till 2007-2023. Olika sökord användes i olika kombinationer vilket resulterade i ett brett urval av relevanta artiklar (se Tabell 1). De sökorden som användes mest var:

- “Requirement engineering”
- "Non-functional requirements”
- “Quality requirements”
- “Software development”
- “Documentation”
- “Agile”
- “Traditional”
- "Requirement specification”
- “Quality”
- “Software requirements engineering”

För att specificera sökningen till just dessa ord användes dessa termer som ämnesord. I urvalsprocessen identifierades relevanta artiklar efter att ha granskat dess titel, abstrakt och slutsats och därefter kunde dess relevans avgöras. Sammanlagt lästes 76 artiklar varav de artiklar som hade blivit citerade mest av andra författare användes. De artiklar som valdes ut till litteraturstudien innehöll beskrivningar av IF krav; hur de dokumenteras och kvalitetssäkras samt relevant information om ansvarsroller för kvalitetssäkring. För att få tillgång till relevant information valdes flertalet av artiklarna med utgivningsår från 2017 och framåt. Tre artiklar valdes ut med utgivningsår från 2010-2016. Dessa artiklar valdes eftersom de innehöll definierade beskrivningar av IF krav, vilket var en viktig del av litteraturstudien.

Under en tredje iteration användes Google Scholar som sökmotor för att hitta relevanta artiklar. Flertalet av artiklarna som hittades i Högskolan i Borås bibliotekstjänst Primo återfanns även i Google Scholar. Resultatet av sökningen "Non-Functional Requirements" + "Documentation" resulterade i 26800 artiklar. På grund av en begränsad tidsram ingick endast artiklar från Google Scholar med över 1000 citeringar i urvalsprocessen. Sammanlagt valdes 13 artiklar till grund för litteraturstudien (se Tabell 2)

Sökord	Databas	Antal träffar	Valda artiklar
Requirements engineering + Documentation	Information system journal	8	1
Requirement engineering + Quality requirements + Agile	Primo	57	3
Quality requirements + Documentation	Primo	37	4
Requirements specification + Agile software development	Primo	19	1
Non-functional requirement + Software development + Documentation	Primo	6	3
"Non-Functional Requirements" + "Documentation"	Google Scholar	26800	1

Tabell 1: Söktjänster och sökord som användes

2.3.2 Valda Artiklar

Nr	Titel	Författare/ Årtal	Databas/ Citeringar	Utgivare
1	Quality Requirement Documentation Guidelines for Agile Software Development	Behutiye, Rodríguez & Oivo (2022)	IEEE Access Antal citeringar: 1	IEEE
2	A Qualitative Study on Non-Functional Requirements in Agile Software Development	Jarzębowski & Weichbroth (2021)	IEEE Access Antal citeringar: 19	IEEE
3	Strategies to manage quality requirements in agile software development: a multiple case study	Karhapää et al. (2021)	ProQuest Antal citeringar: 22	Empirical software engineering :

				an international journal
4	Documentation of Quality Requirements in Agile Software Development	Behutiye, Seppänen, Rodríguez & Oivo (2020b)	ACM Digital Library Antal citeringar: 16 (Konferensrapport)	Proceedings of the Evaluation and Assessment in Software Engineering
5	Quality requirements challenges in the context of large-scale distributed agile: An empirical study	Alsaqaf, Daneva & Wieringa (2019)	ScienceDirect Antal citeringar: 84	Information and Software Technology
6	Quality of software requirements specification in agile projects: A cross-case analysis of six companies	Medeiros, Vasconcelos, Silva & Goulão (2018)	ScienceDirect Antal citeringar: 43	The Journal of systems and software,
7	Non-functional Requirements Documentation in Agile Software Development: Challenges and Solution Proposal	Behutiye, Karhapää, Costal, Oivo & Franch. (2017)	ProQuest Antal citeringar: 63 (Konferensrapport)	arXiv.org
8	Are “Non-functional” Requirements really Non-functional?	Eckhardt, Vogelsang & Fernández (2016)	IEEE Antal citeringar: 135 (Konferensrapport)	2016 IEEE/ACM International Conference on Software Engineering (ICSE)
9	How do Software Architects Consider Non-functional Requirements: An Exploratory Study	Ameller, Ayala, Cabot & Franch (2012)	IEEE Access Antal citeringar: 137	IEEE International Requirements Engineering Conference (RE)
10	On Non-Functional Requirements	Glinz (2007)	IEEE Access Antal citeringar: 1014	RE
11	Agile requirements engineering practices and challenges: an empirical study	Ramesh, Cao, Baskerville (2010)	Wiley Online Library Antal citeringar: 489	Information Systems Journal
12	Towards optimal quality requirement documentation in agile software development: A multiple case study	Behutiye et al. (2022)	ScienceDirect Antal citeringar: 7	Journal of Systems and Software
13	Role of Non-functional Requirements in projects' success	Ali, Khalil, Ahmad, Parveen & Zaman (2022)	IEEE Antal citeringar: 0 (Konferensrapport)	Conference on Digital Futures and Transformative Technologies (ICoDT2)

Tabell 2: Utvalda artiklar till litteraturstudie

2.3.3 Presentation av valda artiklar

De 13 artiklarna som valdes till denna studie (se Tabell 2) behandlar ämnen såsom dokumentation av IF krav, kvalitetssäkring, uppfattningar om, ansvar för och utmaningar inför IF krav samt roller. Varje artikel hanterar flera av dessa ämnen vilket resulterar i olika perspektiv från olika författare. Flera av artiklarna är skrivna av samma författare som anses kända inom sitt forskningsområde.

"Quality Requirement Documentation Guidelines for Agile Software Development" (Behutiye et al. 2022) och *"Documentation of Quality Requirements in Agile Software Development"* (Behutiye et al. 2020b) - Båda dessa artiklar behandlar dokumentation av kvalitetskrav inom agil mjukvaruutveckling och diskuterar strategier och utmaningar för att hantera dessa krav.

Artiklarna *"Quality of software requirements specification in agile projects: A cross-case analysis of six companies"* av Medeiros et al. 2018 och *"Non-functional Requirements Documentation in Agile Software Development: Challenges and Solution Proposal"* (Behutiye et al. 2017) tar upp betydelsen av erfarenhet och tyst kunskap för dokumentation av IF krav samt utmaningar som kan uppstå vid dokumentation. Deras slutsats är att den tysta kunskapen prioriteras före dokumentation, vilket flertalet av artiklarna även tar upp, att när det finns erfarenhet förekommer mindre dokumentation.

"Towards optimal quality requirement documentation in agile software development: A multiple case study" (Behutiye et al. 2022) och *"Strategies to manage quality requirements in agile software development: a multiple case study"* (Karhapää et al. 2021) - Båda dessa artiklar fokuserar på kvalitetskrav och dokumentation i agila projekt samt utforskar utmaningar och strategier för att hantera dessa krav. Artikeln *"Strategies to manage quality requirements in agile software development: a multiple case study"* av Karhapää et al. (2021) fokuserar på utmaningar med IF krav och betonar att utvecklarna behöver förstå "varför" bakom dessa krav för att förstå deras värde. Författarnas diskussion om oklarheter och förvirring kring IF krav överlappar med andra artiklar som också diskuterar svårigheter med att hantera och tolka IF krav.

"A Qualitative Study on Non-Functional Requirements in Agile Software Development" av Jarzębowicz och Weichbroth (2021) samt *"Are 'Non-functional' Requirements really Non-functional?"* av Eckhardt, Vogelsang och Fernández (2016): Dessa två artiklar undersöker allmänna uppfattningar om IF krav och hur de uppfattas inom mjukvaruutveckling. De överlappar genom att båda utforskar det bredare perspektivet av hur utvecklare och praktiker ser på IF krav, vilket ger en nyanserad förståelse av IF kravens roll och uppfattning.

"Quality requirements challenges in the context of large-scale distributed agile: An empirical study" av Alsaqaf, Daneva och Wieringa (2019): Denna artikel diskuterar 15 utmaningar relaterade till IF krav och tar upp olika aspekter som är relevanta för kravhanteringsprocessen, inklusive dokumentation, kvalitetssäkring, kommunikation och roller.

"Role of Non-functional Requirements in Projects' Success" av Ali et al. (2022): Denna kvantitativa studie utforskar IF kravens betydelse och deras påverkan på projektframgång.

"How do Software Architects Consider Non-functional Requirements: An Exploratory Study" av Ameller, Ayala, Cabot och Franch (2012). Författarna lägger stor fokus på att det är mjukvaruarkitekten som bör ha ansvar för de IF kraven eftersom de besitter helhetsperspektiv och har teknisk kompetens.

Artikeln *"Agile requirements engineering practices and challenges: an empirical study"* av Ramesh, Cao och Baskerville (2010) fokuserar delvis på IF krav och dokumentation och tar även upp utmaningar med krav överlag.

Artikeln *"On Non-Functional Requirements"* av Glinz (2007) utforskar IF krav och deras roll inom mjukvaruutveckling. Den analyserar vikten av att behandla IF krav korrekt och ger insikter i utmaningar och strategier för att hantera dem på ett effektivt sätt.

2.3.4 Motiv till valda artiklar

"Quality Requirement Documentation Guidelines for Agile Software Development" av Behutiye, Rodríguez och Oivo (2022) valdes på grund av dess riktlinjer för bättre dokumentationsstrategier för IF krav. Artikeln erbjuder konkreta rekommendationer som kan hjälpa till att förbättra dokumentationsprocessen och därigenom öka förståelsen för och betydelsen av dessa krav i mjukvaruutveckling.

"Documentation of Quality Requirements in Agile Software Development" av Behutiye et al. (2020b) valdes för dess fokus på praktiker inom agil mjukvaruutveckling och hur dessa påverkar dokumentationen av IF krav. Genom att undersöka sambandet mellan erfarenhet och dokumentationsbehov ger artikeln insikter i hur olika faktorer kan påverka dokumentationen av IF krav.

Artikeln *"Quality of software requirements specification in agile projects: A cross-case analysis of six companies"* av Medeiros, Vasconcelos, Silva och Goulão (2018) inkluderades för att den belyser behovet av erfarenhet för att framgångsrikt dokumentera IF krav. Dess analys av faktorer som påverkar kravhantering ger en bredare förståelse för dokumentationsprocessen och dess utmaningar.

Artikeln *"Non-functional Requirements Documentation in Agile Software Development: Challenges and Solution Proposal"* av Behutiye et al. (2017) bidrar med insikter om olika dokumentationsmetoder inom olika företag och betonar vikten av erfarenhet samt tyst kunskap för att hantera IF krav. Dess kvalitativa ansats ger djupgående förståelse för de utmaningar som kan uppstå vid dokumentation.

"Towards optimal quality requirement documentation in agile software development: A multiple case study" av Behutiye et al. (2022) betonar dokumentationens roll i att säkerställa kvalitet och adresserar kommunikationsgap som kan uppstå vid IF krav. Genom att undersöka flera fallstudier ger artikeln en omfattande bild av utmaningarna med dokumentationen av IF krav samt strategier inom mjukvaruutveckling.

Motivet till att inkludera *"A Qualitative Study on Non-Functional Requirements in Agile Software Development"* av Jarzębowicz och Weichbroth, (2021) och *"Are 'Non-functional' Requirements really Non-functional?"* av Eckhardt et al. (2016) är att dessa artiklar utforskar och analyserar allmänna uppfattningar om IF krav inom mjukvaruutveckling. Genom att

undersöka hur IF krav uppfattas och hanteras av aktörer inom detta område, kan dessa artiklar ge värdefulla insikter om hur sådana krav påverkar utvecklingsprocessen och projektframgångarna.

"Strategies to manage quality requirements in agile software development: a multiple case study" av Karhapää et al. (2021): Denna artikel valdes på grund av dess fokus på utmaningar med IF krav och hur utvecklarna behöver förstå "varför" bakom dessa krav. Artikeln erbjuder en alternativ synvinkel på hanteringen av IF krav och hur förvirring och oklarheter kan uppstå.

"Quality requirements challenges in the context of large-scale distributed agile: An empirical study" av Alsaqaf, Daneva och Wieringa (2019): Artikeln valdes för dess fokus på utmaningar relaterade till IF krav och dess utforskning av olika aspekter som är viktiga för kravhanteringsprocessen. Genom att identifiera och diskutera dessa utmaningar ger artikeln en djupgående inblick i svårigheter som kan uppstå vid hantering av sådana krav.

"Role of Non-functional Requirements in Projects' Success" av Ali et al. (2022): Denna artikel inkluderades på grund av dess kvantitativa ansats för att utforska betydelsen av IF krav och deras påverkan på projektframgång. Genom att mäta och analysera dessa aspekter ger artikeln en empirisk grund för att förstå IF kravens roll i projektets utfall.

"How do Software Architects Consider Non-functional Requirements: An Exploratory Study" av Ameller, Ayala, Cabot och Franch (2012): Denna artikel valdes för att den undersöker hur mjukvaruarkitekter betraktar IF krav. Genom att betona att mjukvaruarkitekter har en viktig roll i att hantera dessa krav på grund av deras helhetsperspektiv och tekniska kompetens, ger artikeln insikter i hur ansvar och förståelse för IF krav fördelas inom mjukvaruutvecklingsprocessen. Genom att utforska arkitekters perspektiv på IF krav kan artikeln ge en djupare insikt i hur dessa krav hanteras och integreras i mjukvaruutveckling.

"Agile requirements engineering practices and challenges: an empirical study" av Ramesh, Cao och Baskerville (2010): Denna artikel inkluderades på grund av dess delvisa fokus på IF krav och dokumentation samt dess diskussion om utmaningar med kravhantering generellt. Genom att ta upp utmaningar som kan uppstå vid hantering och dokumentation av krav ger artikeln insikter i hur dessa utmaningar kan påverka mjukvaruutveckling. Författarnas kvalitativa studie utforskar hur IF krav hanteras och de svårigheter som de IF kraven kan möta.

Trots att artikeln *"On Non-Functional Requirements"* av Glinz (2007) är äldre har den ackumulerat en betydande mängd citeringar (1014) från forskarsamhället. Denna höga citeringsfrekvens indikerar att artikeln har haft en påverkan på forskning inom området IF krav och att den har blivit en referenspunkt för fortsatt diskussion och studier. Genom att inkludera denna artikel i litteraturstudien kan den bidra med en historisk och referensbaserad förståelse av hur synen på och förståelsen av IF krav har utvecklats över tiden och hur de har påverkat mjukvaruutvecklingen.

2.3.5 Litteraturanalys

Insamlingen av artiklar resulterade i 13 artiklar där varje artikel lästes igenom för att skapa en uppfattning om vad artikeln handlade om men även för att hitta relevanta koncept som sedan

kunde specificeras i det teoretiska ramverket. Varje person i gruppen ansvarade för att läsa igenom cirka fyra artiklar var och skriva ner och färgmarkera viktigt innehåll ur dessa artiklar. Detta gjordes för att kunna få en översikt på relevant information som fanns i varje artikel och som sedan kunde ligga till grund när teoridelen skulle skrivas. För att alla skulle få en mindre motstridig uppfattning läste alla gruppmedlemmar igenom varje artikel för att se om samma tolkning av informationen fanns kring innehållet i artiklarna.

Litteraturanalyser innebär enligt Webster och Watson (2022) att hitta gemensamma koncept och mönster utifrån flera olika källor. De 13 artiklarna som användes till denna studie hade gemensamma koncept kopplat till våra forskningsfrågor. Varje artikel lästes igenom för att kunna urskilja vilka olika tänkbara teman varje artikel kunde ha.

2.3.6 Teman

De fyra teman som presenterades i det teoretiska ramverket utgjorde underlaget för att kartlägga och ringa in vårt forskningsområde och problembilden med IF krav. Temana identifierades efter en kvalitativ litteraturanlys av litteraturen relaterat till våra forskningsfrågor och omfattade:

- Utmaningar med att identifiera IF krav
- Dokumentationens betydelse för IF krav
- Faktorer som påverkar kvalitetssäkring av IF krav
- Ansvarsroller för att kvalitetssäkra IF krav

Här nedan beskrivs vilka artiklar som tillhörde vilket tema.

Tema →	Utmaningar med att identifiera IF krav	Dokumentationens betydelse för IF krav	Faktorer som påverkar Kvalitetssäkring av IF krav	Ansvarsroller för att kvalitetssäkra IF krav
Artikel				
1		x	x	
2	x	x		
3	x	x	x	
4	x	x		x
5	x	x	x	x
6		x		
7	x	x		
8	x	x		
9		x		x
10		x		

11	x	x	x	
12	x	x	x	x
13	x			

Tabell 3: Teman kopplat till respektive artikel

Teman framtoqs för att kunna bevisa och kartlägga det teoretiska ramverket som behövdess för att kunna analysera resultatet från enkätundersökningen. Eftersom IF krav är i sig tvetydigt och ofta överskuggas av FK ansåg vi att detta behöver noteras i teorin, därmed temat "Utmaningar med att identifiera IF krav". Forskningen visar att den så kallade tysta kunskap dominerar över skriven dokumentation och att den är svår att föra vidare därmed ansåg vi att ha med ett avsnitt om "(Tyst) Kunskap om IF krav" för att tydliggöra vad detta begrepp innebär och vilka utmaningar den tysta kunskapen bidrar till. Temat "Dokumentationens betydelse för IF krav" och underrubriken "Tillvägagångssätt för att dokumentera IF krav" täcker problematiken kring att dokumentera IF krav samt tillvägagångssätten som noterats i tidigare forskning för att dokumentera IF krav i svenska mjukvaruutvecklingsprojekt idag. Nästkommande tema "Faktorer som påverkar kvalitetssäkring av IF krav" beskriver hur IF krav kan kvalitetssäkras och vad kvalitetssäkring innebär. För att kunna besvara vår forskningsfråga behövdess det finnas ett avsnitt för vad kvalitetssäkring innebär och hur det sker på IF krav. Det sista temat "Ansvarsroller för att kvalitetssäkra och dokumentera IF krav" sammanställer vilka perspektiv på olika roller som finns i vår litteraturstudie, samt dess förmåga att kvalitetssäkra och dokumentera de IF kraven. Detta avsnitt är en viktig grund för utvärdering av resultatet av roller i svenska mjukvaruutvecklingsprojekt.

I och med att vi följt en abduktiv metod var det oklart vilken inriktning studien skulle ta. Vi studerade ämnet på bredden initialt i form av en kvalitativ litteraturstudie då de enda relevanta tidigare forskning som gjorts inom ämnet var kvalitativa. Med den kvalitativa litteraturanalysen som grund skapade vi enkätfrågorna som i syfte att utmana de teman vi identifierat i teorin. Data som samlades in från enkäten analyserades kvantitativt och genererade nya upptäckter till vår studie som diskuterades i analys och diskussion. Genom systematisk dataanalys av empirin undersökte vi huruvida den stred mot eller stöttade teorin.

2.4 Kvantitativ undersökning

I följande kapitel beskrivs urval av respondenterna, bakgrundsinformation till respondenterna, enkätundersökningen, hur datainsamlingen skedde samt hur enkäten formulerades. Kapitlet avslutas med en beskrivning av dataanalysen till den kvantitativa undersökningen.

2.4.1 Urval av respondenter

Enligt Jacobsen (2017) är en undersökning tidsbegränsad vilket kan medföra svårigheter att få ett representativt urval ifrån den generella befolkningen. Med denna studie ville vi nå ut till ett representativt urval där respondenterna arbetade med både FK och IF krav samt dokumentation av dessa. Den insamlade data syftade till att representera en bred grupp av respondenter med olika roller inom mjukvaruutvecklingsprojekt, därför blev den data som samlades in mer individualistisk (ibid).

Kriteriet för empirin var att komma i kontakt med personer vars arbete kommer i kontakt med kravhantering. Genom ett bekvämlighetsurval skickades inbjudan att delta i undersökningen via mejl, inklusive länk till enkäten, till personer i vår omgivning som arbetade med krav eller mjukvaruutveckling. Det var frivilligt att besvara enkäten, vilket påpekades i mejlet. Länken till enkäten publicerades från våra personliga konton på sociala medier såsom LinkedIn, Facebook samt i Discord-grupper vars huvudintresse var mjukvaruutveckling i Sverige.

Inför den andra iterationen av mejlutskick identifierades företag via Google och LinkedIn vars huvudsyssla var mjukvaruutveckling och hade kontor i Sverige. Dessa företag, vilka totalt blev 72 stycken, mejlades genom vardera info-mejl. I mejlet har vi specificerat vilka roller som personer borde ha som söktes inom deras företag och mottagaren ombads att vidarebefordra mailet internt till personer som kunde tänkas ingå i studiens målgrupp. I mejlet uppmanas även mottagaren att bidra till snöbollsmetoden genom att vidarebefordra mailet till personer som de personligen ansåg kunde besvara enkäten. Jacobsen (2017) beskriver snöbollsmetoden som ett sätt att få tips och idéer om vilka andra som kan vara intressanta respondenter för studien, från själva respondenten man intervjuar eller korresponderar med. I och med upprepning av metoden så rullar snöbollar eller så kallade snöbollseffekten kör igång (Jacobsen 2017).

Sammanlagt specificerade vi i mejlen vilka personer som kunde ingå i vår målgrupp och bad även att mejlet skulle vidarebefordras:

*“[...] Vi undrar om du och dina kollegor skulle vilja hjälpa oss att samla in empiri för vår undersökning genom att svara på en enkät (tar ca 5 min). **Svarspersonerna bör arbeta med med krav/icke-funktionella krav (kan men måste inte vara: IT/teknisk projektledare, arkitekter, business analysts, produktägare, applikationskonsulter, programmerare, osv).**”*

Vi måste få in minst 100 svar för att kunna använda datan, så vi vore väldigt tacksamma om du även vill vidarebefordra detta mejl till någon du tror kan svara på enkäten, eller eventuellt mejla oss deras kontaktuppgifter. [...]”

Dessa personer som arbetade för något av mjukvaruutvecklingsföretagen vi kontaktade valdes som ett godtyckligt urval, i och med deras anställning på ett mjukvaruföretag eller om enkäten nått dem via snöbollsmetoden, ansågs av andra inom samma bransch ha godtycklig kunskap om ämnet i fråga. Ett godtyckligt urval innebär att vi ansåg att de erhöll kunskaper och erfarenheter inom kravhantering. Enligt Jacobsen (2017) beskrivs ett godtyckligt urval som urval av personer som undersökarna anser vara representativa. Däremot valdes inte respondenterna ut, då massmejl skickades till relevanta företag var det inte upp till författarna att avgöra vilka som skulle svara på enkäten eller ej. Detta kan vara anledning till varför 51% av respondenterna var kvinnor, trots att tech-branschen utgörs av 30% kvinnor enligt statistik rapporterad år 2021 från SCB (TechSverige, u.å.). Det skulle kunna betyda att fler kvinnor än män inom mjukvaruutvecklingsbranschen svarade på enkäten, men är inte en slutsats. Enligt Jacobsen (2017) är detta en av nackdelarna med ett bekvämlighetsurval, eftersom nyckelpersoner som kan ha format resultaten annorlunda kanske inte medverkade. Trots detta, formades så småningom trender i resultaten och mättnad kunde uppnås vid 43 respondenter.

Respondenternas identitet utgjordes av ålder, genus, utbildning, deras arbetserfarenhet i år och storlek på företaget de jobbar på i dagsläget, se avsnitt 2.4.1 Respondenterna. Mejlutskick för båda iterationerna gjordes under sju arbetsdagar. Under tiden sparades information om de företag och personer som kontaktats i ett Excel-dokument samt genom vilket medium de kontaktas via och vem av skribenterna till uppsatsen som initierat kontakten. Detta för att kontrollera att vi nått ut till en bred publik, kunna hålla reda på vilka som hade kontaktats.

2.4.2 Respondenterna

Totalt medverkade 43 respondenter. Den överskådliga tabellen (se tabell 4) nedan visar respondenternas bakgrundsinformation i följande ordning från vänster: respondentnummer, genus (M= man, K=kvinn), ålder, utbildningsområde , högst avklarade examen (YH= Yrkehögskola), deras nuvarande roll (BA=Business Analyst), deras erfarenhet i nuvarande roll, storlek på företaget de jobbar på, angiven förståelse för IF krav på en likertskala (1=väldigt tydligt, 5= väldigt otydligt), näst sista kolumnen är vare sig de tidigare arbetat med IF krav eller inte och den sista är en sammanfattande uppfattning om deras senioritet baserat den givna bakgrundsinformationen (S=Senior, M= Mid-level, J=Junior).

Med senioritet menas till vilken grad personen i fråga har samlat på sig en betydande mängd kunskap och erfarenhet. En junior är den med minst erfarenhet och senior den med mest, mid-level är däremellan. Det finns ingen absolut standard för hur många år som krävs för att anses vara senior i en arbetsroll. Det handlar inte bara om antalet år i en specifik roll, utan också om den kompetens, expertis och ledarskap/ansvar som de har utvecklat under tiden (Heinsohn Business Technology, u.å.). Genom dessa aspekter har respondenternas senioritet analyserats och angetts i sista kolumnen i tabell 4, men är endast ett antagande på deras senioritet och inte vad de själva angett.

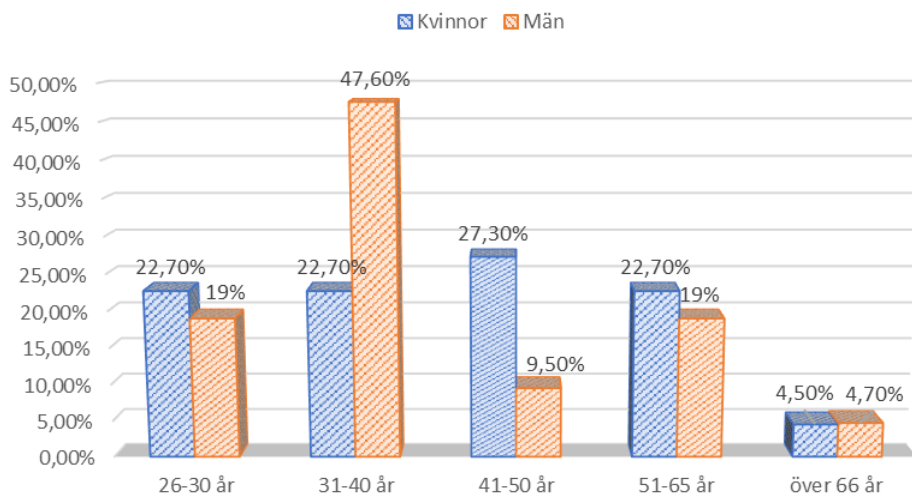
1	M	31-40 år	Data/Informatik, Ekonomi, affärsutveckling, marknadsföring	Kandidat	BA, service owner	2-4 år	Medel	1	Ja	M
2	K	31-40 år	Data/Informatik	Master	Systemutvecklare	5-7 år	Stort	1	Nej	M
3	M	26-30 år	Data/Informatik	Master	Systemutvecklare	0-1 år	Stort	2	Nej	J
4	M	26-30 år	Data/Informatik	Kandidat	Applikationskonsult	0-1 år	Stort	4	Ja	J
5	M	51-65 år	Data/Informatik	Sjävlärd/kurser	Applikationskonsult	8-10 år	Stort	3	Vet ej	M
6	K	26-30 år	Data/Informatik	Master	Systemutvecklare	2-4 år	Stort	4	Ja	J
7	K	26-30 år	Data/Informatik, Teknik	YH	Systemutvecklare	2-4 år	Stort	5	Ja	J
8	K	51-65 år	Data/Informatik	Kandidat	Testingenjör	2-4 år	Stort	3	Ja	S
9	K	31-40 år	Teknik	Master	Systemutvecklare, arkitekt	8-10 år	Stort	1	Ja	S
10	K	41-50 år	Ledarskap, administration, Ekonomi, affärsutveckling, marknadsföring	Kandidat	BA, projektledare, funktionell applikationskonsult	5-7 år	Stort	1	Ja	S
11	M	51-65 år	Ekonomi, affärsutveckling, marknadsföring	Magister	Applikationskonsult	+11 år	Stort	4	Nej	S
12	M	26-30 år	Teknik	Kandidat	Applikationsspecialist	5-7 år	Stort	1	Ja	M
13	M	31-40 år	Data/Informatik	Kandidat	Systemutvecklare	5-7 år	Medel	2	Ja	M
14	M	31-40 år	Data/Informatik, Teknik	Kandidat	BA, produktägare	2-4 år	Medel	2	Ja	J
15	M	41-50 år	Ekonomi, affärsutveckling, marknadsföring	Gymnasium	Kravanalytiker	2-4 år	Medel	1	Ja	J
16	M	31-40 år	Data/Informatik	Kandidat	Kravanalytiker, BA	8-10 år	Stort	1	Ja	S
17	M	31-40 år	Data/Informatik	Kandidat	Kravanalytiker, BA	8-10 år	Stort	1	Ja	S
18	K	41-50 år	Samhällsvetenskap	Kandidat	Applikationskonsult	2-4 år	Medel	5	Nej	J
19	K	51-65 år	Data/Informatik, Ledarskap, administration	Kandidat	Projektledare	2-4 år	Medel	1	Ja	S
20	K	31-40 år	Teknik	YH	Kravanalytiker, testare	2-4 år	Stort	2	Ja	J
21	M	26-30 år	Teknik	Master	Systemutvecklare, produktägare	2-4 år	Medel	1	Ja	J
22	K	26-30 år	Data/Informatik	Kandidat	Tech Sales	0-1 år	Stort	5	Ja	J
23	K	26-30 år	Data/Informatik, Media, design	Kandidat	UX-designer	2-4 år	Medel	1	Ja	J
24	K	över 66 år	Data/Informatik, Civilingenjör	Master	Kravanalytiker, BA	Över 11 år	Micro	1	Ja	S
25	K	41-50 år	Data/Informatik	Kandidat	Systemutvecklare, arkitekt, produktägare	8-10 år	Litet	1	Ja	S
26	M	51-65 år	Ingenjör	Master	Projektledare	8-10 år	Medel	1	Ja	S

27	K	31-40 år	Data/Informatik, Media, design	Magister	UX-designer	5-7 år	Stort	2	Ja	M
28	M	31-40 år	Data/Informatik, Ekonomi, affärsutveckling, marknadsföring, Ingenjör	Master	CTO	5-7 år	Medel	1	Ja	S
29	M	31-40 år	Data/Informatik	Kandidat	Mjukvaruansvarig	0-1 år	Stort	1	Nej	S
30	K	31-40 år	Data/Informatik	YH	Systemutvecklare, UX-designer	2-4 år	Medel	5	Ja men aldrig hört terme n	J
31	K	51-65 år	Data/Informatik, Ledarskap, administration	Sjävlärd/kurser	Kravanalytiker, projektledare	Över 11 år	Micro	1	Ja	S
32	M	31-40 år	Data/Informatik, Teknik, Ledarskap, administration	YH	Produktägare	2-4 år	Stort	1	Ja	S
33	M	41-50 år	Data/Informatik	Magister	Utvecklingschef IT	5-7 år	Stort	2	Ja	S
34	M	över 66 år	Data/Informatik, Teknik, Ingenjör	Master	Systemutvecklare, kravanalytiker, BA	Över 11 år	Micro	2	Ja	S
35	K	51-65 år	Ingenjör	Gymnasium	Kravanalytiker, BA	Över 11 år	Stort	1	Ja	S
36	K	51-65 år	Data/Informatik, Ledarskap, administration, Ekonomi, affärsutveckling, marknadsföring	Sjävlärd/kurser	Verksamhetsutvecklare IT - ansvarig regelverksfrågor IT	0-1 år	Stort	1	Ja	S
37	M	31-40 år	Ingenjör	Master	Systemutvecklare	Över 11 år	Medel	3	Ja	S
38	K	26-30 år	Ingenjör	Master	Programmerare	2-4 år	Stort	4	Ja	J
39	M	31-40 år	Data/Informatik	Master	UX-designer	0-1 år	Stort	2	Ja	J
40	M	41-50 år	Data/Informatik, Ekonomi, affärsutveckling, marknadsföring	Kandidat	Kravanalytiker, BA	5-7 år	Medel	2	Ja	M
41	K	41-50 år	Ledarskap, administration, Ekonomi, affärsutveckling, marknadsföring	Sjävlärd/kurser	Kravanalytiker, BA	Över 11 år	Micro	1	Ja	S
42	K	41-50 år	Data/Informatik, Ingenjör	Master	Systemutvecklare	Över 11 år	Stort	1	Ja	S
43	M	51-65 år	Data/Informatik, Ingenjör	Master	Arkitekt	Över 11 år	Medel	5	Ja	S

Tabell 4: Översiktlig tabell över respondenternas bakgrundsinformation

Nedan följer en genomgång av fråga 1-9 i enkäten (se bilaga 1) som beskriver procentuella fördelningar bland respondenternas svar.

Ålder och genus



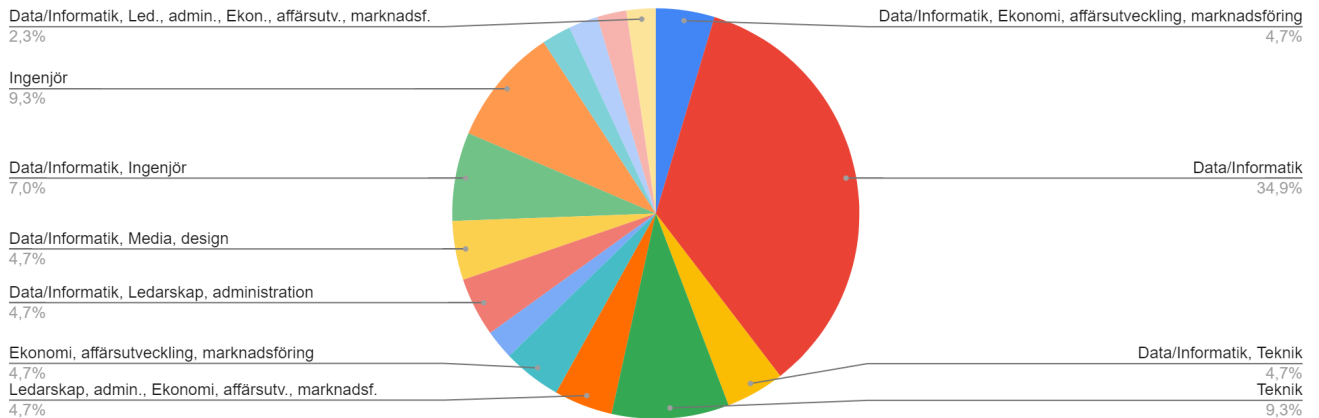
Figur 1: Stapeldiagrammet visar procentuell fördelning av åldersfördelning mellan kön

Det framgick i fråga ett i enkäten att 51,2 % definierade sig som kvinnor och (48,8 %) som män. Ingen av respondenterna angav "icke-binär" eller "vill ej ange". Genusfördelningen var därmed jämn, men skilde sig inom vissa åldersgrupper (figur 1). Den totala fördelningen mellan åldersgrupper var enligt följande: ingen av respondenterna var 19-25 år gamla, totalt 9 respondenter (20,9 %) var 26-30 år gamla, totalt 15 respondenter (34,9 %) var 31-40 år, totalt 8 respondenter (18,6 %) var 41-50 år, totalt 9 respondenter (20,9 %) var 51-65 år och slutligen var 2 respondenter (4,7 %) äldre än 66 år gamla. Majoriteten av respondenterna var mellan 31-40 år gamla.

Utbildning

Vid fråga 3, vilket område deras senaste examen behandlade, valdes ofta fler än ett alternativ av respondenterna. Mindre än hälften, 43,7 % (28 st) angav enbart ett alternativ. 25 av dessa 28 är utbildade inom data/information, teknik eller ingenjör. Två respondenter har svarat enbart inom ekonomi och affärsutveckling. En person har enbart samhällsvetenskap som utbildning.

Procentuell fördelning för utbildningsområden

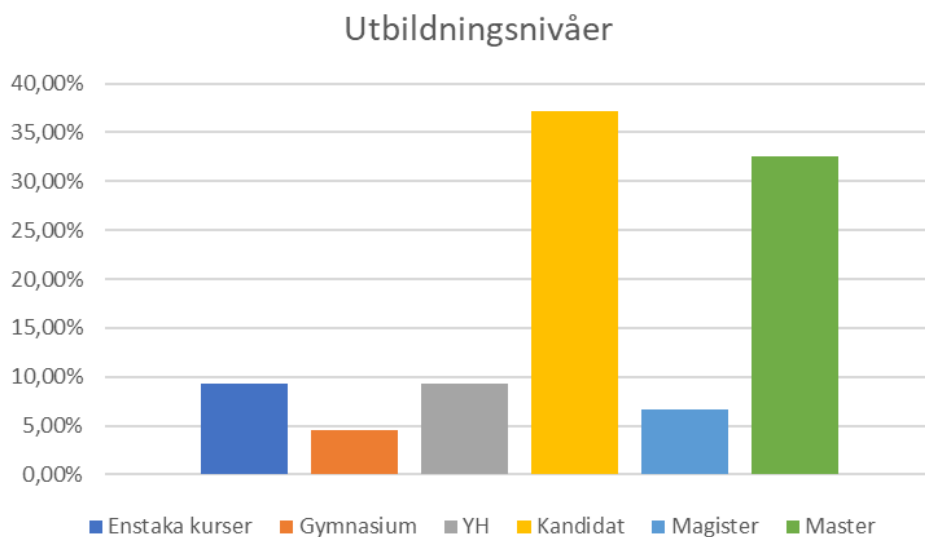


Figur 2.: Diagrammet representerar fråga tre i enkäten som löd "Vilken typ av utbildning har du? Ange det alternativ som är närmast din utbildning eller ange ett eget svar. Ange din senaste avklarade utbildning. Kryssa i fler alternativ om inget enstaka alternativ beskriver din senaste avklarade utbildning."

20,3 % valde två utbildningsområden. Alla utom en person har valt data/informatik, kombinerat med en av följande: ekonomi/affärsutveckling/marknadsföring, ingenjör, ledarskap, administration, teknik, ledarskap/administration, eller media/design, som utbildningsområde.

6,2 % har valt tre utbildningsområden. Utav dessa hade de flesta respondenterna data/informatik och antingen teknik, ingenjör, ledarskap, och/eller affärsutveckling/ekonomi.

Sammanfattningsvis visar fråga 3 att 92 % av respondenterna hade minst en utbildning inom ett tekniskt område såsom data/informatik, teknik, eller ingenjörsvetenskap. 7,8 % av respondenterna hade inte en utbildning inom ett tekniskt område, såsom ledarskap, administration, ekonomi/affärsutveckling/marknadsföring samt samhällsvetenskap.



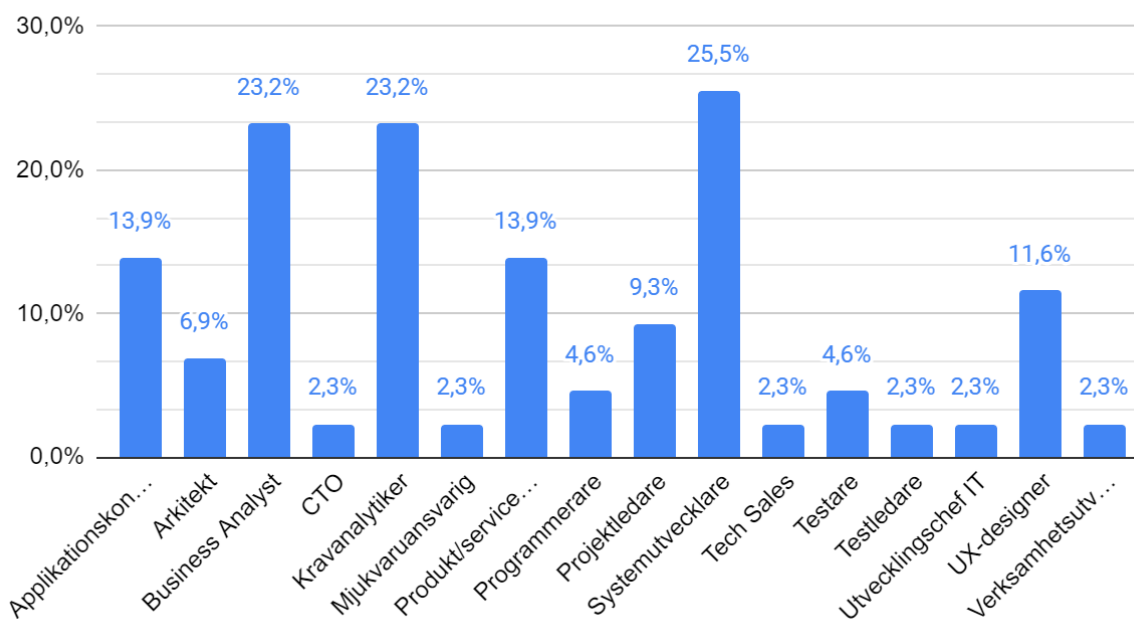
Figur 3: Stapeldiagram över respondenternas högsta utbildningsnivå, angett i procentuell fördelning.

Stapeldiagrammets Y axeln representerar respondenternas antal. Det framgår av fråga fyra i enkäten att 4 stycken (9,3 %) av respondenterna var självlärda eller hade tagit enstaka kurser,

2 stycken (4,7 %) hade gymnasial utbildning, 4 stycken (9,3 %) hade yrkeshögskoleexamen/yrkesutbildning, 16 stycken (37,2 %) hade en kandidatexamen, 5 stycken (4,7 %) hade en magisterexamen 14 stycken (34,9 %) hade en magisterexamen och ingen av respondenterna hade en doktorsexamen. Sammanfattningsvis visar svaret på denna fråga att 37 stycken (76,8 %) av respondenterna hade någon form av eftergymnasial utbildning.

Roller och erfarenhet

Procentuell fördelning av roller

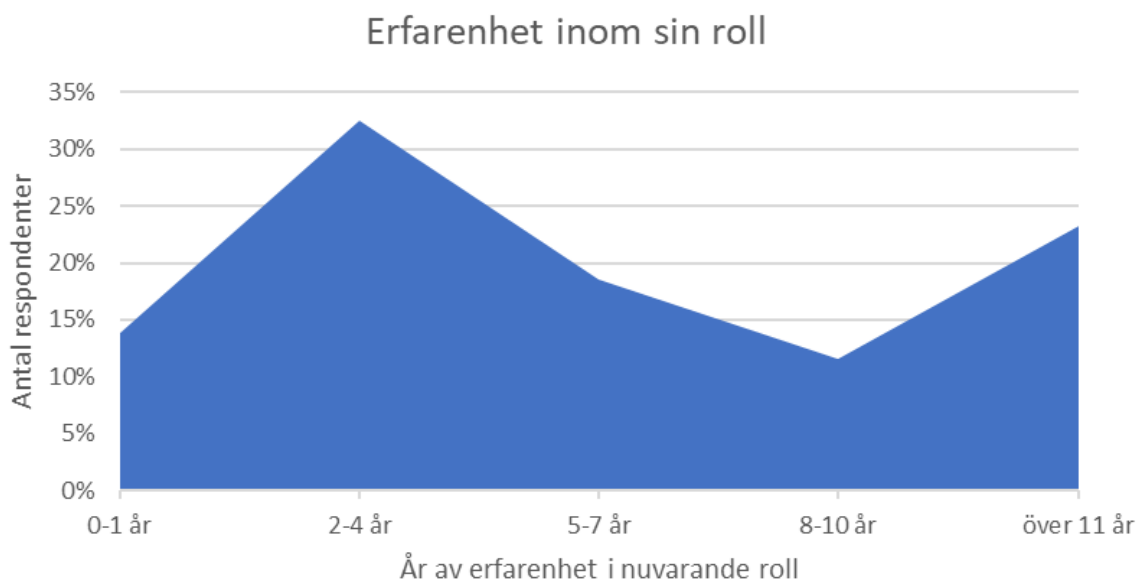


Figur 4: Stapeldiagram av svaren från fråga 5 i enkäten: ”Vilken roll har du idag?”.

Figur 4 visar ett stapeldiagram av svaren från fråga 5 i enkäten: ”Vilken roll har du idag?”. Applikationskonsult och applikationsspecialist har summerats till en gemensam kategori, liksom produktägare och serviceägare. De mest förekommande rollerna var systemutvecklare (25,5%), business analyst (23,2%) och kravanalytiker (23,2%), samt produkt/serviceägare (13,9%) och applikationskonsult/specialist (13,9%). De minst förekommande var CTO, mjukvaruansvarig, tech sales, testledare, utvecklingschef IT och verksamhetsutvecklare IT, samtliga representerades av 2,3% vardera.

39,5 % av respondenterna har kryssat i två till tre alternativ som svar till vilken roll de har. Dessa respondenter kombinerar roller som:

- Business analyst med kravanalytiker
- Produktägare med systemutvecklare
- Produktägare med kravanalytiker, business analyst och arkitekt

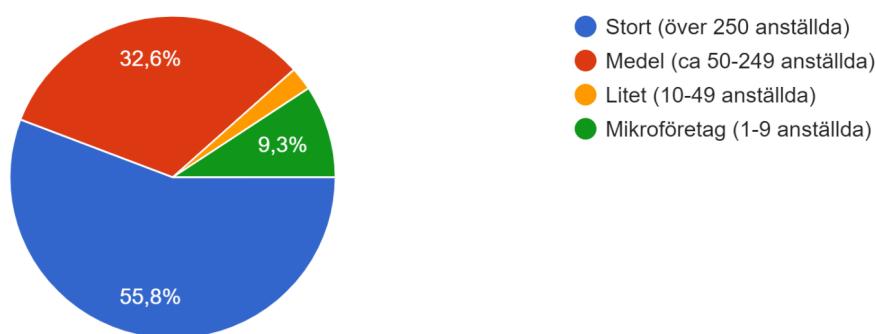


Figur 5: Diagram över respondenternas erfarenhet inom sin nuvarande roll som visas i %.

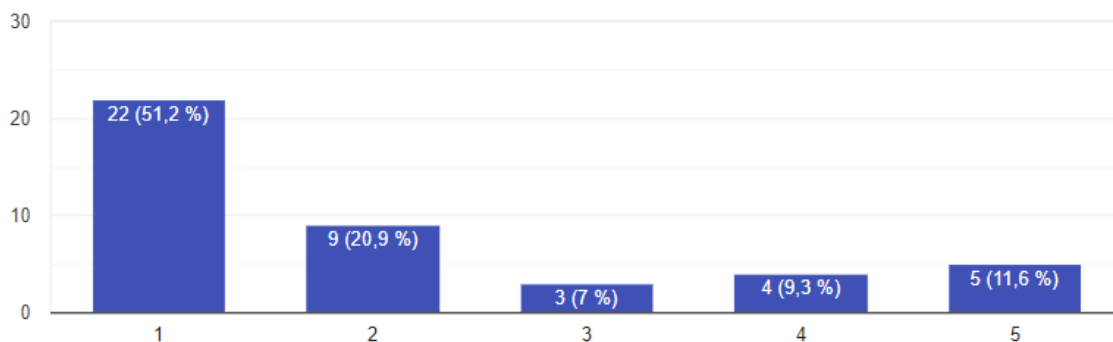
Enligt fråga sex i enkäten har sex respondenterna (14 %) arbetat mellan 0-1 år; 14 stycken (32,6 %) mellan 2-4 år; 8 stycken (18,6 %) mellan 5-7 år; 5 stycken (11,6 %) mellan 8-10 år och 10 stycken (23,3 %) har arbetat mer än 11 år i sin nuvarande roll. Enkäten indikerar att de flesta av respondenterna har 2-4 års erfarenhet av sin nuvarande position.

Mognad och förhållande till IF krav

Fråga sju (bilaga 1) behandlar hur stora organisationer som respondenterna jobbar på och illustreras i figur 6. Det framgår av figur 7 är att 55,8 % arbetar på ett stort företag med över 250 anställda, 32,6 % arbetar på ett medelstort företag med 50-249 stycken anställda, 2,3 % arbetar på ett litet företag med 10-49 anställda medan 9,3 % arbetar på ett mikroföretag med 1-9 anställda. Detta visar att majoriteten av respondenterna som besvarade enkäten arbetade inom ett stort företag och mindre del arbetade inom medelstora företag.



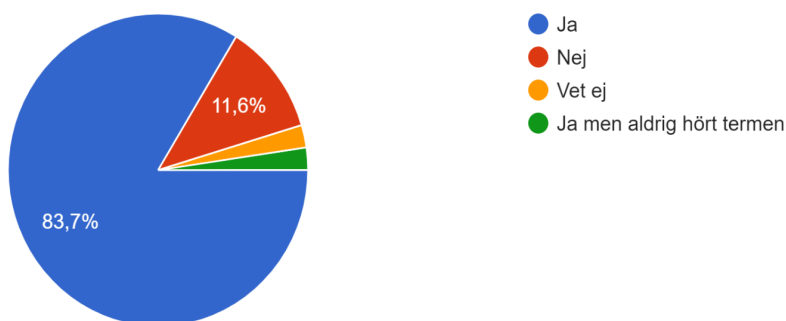
Figur 6: Cirkeldiagram över procentuell fördelning av svaren på frågan: "Hur stort är företaget du jobbar på idag?"



Figur 7: Stapeldiagram över procentuell fördelning av förståelse kring IF krav, Y axeln representerar %.

Figur 7 representerar fråga 8 som löd: "Skulle du säga att du förstår vad man syftar till när man pratar om icke-funktionellt krav?" där svar angavs på en likertskala. En etta innebär att "det är väldigt tydligt" respektive femma innebär att "det är väldigt otydligt".

Enligt enkäten anser majoriteten av respondenterna att de förstår vad IF krav handlar om då de två största svarsgrupperna var "1: väldigt tydligt" (51,2%) och "2: tydligt" (20,9 %). Därefter är det tredje mest förekommande svaret "5: väldigt otydligt" (11,6%) och minst respondenter hade en medel- till sämre förståelse för icke-funktionella krav (7% och 9,3%).



Figur 8: Cirkeldiagram över procentuell fördelning över respondenternas erfarenheter av att arbeta med icke-funktionella krav enligt fråga 9 i enkäten.

Cirkeldiagram presenterar svar på fråga nio (Fråga 9: Har du tidigare jobbat med icke-funktionella krav?) och visade att en övervägande majoritet (83,7%) hade erfarenhet av att arbeta med IF krav. Två personer har varken svarat ja eller nej: 2,3 % av respondenterna (1 person) har angett vet ej och 2,3 % (1 person) har angett ett eget svar: "ja men aldrig hört termen".

2.4.3 Datainsamling

För att ta fram information om individuella synpunkter gällande dokumentation, hantering, ansvarsfördelning och uppfattning av IF krav har en enkät med olika frågor skapats som skickades ut till respondenter som arbetar med kravhantering inom svenska mjukvaruutvecklingsprojekt. Enkätundersökning anses vara det mest lämpliga tillvägagångssättet till studiens forskningsfrågor och är den dominerande datainsamlingsmetoden inom den kvantitativa ansatsen (Jacobsen 2017). Genom att samla in

data med hjälp av en enkätundersökning utvecklades inga personliga bindningar till de respondenter som besvarade enkätfrågorna vilket medförde en mer kritisk distans till de undersökande fenomen (ibid).

Efter att ha litteraturgranskat området IF krav uppnåddes en hög grad av mättnad av kvalitativa undersökningar av hur IF krav hanteras och dokumenteras. Detta syftade till att undersöka fenomenet IF krav ur ett snävare men allt djupare perspektiv för att bättre förstå ämnet. Eftersom det är ett ämne som beskrivs som tvetydigt och till och med saknar en accepterad definition är det rimligt att också undersöka ämnet med en kvantitativ ansats. Denna studie har använt en kvantitativ ansats eftersom den syftar till att förstå hur fenomenet IF krav hanteras gällande ansvar, kvalitetssäkring och dokumentation i den svenska mjukvaruutvecklingsprojekt. Genom att fråga ett brett spann av olika personer med olika roller inom mjukvaruutvecklingsprojekt, förstå vilka typer av utbildningar, roller, tillvägagångssätt för att dokumentera IF krav, ansvarsområden och hantering av IF krav som förekommer, samt förstå deras uppfattning och dokumentation om IF krav, kommer forskningsfrågorna att besvaras.

Målet med uppsatsen var att undersöka dokumentation och ansvar för IF krav inom kravhanteringsfasen i SDLC. Istället för att jämföra det traditionella och det agila perspektivet av SDLC, ligger fokus på att analysera fenomenen som är dokumentering och ansvar för IF krav, där de mest förekommande tillvägagångssätten för just det tas fram och identifieras under datainsamlingen såväl som teorin och blir uppsatsens fokusområden. Resultatet blir därför en förankring av hur väl de IF kraven dokumenteras i svenska mjukvaruutvecklingsprojekt och vilka som är ansvariga för dem, utan hänsyn till om det är ett agilt eller traditionellt tillvägagångssätt.

2.4.4 Enkätundersökning

Det krävs enligt Jacobsen (2017) ett gediget förarbete innan en enkät kan konstrueras eftersom frågorna blir svåra att revidera efter att enkäten har skickats ut. Enkäten (Se bilaga 1) konstruerades i en iterativ process där frågorna reviderades för att inte vara otydliga, ledande eller dubbla. Det som är viktigt att åstadkomma innan ett formulär skapas är att konkretisera de begrepp vi ville mäta (Jacobson 2017). Enkätundersökningen syftade att mäta förekomsten av IF krav; kännedomen om och förståelsen för IF krav hos personer som jobbar med krav, vilken uppfattning personer har om vikten av IF krav i mjukvaruutvecklingsprojekt, förekomsten av ansvarsroller relaterade till IF krav samt hur och även om de IF krav dokumenteras eller inte.

Vi operationaliserade begrepp, det vill säga att gå från något abstrakt till något konkret och mätbart (Jacobsen 2017). Utifrån vår problemformulering diskuterade vi tillsammans vilka frågor och begrepp som var viktiga att ha med och tog hjälp av vår kvalitativa litteraturanlys. Genom litteraturanalysen fick vi ett brett perspektiv över vilka utmaningar det finns med IF krav samt en översikt över vilka intervjufrågor och frågeställningar som redan använts i dessa artiklar. Artiklarna i vår litteraturstudie hade kvalitativa ansatser vilket innebär att det är mer djupgående svar från enskilda individer.

Vårt fokus i denna studie var att studera IF krav och dokumentation av dessa i ett bredare perspektiv för att kunna generera generaliserade svar som representerade den svenska mjukvarumarknaden. För att kunna göra det använde vi återkommande begrepp som förekommer i artiklarna i litteraturstudien såsom dokumentation, ansvar, dokumentationstekniker och kvalitetssäkring med mera, det vill säga de begrepp som finns i

vår studie för att se om insamlad empiri skulle skilja sig mot tidigare forskning. Varje artikel lästes igenom och vissa begrepp blev mer synbara såsom IF krav, uppfattning och förståelse samt tillvägagångssätt för dokumentation av IF krav, ansvar och hur kvalitetssäkring sker (se tabell 3 för respektive tema som har hittats i respektive artikel). De förekommande begrepp bröts ner till mer konkreta frågor genom att ta i beaktning kring vilka utmaningar som förekommer samt vilka frågor som författarna av artiklarna ansåg skulle förekomma i framtida studier. Genom vår kvantitativa analys av empirin fick vi en desto djupare förståelse för hur den svenska marknaden förhåller sig till fenomenet, tack vare problembilden som vi tidigare studerat i tidigare forskning.

Google Forms användes för att skapa enkäten, samla in svar och analysera svaren, vid vissa frågor användes även Office Excel för att sammanställa resultatet. Totalt ingick 24 frågor i enkätundersökningen. Det förekom både kategorisvar som var av nominal och ordinarie mätnivå i våra frågor. Nominella svar är av karaktären "Ja", "Nej" och "Vet inte" och används för att skapa tydlighet i mätnivån till inställningen att vara för eller emot något. Ordinal mätnivå beskrivs i termer av att mäta intensiteten, frekvensen, jämförelse eller prioritering av enskilda förhållandet i en specifik fråga, exempelvis "Mycket bra", "Bra", "Snarare dåligt" och "Mycket dåligt" (ibid). Det förekom även frågor utifrån en Likertskala med svar från 1-5. Fråga 1-7 beskriver respondentens bakgrund gällande könsidentitet, ålderskategori, utbildningsnivå, utbildningsområde, nuvarande roll och hur länge hen arbetat i sin nuvarande roll samt hur storleken på företaget de arbetar på idag - angivet i antal anställda. Definition av olika storlekar på företag (stort, litet, medel- och mikroföretag) härleddes från Ekonomifakta (2023). Med dessa frågor erhöles grundläggande bakgrundsinformation om varje respondent då en inblick gavs om skillnader i ålder, storlek på företaget, könsidentitet och nuvarande arbetsroll samt dess relaterbarhet till hur dokumentation av krav uppfattas och hanteras. Dessa frågor visade på den procentuella fördelningen på varje fråga för att på ett överskådligt sätt se vilket som var det mest förekommande alternativet.

Fråga 3, 5, 13 och 18 innehöll ett "Annat" alternativ vilket betyder att respondenten kunde ange ett eget alternativ om de fördefinierade alternativen inte stämde för dem. Det öppna svarsalternativet bidrog till att enkäten hade mer heltäckande kategorier och svar. Att erbjuda öppna/egendefinierade svarsalternativ kan göra att respondenterna inte känner sig begränsade att svara utifrån de fördefinierade svarsalternativen (Jacobsen 2017). Fråga 18 hade flest angivna öppna svar där respondenten själv skrev in ett svarsalternativ. Fråga 17 representerar en fråga utifrån en ordinal mätnivå där frågan syftar på att belysa hur ofta IF krav revideras tillsammans med kund, det vill säga hur ofta förändras dokumentationen. För att tydliggöra skillnaden mellan IF krav och FK ber samma fråga respondenten att svara på hur ofta de reviderar FK med kund.

Frågorna 8, 15, 16, 23, 24 beskriver en Likertskala där studien vill belysa hur lätt, tydligt och viktigt dokumentation av IF krav är hos respondenterna samt hur den generella förståelsen kring IF krav ser ut. Frågorna 11,12, 19, 20, 21 och 22 utgår från "Ja", "Nej", "Ibland" och "Vet ej" svar. Dessa frågor belyser om dokumentation sker, om det finns en ansvarig person för de IF krav, om ansvar för olika IF krav fördelas, om de fördelar krav och förekommer en likartad uppfattning om vad som ska implementeras. Denna fråga ville belysa om respondenterna dokumenterade sina krav överhuvudtaget och om det finns någon som ansvarar över detta och om ansvarsfördelningen varierar eller inte. Fördelen med en kvantitativ studie över en kvalitativ i detta syfte är att kunna generalisera roller från flera olika företagskulturer, riktlinjer och storlekar på företagen och därmed få en bättre förståelse för vilka trender som finns (Jacobsen 2017).

2.4.5 Webbaserad enkät

För att få synpunkter på den skrivna enkäten skickades den ut till tre personer för en första revision. Både skriftliga och muntliga synpunkter på frågor samlades in som sedan användes som underlag för att uppdatera, förenkla och förtydliga enkäten. Jacobsen (2017) argumenterar att denna metod är värdefull då den ger insikter om oklarheter och fel som den som skriver formuläret kan ha svårigheter att inse. De som kritiskt granskade enkäten fick även fylla i den som om de skulle svara på den på riktigt och tog tiden för hur snabbt det gick att genomföra, vilket landade i 3-5 minuter.

Google Forms användes för att på ett enkelt sätt strukturera och skriva frågorna i den ordning som ansåg vara mest funktionellt och trovärdigt då Google är en välkänd internetsida, men även för att insamlingen skulle ske på ett enkelt och snabbt sätt. Detta medför att spridningen kan bli bredare om det förekommer aktiva utskick av mejl med en klickbar länk. Utskicket av enkätundersökningen startade den 5 april och målet var att få in 100 svar innan den 25 april. Företag kontaktades via den mail som finns publicerad på företagets webbplats, den användes för att skicka ut en förfrågan om att delta i denna enkätundersökning tillsammans med bifogad länk till enkäten. Mejlet som skickades ut till företag standardiserades för att effektivare kunna skicka ut till flera företag samtidigt och nå representativa personer för denna studie. Mejlet innehöll en kort presentation av vad studien ville undersöka, syfte med studien, vilka vi var, vilken skolan vi gick på, hur många svar vi hoppades få in, hur lång tid enkäten skulle ta att göra samt att vi inte samlat in kontaktuppgifter i samband med enkäten. Detta sätt att formulera ett mejl är ett gediget sätt att mejla ett företag enligt Jacobsen (2017).

Den 24 april hade 43 enkätsvar samlats in och olika teman i svaren började framträda. En tillräckligt hög mättnad hade uppnåtts av information för att kunna gå vidare med studien. Av 72 företag som kontaktades var det nio stycken företag (12,5 %) som hade svarat på det utskickade mejlet och valde att inte medverka i enkätundersökningen. Skälen till detta var företagspolicy och tidsbrist. Datainsamlingen resulterade i en tillfredsställande svarsfrekvens på 59,7 % eftersom antagande gjordes att inom varje företag hade en person bidragit med ett svar. Svarsfrekvensen är viktig inom enkätundersökningar och betyder hur stor andel av dem som har kontaktat och frågat verkligen svarade på enkäten (Greve 2021). Svarsfrekvensen beräknades med hjälp av en svarsprocent formel som presenterades av Jacobsen (2017), där 43 svar togs som total antalet enheter som har svarat och delades per total antal enheter i det ursprungliga urvalet som vi hade på 72 personer som kontaktades via företag. Eftersom studien ämnar undersöka om förekomsten av IF krav och olika nyanser med dem när det gäller förståelse och dokumentation, räknades alla svar som inkommit som användbara svar. Till de som inte besvarade enkäten beräknas 40,3 % som bortfall, vilket motsvarar 29 uteblivna svar. Jacobsen (2017) förespråkar dock att mindre än 100 svar kan påverka möjligheten att kunna utföra en gedigen analys av den insamlade data men även att felmarginalerna kan påverkas negativt.

Att använda webbaserade undersökningar kan ge större bortfall av svarsprocent jämfört med personliga intervjuer eller postenkäter. I den personliga intervjun kan följdfrågor ställas och oklarheter kan diskuteras. Att använda sig av webbaserade enkäter tar bort den personliga kontakten mellan intervjuaren och respondenten (Jacobsen 2017). Jacobsen (2017) framför att människor idag kan få många mejl dagligen och att det kan finnas skepticism till att öppna okända länkar från okända personer på grund av risken att eventuellt få virus till dator.

2.4.6 Dataanalys

En univariat analys användes för att analysera enkätsvaren. Detta innebär enligt Jacobsen (2017) att beskriva en variabel i taget för att exempelvis ta fram det typiska svaret, variationen eller procentuella fördelningen i ett svar. Google Forms och Office Excel tog fram cirkeldiagram och stapeldiagram på alla separata frågor automatiskt beroende på vilka typer av frågor det var. Varje enskilt respondents svar kunde även granskas, vilket gjordes i särskilda fall.

Utifrån varje fråga skrevs den procentuella fördelningen ner i text för att få en bättre textuell översikt som sedan kunde användas i resultatdelen av denna studie. Fråga 3 handlade om vilka utbildningsområden respondenten hade och det som framfördes var den procentuella fördelningen om respondenterna hade valt ett till tre olika alternativ samt hur många hade ett tekniskt utbildningsområde och vilka som inte hade det. Fråga 10 var en fråga där respondenterna testades på deras uppfattning om vad de anser är IF krav utifrån text. Procentuellt hur många rätt och fel respondenterna hade beskrevs för att se hur olika de uppfattade texten. Fråga 13 var en fråga som även krävde lite mer detaljerad analys. Frågan vill beskriva hur och om respondenten dokumenterar IF krav idag. Procentuell beskrivning togs fram för att visa hur många respondenter som använder ett till sju olika tillvägagångssätt för dokumentation av IF krav för att kunna påvisa hur många som använder få eller flera olika tillvägagångssätt. Fråga 18 handlade om vad respondenter utgår ifrån när de ska implementera de IF krav. Eftersom denna fråga hade ett öppet svarsalternativ, utöver de förutbestämda svarsalternativen kunde respondenten själv skriva in ett svar. Utifrån de svar som kom in på denna fråga tematiseras innehållet för att kunna på ett mer strukturerat sätt beskriva de svaren, då det fanns stor variation.

2.5 Metodreflektion

En stor utmaning med att använda enkäter är att mäta det som egentligen ska mätas. Genom att ha en kritisk syn på arbetsgången kan det bidra till en tillräckligt bra giltighet. Kritisk syn handlar om att svara på frågor gällande komplexa fenomen som kan bestå av flera olika delmoment (Jacobsen 2017). Genom arbetet av denna kandidatuppsats upptäcktes flera problem och saker som missats, dessa beskrivs i avsnitten nedan.

2.5.1 Reliabilitet

De personer som svarade på enkätundersökningen hade tillförlitlig och objektiv information, då de arbetade med kravhantering och dokumentation av krav och kunde därmed tala ur sina egna erfarenheter. Detta beskriver Jacobsen (2017) som förhandskällor. Han argumenterar att källor behöver kontrolleras mot andra oberoende källor för att få olika tolkningar. Eftersom enkäten besvarades av personer med olika befattningar kunde därmed studien få ett brett spektrum av svar för vidare analys. Det fanns inte en möjlighet att med full säkerhet veta att respondenterna som kontaktat arbetar med att dokumentera IF krav utan en tillit skapades till att de på något sätt kommit åt eller hört talas om begreppet IF krav inom sitt arbete. Genom att använda ett godtyckligt urval som denna studie tillämpat finns det en risk enligt Jacobsen (2017) att undersökarna utelämnar vissa brukargrupper utan att vara medvetna om det.

Jacobsen (2017) anser att skicka ut en provenkät är ett bra och snabbt tillvägagångssätt för att få synpunkter på enkäten som skapats. Förslagsvis bör den skickas ut till 4-6 personer inom

det område som enkäten syftar till. På grund av tidsbrist skickades provenkäten ut till tre systemvetare studenter som officiellt inte har arbetat med IF krav eller dokumentation av detta i arbetslivet. Detta kan kopplas till en pilotundersökning som enligt Eliasson (2013) innebär en provundersökning där några tillfrågas att delta men som inte ska medverka i den tilltänkta undersökningsgruppen. Dock saknar dessa personer arbetserfarenhet inom det undersökta fenomenet men var välbekanta med IF krav i teorin och praktik genom sina studier. En sådan pilotundersökning har varit till hjälp att redigera frågor som var otydliga eller dubbla och få olika synpunkter angående enkätfrågorna.

Det missades att inkludera kontaktinformation till någon av skaparna av enkäten om det eventuellt skulle uppstå frågor från respondenterna, vilket Jacobsen (2017) anser är viktig information att inkludera. Han argumenterar att desto fler antal svar som forskare kan samla in, desto större blir reliabiliteten av undersökningen (ibid). Påminnelsemejl skickades inte ut för att kunna minimera bortfall av svar och visa att respondenten anses vara viktig i undersökningen. Jacobsen (2017) påpekar att påminnelser i webbaserade undersökningar ger högre svarsprocent, då den till och med inkluderar länken till enkäten. Då respondenterna var helt anonyma i enkäten var det inte heller möjligt att kontrollera vilka personer som besvarat enkäten. Med hänsyn till detta skickades inte påminnelsemejl ut till dem som enkätinbjudan skickats till, med risk att skicka påminnelse till dem som redan hade svarat. För framtida studier bör en annorlunda teknik tillämpas för att kunna skicka ut påminnelsebrev endast till dem som inte besvarat enkäten.

En orsak som kan ha bidragit till att studien endast fick in 43 svar och inte 100 var att länken till enkäten skickades ut direkt i mejlet till de 72 företagen som kontaktades. Detta var oprofessionellt av den anledningen att det inte först skickades ut ett frågande mejl om företag skulle kunna tänka sig att vara med i enkätundersökningen och sedan efter korrespondens med företaget skicka ett mejl på nytt med länken. Motivet till att inte fler personer via företag kontaktats var att det gjordes ett antagande av att tillräcklig mättnad hade uppnåtts i den information som samlats in. Antal respondenter prioriterades inte, utan den främsta prioriteten var att kontakta relevanta respondenter som kan vara representativa till hela populationen av personer som arbetar inom mjukvaruutvecklingsprojekt. Därmed lades tid på att hitta relevanta företag genom att kontakta de företag som ansågs vara representativa för denna studie. Detta avgjordes genom att hitta företag inom mjukvaruutvecklingsbranschen, gå in på deras hemsidor och se om deras kontor finns i Sverige. Enligt Jacobsen (2017) finns det en enkel tumregel för storleken på urval. Han menar att urval om mindre än 100 enheter försvårar en vettig analys av informationen och dessutom blir felmarginalen större (ibid).

En annan tänkbar anledning till att enkätundersökningen endast genererade 43 svar var att företag inte hade tid att svara på enkäten eller hade ointresse om ämnet. Jacobsen (2017) menar att intresset för webbaserade enkäter är relativt svagt och risken är stor att personer inte väljer att medverka. Enkätundersökningen i denna studie var relativt lång med 24 frågor och detta kan ha haft en bidragande orsak till att så få svarade. En lång enkät kan avskräcka personer att svara då det kan anses vara jobbigt att genomföra men även att ämnet inte är av intresse och därmed vill en medarbetare inte lägga fokus på att besvara enkäten (ibid). Det som studien hoppades ge högre svarsprocent var anonymitet i undersökningen. Enligt Eliasson (2013) är den svarandes identitet helt ointressanta i en kvantitativ undersökning utan det som är viktigt är att respondenten är en representant för undersökningsgruppen. Därför har enkäten beskrivit att alla svar är anonyma och därmed försäkrat svarandes identitet inom studien. En svarsfrekvens på 59,7 % anses vara tillfredsställande enligt Jacobsen (2017) som menar att över 50 % räknas som tillfredsställande. Jacobsen (2017) menar även att en sådan

regel är enkel och problemet ligger mest då inte hur många som har svarat utan vem som har svarat. Detta har studien lagt stor uppmärksamhet på, då studiens urvalsram består av respondenterna med relevant erfarenhet och roll inom mjukvaruutveckling för att delta i studien enkätundersökning.

40,3 % anses vara ett bortfall i denna studie vilket motsvarar 29 uteblivna svar utifrån ursprungliga urvalet som inte har svarat på enkäten. Av dem är det endast sju som har angett skälen till uteblivet svar, som var tidsbrist i de flesta fall. Bortfall på 29 uteblivna svar anser eventuellt ligga i den systematik då de flesta företag var mjukvaruutvecklingsföretag som antagligen inte hade tid att delta i enkätundersökningen och en annan synpunkt var att deras företagspolicy som inte gav oss möjligheten att få flera svar. Flera åtgärder hade kunnat vidtas för att minska den fristående precision i studien, exempelvis genom att skicka flera påminnelser, ringa eller minska brister i frågeformuläret. Det som även kan ha bidragit till 40,3 % bortfall kan ha varit att företagen vi kontaktade inte specifikt arbetar med kravhantering, vilket vi antydde var ett kriterium för att delta i mejlet som skickades ut. Ett godtyckligt urval innebär enligt Jacobsen (2017) att personer som forskaren kontaktar tror är representativa för att delta i studiens enkätundersökning. Genom att vi kontaktade företag inom mjukvaruutvecklingsbranschen kan vi inte påvisa att de arbetade specifikt med IF krav eller inte. Dock gjorde vi ett antagande vid urval av företag att personen som öppnade mailet kunde avgöra om det fanns individer som hen kunde skicka enkäten till eller inte. Detta anser vi kunnat sätta igång snöbollseffekten som vi hade också angett tidigare och använd inom denna studie.

2.5.2 Validitet

Vi har gjort en extensiv undersökningsdesign. Enligt Jacobsen (2017) menas en extensiv undersökningsdesign med att man går på bredden; man studerar många enheter, i motsats till den intensiva designen som går på djupet. I den extensiva undersökningsdesignen är några få centrala variabler starkt fokuserade på. I fallet om denna studie är den centrala variabeln kvalitetssäkring av IF krav genom dokumentation och roll- och ansvarsfördelning. Syftet med studien – att undersöka dokumentationen av IF krav inom svenska mjukvaruutvecklingsprojekt – är att kunna generalisera resultaten från ett urval till en population, i detta fall är populationen yrkesverksamma inom svenska mjukvaruutvecklingsprojekt. Då många perspektiv fångats i studien, inklusive perspektiv från olika åldrar, yrkeserfarenheter och yrkesroller, är risken att populationen inte helt känner igen sig i beskrivningen (ibid). Fördelen med extensiv undersökning är å andra sidan att det skapar bra underlag för statistisk generalisering, se avsnitt 6.1.

Enkäten till denna studie skrevs innan det egentligen fanns ett klart syfte om vad som skulle undersökas. Begränsningen blev till dokumentation och IF krav men osäkerheten gjorde att enkäten skrevs utifrån olika aspekter av dessa områden. Detta kan ha påverkat validiteten inom enkätundersökningen eftersom studien kan ha missat flera relevanta frågor som skulle ha varit relevanta och fruktbara. En fråga som missades var att inkludera i enkäten var om personen jobbar enligt den agila eller traditionella systemutvecklingsmetoden, vilket hade varit givande för studien eftersom mycket av litteraturen hänvisar till just agila metoder angående problem med dokumentation. Det hade varit av ytterst vikt att även inkludera fler frågor om att kvalitetssäkra de IF krav och mer specifikt vilken roll som har ansvar för att dokumentera de IF kraven i företaget respondenterna arbetar på idag.

Kvalitetssäkring handlar enligt svensk kvalitetssäkring mycket om att öka kundnöjdheten (SS-EN ISO 9001:2015). Enbart en fråga behandlade om respondenten var involverad vid revideringar av krav med kunden. Det hade varit av intresse att undersöka till vilken grad dokumentationen av IF krav utförs av kunden istället för mjukvaruutvecklingsföretaget; vidare om kunderna gör det själva eller i samråd med någon från företaget. Det hade då även varit intressant att undersöka vilken typ av roll som kunden hade samarbetat med för att ta fram kraven. Jarzebowicz och Weichbroth (2021) hävdar att det råder tvetydigheter när kunden själva dokumenterar de IF kraven på grund av bristande kunskap om konceptet.

I studien har vi diskuterat vikten av effektiv kommunikation inom utvecklingsteamet samt tidig specifikation av IF krav från kunden som båda kan minska behovet av revisioner. Det finns utmaningar gällande dokumentationen om kunden inte är tillgänglig under dokumentationsprocessen, dock har vi inte mätt kommunikationen varken mellan teammedlemmar eller till kunden i vår enkätundersökning, vilket hade varit en intressant aspekt att beräkna för att svara på vår FF1 hur väl IF krav dokumenteras, då Behutiye et al. (2017) menar att kommunikation med kund kan prioriteras över dokumentationen i agila mjukvaruutvecklingsprojekt.

Fråga 6 belyser hur länge respondenten har arbetat i sin nuvarande roll, vilket ger begränsningar i frågan eftersom hela respondentens erfarenhet under sin karriär av IF krav inte tas med i beaktning. Men på grund av att enkäten skickades ut innan forskningsfrågan var ordentligt avgränsad fanns det inget utrymme för att lägga till dessa frågor.

Fem öppna frågor har valt att användas i tillägg till fem fasta frågor, eftersom det har tänkts ge respondenterna mer utrymme för att uttrycka sig med egna ord genom att ange sitt eget alternativ som svar på enkätens fråga. Detta anser Jacobsen (2017) kan ge mer utrymme för undersökare för att få mycket intressant information. Dock är bortfallet vid öppna frågor betydligt större än vid fasta svarsalternativ (Jacobsen (2017)). Därför har det valt att endast ha öppna frågor som tillägg till fasta svarsalternativ.

Frågor i enkäten konstruerades i en iterativ process, men trots revideringar efter pilotundersökning samt hänsyn till all feedback så landade enkäten i två frågor som anses vara ledande frågor. Jacobsen (2017) avråder undersökare från att ha ledande frågor, då dessa anses styra respondenten i en viss riktning, då sannolikheten är stor för att få respondenten att svara "rätt". Även om de två sista frågorna inom enkäten var ledande (se bilaga 1), så fanns ett behov av de frågorna, då de kunde mäta hur viktigt det är att dokumentera IF krav och vilken betydelse har de för mjukvaruutveckling.

Som en följd av det som nämnt ovan, har det konstaterats både svaga och starka sidor av en webbaserad enkät. Jacobsen (2017) menar att med en webbaserad enkät har en svaghet när det gäller ett representativt urval av befolkningen jämfört med andra datainsamlingsmetoderna. Svarsprocenten på 59,7 % där 51,2 % kvinnor och 48,8 % män har svarat på enkäten är ganska tillfredsställande för studiens omständigheter. Det som även stärker validiteten av denna studie är att enligt fråga 2 (se bilaga 1) förekommer det olika åldersgrupper som har svarat på vår enkät. Flera olika respondenternas åldrar gjorde att vi har fått ett brett perspektiv på samma fenomen.

2.5.3 Etiska ställningstaganden

Respondenter informerades om syftet med enkäten i mejlet som skickades ut och att informationen inte skulle spridas vidare, samt att deras mailadresser inte skulle samlas in. Alla frågor förutom några få var obligatoriska, vilket gör att om respondenten väl påbörjat enkäten kan denne känna sig tvingad att fortsätta, vilket kan påverka om personen vill medverka eller inte. Vid frågan om respondentens erfarenhet fanns det inget alternativ eller information om hur respondenten ska svara om hen varit sjukskriven en del av den totala tiden som respondenten varit anställd och jobbat med krav. Om respondenten har varit sjukskriven en del av sin anställning kan frågan ha skapat förvirring och ovilja att svara med tanke på att respondenten inte vill svara fel. Det saknades förklaring huruvida respondenten ska räkna in de åren hen varit sjukskriven, eller inte.

I och med att IF krav uppfattas som tvetydiga av många kan det ha påverkat villigheten att svara på enkäten hos de personer och företag som tog emot enkäten. Eftersom en enkät ska vara enkel att svara på enligt Jacobsen (2017), går studiens enkät emot den förutsättningen. Denna aspekt kan ha påverkat antalet svar som kom in, även om en definition av IF krav förekom i början av enkäten för att minska tvetydigheten. Detta kan kopplas till fråga 10 (Se bilaga 1) där respondenten fick kryssa i vilka som är IF krav utifrån olika påstående. Denna fråga kan uppfattas som jobbig att besvara då den inkluderar rätt och fel svar. Respondenten kan känna sig motvillig att besvara frågan om hen känner sig osäker på frågan.

3 Teoretiskt ramverk

I följande kapitel behandlas de utvalda artiklarna från litteraturstudien. För att kunna besvara forskningsfrågorna och ge läsaren en förståelse för helheten av studiens fenomen, strukturerades de valda artiklar till fyra teman relaterade till utmaningar med att identifiera IF krav, dokumentationens betydelse, ansvarsroller och kvalitetssäkring av IF krav.

3.1 Utmaningar med att identifiera IF krav

De IF kraven har en central och avgörande roll för att ett mjukvaruutvecklingsprojekt ska lyckas. Till skillnad från FK, specificeras kvalitetsattributen av mjukvaran i de IF kraven (Behutiye et al. 2020b). Karhapää et al. (2021) argumenterar att IF krav är viktiga för systemets funktionalitet och menar att om endast de FK uppnås, kommer systemet inte att motsvara kundens kvalitetskrav. Om IF krav såsom prestanda och användbarhet inte tillhandahålls, kommer systemets kvalitet inte vara tillräckligt hög för att motsvara kundens förväntningar (ibid). Dock visar forskning att de IF kraven ofta är vagt definierade, svåra att definiera samt har lägre prioritet än de FK (Behutiye et al. 2017; Ramesh, Cao & Baskerville 2010). De IF kraven bortprioriteras ofta när en deadline närmar sig och står dessutom inte alltid som enskilda krav utan är relaterade till ett FK. Detta är problematiskt då det sällan är alla FK som implementeras, vilket betyder att om ett FK inte implementeras eller nedprioriteras, implementeras inte heller det IF kravet relaterat till det FK och ärver dessutom samma nivå av prioritet som dess FK har (Alsaqaf, Daneva & Wieringa 2019). Det i sin tur leder till att IF krav inte prioriteras och ett mindre viktigt IF krav kan exempelvis bli implementerat istället för ett viktigt, beroende på vilken prioritet dess FK har (ibid). Karhapää et al. (2021) framför att IF krav kan vara beroende av varandra, vilket innebär att ändringar av ett krav kan påverka andra krav. Detta gör det utmanande att dokumentera dem isolerat och kräver ett helhetsgrepp och förståelse för vardera IF krav för att säkerställa att de alla specificeras.

Eckhardt, Vogelsang och Fernandez (2016) konstaterar att det finns skillnader mellan IF krav och FK både i hur dess karaktär och i hur dessa krav insamlas, dokumenteras och valideras i praktiken. Trots att IF krav definierar hur systemet ska göra något, så kvarstår det faktum att ett allmänt accepterat tillvägagångssätt för dessa inte existerar. En stor utmaning med IF krav i mjukvaruutvecklingsprojekt är att de sällan blir kvantifierade vilket betyder att de blir svåra att analysera och testa. Eckhardt, Vogelsang och Fernandez (2016) beskriver att det är en utmaning att urskilja IF krav från FK och därmed definiera dem, trots befintlig definitionsskillnad: FK avser *vad* systemet skall göra och IF krav avser *hur* något ska göras. Eckhardt, Vogelsang och Fernandez (2016) betonar att problemet med att kunna skilja dem isär är att de flesta IF krav beskriver beteendeegenskaper. Ali et al. (2022) beskriver IF krav som kvalitetsegenskaper hos ett mjukvarusystem och argumenterar att hur väl de definieras är avgörande för att mjukvaran ska vara framgångsrik. Vidare menar de att odefinierade IF krav leder till dålig kvalitet på systemet och därmed till att mjukvaruutvecklingsprojekt misslyckas (ibid).

3.1.1 (Tyst) Kunskap om IF krav

Jarzebowicz och Weichbroth (2021) argumenterar att om det finns gedigen kunskap om IF krav i en mogen organisation så förbises inte de IF kraven. Ibland förekommer det att kunden "över-specificerar" de IF krav till den grad att det behövs nå konsensus mellan kunden och leverantören, då alla krav inte kan vara tekniskt eller ekonomiskt genomförbara (ibid). Ramesh, Cao och Baskerville (2010) hänvisar till att det kan leda till försummelse av IF krav eftersom kunden överlag prioriterar de FK då dessa anses ge mer direkt affärsvärde. Behutiye et al. (2020b) beskriver att kunden oftast saknar insikt och tillräcklig kunskap om IF krav och har på så sätt svårigheter att kommunicera dessa behov. Karhapää et al. (2021) beskriver att bristande kunskaper kan i vissa fall finnas hos utvecklarna gällande IF krav såsom användbarhet eftersom dessa krav är mer "osynliga" och därmed ignoreras dessa. Det som även blir viktigt är att involvera kunden och skapa medvetenhet för att visa på att IF kraven är en viktig del av ett mjukvarusystem (ibid). Dock framförde Karhapää et al. (2021) att seniora utvecklare som har en god kunskap om IF krav, men att denna kunskap är deras tysta kunskap. Utvecklarnas tysta kunskap är deras intuitiva förmåga att förstå och hantera IF krav prioriteras före dokumentation.

Alsaqaf, Daneva och Wieringa (2019) argumenterar att det är svårt att föra vidare tyst kunskap från de mer seniora utvecklarna eftersom många gånger är den inte tillräckligt definierad och det finns inga riktlinjer för hur kunskapen ska föras vidare till andra, till exempel juniora utvecklare. Detta kan kopplas till det Behutiye et al. (2017) beskriver att inom den agila mjukvaruutvecklingen förespråkas mindre dokumentation men mer kommunikativ interaktion med kunden, detta för att minska behovet av att specificera de IF kraven. Om dokumentation inte finns kan nya utvecklare i mjukvaruutvecklingsteamet ha svårigheter att sätta sig in i och skapa en överblick över de IF kraven (ibid).

Alsaqaf, Daneva och Wieringa (2019) beskriver att det som utvecklarna ser som utmaning är att fokusera på de "rätta" IF kraven att implementera utifrån dokumentationen eftersom vissa krav kan påverka varandra eller vara otydligt specificerade. Utvecklarna behöver därmed tillgång till ett helhetsperspektiv över alla kraven, både FK som IF krav, vilket återfinns i dokumentet kravspecifikation (se avsnitt 3.4.1) (ibid). Karhapää et al. (2021) beskriver att utvecklare behöver förstå "varför" ett krav ska implementeras eftersom de kan ha olika "hur" de kan göra det på, det vill säga olika tillvägagångssätt kan leda fram till samma funktionalitet. När det gäller IF krav är "varför" frågan viktig då det är vad som skapar värde för kunden (Karhapää et al. 2021).

Alsaqaf, Daneva och Wieringa (2019) beskriver att produktägaren av projektet vanligtvis prioriterar de IF kraven åt mjukvaruutvecklingsteamet och därmed kan prioritera dem utifrån sitt eget partiska synsätt över vad som anses vara mest effektivt. Detta gjorde att teamet ibland fick släppa alla IF krav de höll på att implementera om produktägaren ville att de skulle fokusera på andra IF krav som hen ansåg vara viktigare, särskilt inför en deadline.

3.2 Dokumentationens betydelse för IF krav

Behutiye et al. (2022) beskriver att dokumentation är det första som stryks om det förekommer tidsbegränsningar då fokus ligger mer på att leverera en fungerande mjukvara vilket resulterar i att dokumentationen lättare glöms bort. Medeiros et al. (2018) betonar vikten av att dokumentera eftersom det kan minimera kunskapens bortfall i en verksamhet. Författarna understryker dock att det är problematiskt att underhålla dokumentationen och att

ha motivation till att göra det. Att använda flertalet olika tillvägagångssätt för att dokumentera IF krav kan skapa en negativ inverkan på tilliten till exempelvis kravspecifikationen eftersom utvecklarna inte är medvetna om att kravspecifikationen är uppdaterad eller inte (ibid).

Medeiros et al. (2018) argumenterar att användning av en gemensam dokumentation kan göra det enklare för utvecklare och andra som är involverade i IF kraven att hitta ny och uppdaterad information utan att behöva gå igenom flera olika dokumentationsartefakter. Att använda en och samma gemensamma dokumentation skulle kunna leda till en komplett förståelse över vad som behöver implementeras, reducera komplexitet, mindre redundans och samtidigt effektiv delning av dokumentation till kunden. Jarzebowicz och Weichbroth (2021) beskriver att själva dokumentationen av IF krav inte uppfattas som problematiskt utan att det handlade om brist på verktyg för att stödja samarbete med kunden. Behutiye, Rodríguez och Oivo (2022) anser att det behövs fler riktlinjer för att dokumentera IF krav, eftersom befintliga riktlinjer anses vara otillräckliga för att minska dokumentationsbrist. Vidare menar de att IF kraven anses vara underförstådda, vilket resulterar att det inte finns behov av att dokumentera dem. Genom ett väldefinierat tillvägagångssätt för att dokumentera IF krav kan kvaliteten säkerställas men även hjälpa verksamheter med beslutsfattning (ibid).

Dokumentationen försäkras kvaliteten genom att stödja verifiering, det vill säga påvisa att alla IF krav har blivit implementerade. Alsaqaf, Daneva och Wieringa (2019) beskriver att om det inte finns definierade IF krav eller om de IF kraven är dåligt definierade så kan det resultera i att utvecklare gör egna antaganden. Dessa antaganden görs även när utvecklare är medvetna om de IF kraven men inte anser att de behöver verifieras för att de är för "självklara".

Behutiye et al. (2022) beskriver att tidsbegränsningar eller tidsbrist kan vara en bidragande orsak till att IF krav skrivs ner otydligt och felaktigt i dokumentationen. Otydlig och felaktig dokumentation kan i sin tur leda till att projektet behöver fler iterationer för att kunna försäkra kvaliteten.

Brist på dokumentation leder till försämrad kvalitet, ökade underhållskostnader och omarbetningar kan ses som teknisk skuld enligt Behutiye et al. (2020b). Behutiye et al. (2022) framför att dokumentation av IF krav som inte uppdateras kan leda till teknisk skuld och brist på gemensam förståelse. Behutiye, Rodríguez och Oivo (2022) framför att en typ av instruktionsmanual kan användas för att hjälpa till med dokumentationen av IF krav genom att erhålla kunskap och tillvägagångssätt för hur de IF kraven ska dokumenteras och implementeras. Medeiros et al. (2018) anser att det finns begränsat med tillvägagångssätt för att kunna beskriva IF krav men att inom den traditionella mjukvaruutvecklingen förespråkas det att använda olika artefakter för att dokumentera krav generellt för att kunna möta olika behov hos olika intressenter. Behutiye et al. (2020b) förespråkar även en mer flexibel ansats när det gäller dokumentation av IF krav. Det finns inte enbart ett tillvägagångssätt att göra det på utan att sättet kan variera beroende på mål och kontext. Behutiye et al. (2017) beskriver att IF krav beskrevs som både översiktligt det vill säga hela systemet eller i detalj vilket kan göra det svårt att hitta en representativ artefakt för dokumentation som är användbar för dem alla och på olika nivåer.

Den viktigaste aspekten vid dokumentation är klarheten av de IF kraven. Klarhet relaterar till att försäkra att rätt uppfattning och definition av IF krav finns, vilket bidrar till mindre feltolkningar vid dokumentation. Klarhet av IF krav bidrar även till att IF krav går att mäta,

övervaka och rapportera om fel skulle uppstå eftersom de IF krav görs mer spårbara (Behutiye, Rodríguez & Oivo 2022). Detta kan kopplas till det Glinz (2007) framför att det finns en brist på konsensus om vart dokumentationen av IF krav ska ske. Författaren förespråkar att dela upp de IF kraven och de FK för att det ska bli mer tydligt i dokumentationen.

Behutiye et al. (2022) argumenterar att det kan finnas kommunikationslyftor i ett team. Det kan handla om frånvarande medlemmar, olika förståelser och tolkningar av IF krav. Detta kan bero på att det finns brist på diskussioner kring IF krav. Genom att ha öppna diskussioner om IF krav bland teammedlemmar kan det minimera missförstånd och förbättra dokumentationen genom att revidera, klargöra och beskriva mer tydligare vad exakt som menas med det IF kraven. Klyftorna kan även förekomma om IF krav inte är specificerade eller överenskomna i ett tidigt stadiet av ett projekt. Teammedlemmar kan få för sig att dokumentera IF krav på eget sätt vilket kan leda till onödiga konflikter och förvirring (ibid).

IF krav kan klassificeras till olika kategorier, såsom funktionalitet, användbarhet, pålitlighet, säkerhet, effektivitet, underhållbarhet och portabilitet, med flera. De kan även benämnas som klassifikationer av IF krav enligt Eckhardt, Vogelsang och Fernandez (2016), som konstaterar att det finns väldigt många olika klasser, eller kategorier, av IF krav. De kategoriserade i sin studie 530 IF krav och jämför sina resultat med Glinz (2007) och Ameller et al. (2012) som också observerat att vissa kategorier av IF krav var svårare att definiera än andra. Eckhardt, Vogelsang och Fernandez (2016) drar slutsatsen att till exempel IF krav som tillhör kategorin tillgänglighet ofta missuppfattas i dokumentationen och används som kategorin prestanda istället. De kunde även dra slutsatsen att kategorierna prestanda, användbarhet, säkerhet och tillgänglighet var viktigast för mjukvaruarkitekter och även de mest omnämnda i litteraturen. Karhapää et al. (2021) menar att användbarhet och säkerhet är de två mest diskuterade kategorierna av IF krav i litteraturen. Enligt Jarzebowicz och Weichbroth (2021) kan kategorisering av IF krav främja insamlingen av IF krav. Men medan en kategorisering av IF krav kan vara till hjälp för att uttrycka avsikten med det IF kravet, ska det inte avgöra hur kravet specificeras, implementeras eller testas (Eckhardt, Vogelsand & Fernandez 2016).

3.2.1 Tillvägagångssätt för att dokumentera IF krav

Behutiye et al. (2022) redogör att dokumentation av IF krav är begränsad och utmanande i agila mjukvaruutvecklingsprojekt på grund av praxis att enbart föra informell dokumentation samt på grund av bristande kunskap om IF krav. Vidare identifierades behov av "*lightweight*" dokumentation av IF krav samt ledarskapsstrategier, eftersom det enligt författarnas studie fanns bristande riktlinjer för det. I den undersökning Behutiye et al. (2022) genomförde existerade lätt dokumentation, men det fanns en tendens att enbart fokusera på vissa IF krav, såsom säkerhet och användbarhet.

Nedan beskrivs de vanligaste tillvägagångssätten för att dokumentera IF krav idag.

Use Cases

Larman (2005) beskriver att use cases (UC) är scenarior av hur en användare interagerar med ett mjukvarusystem. UC beskrivs som ett krav, där målet specificeras i en modell som påvisar *vad* ett system ska göra. De beskrivs även som visuella textdokument som är enkla att förstå, speciellt för kunder. Medeiros et al. (2018) belyser att UC anses lätta att förstå men att det krävdes ansträngning att skapa dem samt kontinuerligt samarbete med kunden. Det som

ansågs problematiskt var att kunna specificera de tekniska aspekterna då ett UC kan innehålla mycket information och är mer anpassade till FK. Oftast bryts de mer översiktliga UC ner till mer detaljerade user stories (se avsnitt 3.5.3) (ibid).

Larman (2005) hävdar också att UC främst är avsedda för att dokumentera FK, som beskriver systemets beteende och funktionalitet. Han noterar dock att UC kan användas för att dokumentera vissa IF krav, såsom prestanda, tillförlitlighet och säkerhet. Larman (2005) föreslår vidare att IF krav kan integreras i UC genom att specificera dem som begränsningar för UCs beteende, det vill säga begränsningar mot de FK. Till exempel kan en prestandabegränsning inkluderas i ett UC för att specificera att systemet måste svara på användaråtgärder inom en viss tid. På liknande sätt kan en säkerhetsrestriktion inkluderas för att specificera att systemet måste autentisera användare innan de tillåter dem att utföra vissa åtgärder. Även om UC kan användas för att dokumentera vissa IF krav, noterar Larman (2005) att det inte är det bästa tillvägagångssättet för att dokumentera alla IF krav. Vissa IF krav, såsom skalbarhet eller underhållbarhet, dokumenteras bättre med tillvägagångssätt, såsom arkitekturmodeller eller i form av scenarion.

Kravspecifikation

Kravspecifikationen är skriven i ett naturligt textbaserat språk som beskriver de FK och IF kraven för ett mjukvarusystem. Målet är att specificera de krav och behov som kunden eller andra intressenter erhåller. Att använda sig av en kravspecifikation anses vara en olämplig form inom den agila mjukvaruutvecklingen eftersom i de projekten kan någon behöva revidera kraven ofta istället för att ta fram entydiga och fullständiga krav redan från början, vilket görs i en kravspecifikation (Ramesh, Cao & Baskerville 2010).

Karhapää et al. (2021) beskriver att kravspecifikationen bör tas fram i samråd med utvecklarna själva för att undvika feltolkningar och missförstånd. Oftast kommunicerar inte utvecklarna direkt med kunden själva. Det kan förekomma att det finns separata grupper som tar fram kravspecifikationen och om det saknas kommunikation och samarbete mellan dessa individer kan utvecklarna implementera fel saker vilket ökar kostnader och efterarbete. En mer detaljerad kravspecifikation inger en entydig och precis helhetsbild av alla krav som utvecklarna kan förlita sig på (ibid). Inledningsvis bör kraven ses som en startpunkt och definieras mer översiktligt då utvecklarna förväntas ha en övergripande förståelse initialt som blir mer detaljerad ju längre in i utvecklingen de kommer (Ramesh, Cao & Baskerville 2010).

Medeiros et al. (2018) beskriver att företag oftast är kunddrivna vilket kan resultera i en lång kravspecifikation. En lång kravspecifikation kan påverka dess kvalitet och försvåra förståelsen hos de som ska implementera de IF kraven. Det som kan vara problematiskt med en kravspecifikation är att hålla den uppdaterad med nuvarande och korrekta krav som även fokuserar på mer tekniska aspekter. Ett sätt att förbättra kvalitet med en kravspecifikation kan vara att lägga mer fokus på att specificera IF krav. Kravspecifikationen inom mjukvaruutvecklingsprojekt ansågs vara ett bra sätt att bevara kunskapen om projektet exempelvis om projektgrupper löstes upp eller att nya ansluter sig (ibid).

User Stories

User stories (US) är det mest använda tillvägagångssättet att dokumentera IF krav enligt Alsaqaf, Daneva och Wieringa (2019). US beskrivs i naturliga språk och är enkla textbaserade berättelser i ett dokument där viktiga element, FK och IF krav kan identifieras i

form av “vem”, “vad” och “för vad” och “varför” (Karhapää et al. 2021). Behutiye, Rodríguez och Oivo (2022) argumenterar att US är ett sätt att reducera fokus på kravspecifikationen och mer fokusera på att dokumentera och specificera kraven, både de FK och IF kraven, i nära samarbete med kunden. Författarna belyser även att IF krav i US ska specificeras utifrån användarens perspektiv och mål.

Behutiye et al. (2020b) motsätter sig användningen av US när IF krav ska dokumenteras eftersom det kan vara problematiskt att beskriva interaktionen mellan IF krav. Detta kan leda till att IF krav såsom användbarhet och säkerhet kan förbises samt att flera US kan länkas till flera olika IF krav. Till skillnad från Behutiye et al. (2020b) beskriver Jarzębowicz och Weichbroth (2021) att US är en bra och enkel form av att dokumentera IF krav eftersom det förenklar den direkta kommunikationen av IF krav mellan utvecklarna i ett projektteam. Detta kan kopplas till det Medeiros et al. (2018) belyser att US ska vara skrivna på ett enkelt sätt och innehålla relevant information. Dock är det svårt att skriva US om erfarenhet saknas. Karhapää et al. (2021) anser å andra sidan att det förekommer utmaningar med att specificera IF krav i US men att IF krav bör ses som tillägg till de FK medan Behutiye, Rodríguez och Oivo (2022) framför att US bör fokusera mer på FK eftersom det enligt Medeiros et al. (2018) saknar teknisk jargong.

Behutiye et al. (2022) kritiserar US som tillvägagångssätt för dokumentation av IF krav eftersom tillvägagångssättet inte anses vara tillräcklig för att kunna specificera och dokumentera säkerhets- och användbarhetskrav, vilka är bland de mest omtalade IF kraven i litteraturen som rör hantering av IF krav. De rekommenderar att använda artefakter som är skraddarsyddade efter just säkerhets- och användbarhetskrav för att kunna hantera de kraven. Alsaqaf, Daneva och Wieringa (2019) beskriver att agila team använder sig av kreativa lösningar när de utgår från US då detta tillvägagångssätt även kritiserar som en oanvändbar metod för att dokumentera de IF kraven på.

Scrum, Product Backlog och Features

Scrum är en agil mjukvaruutvecklingsmetod som består av korta iterativa sprintar vilket enligt Behutiye et al. (2020b) kan resultera i att viktiga IF krav förbises exempelvis användbarhetskrav. Jarzębowicz och Weichbroth (2021) beskriver att agil mjukvaruutveckling tenderar att försumma IF krav genom att istället fokusera på att inkrementellt leverera funktionella funktioner, även kallat *features*. Features är ofta huvudfokuset vid mjukvaruutveckling och de dokumenteras som FK. Ett systems IF krav påverkar inte features direkt. Karhapää et al. (2021) beskriver att Scrum inte adresserar säkerhetskrav och att dessa förbises desto mer då fokus läggs på funktionella features.

I Scrum specificeras alla krav som US i en Product Backlog (PB), vilket betyder att kraven skrivs som korta och beskrivande meningar i en PB. PB är en lista där kraven listas i sin prioriterade ordning. Alsaqaf, Daneva och Wieringa (2019) beskriver att användning av PB är ett sätt för agila mjukvaruutvecklingsprojekt att organisera och dokumentera sina krav där kontinuerliga ändringar och prioriteringar kan ske. Det förekommer även att flera olika PB används för att kunna inkludera olika synvinklar av de IF kraven. Mjukvaruutvecklingsteamet använder sig av PB när de ska få en översikt över vilka krav och features som ska implementeras (Behutiye et al. 2020b). Behutiye, Rodriguez och Oivo (2022) anser att de IF kraven bör dokumenteras i en separat PB för att kunna specificeras mer i detalj.

Features dokumenteras i PB tillsammans med US. Att använda features är ett informellt sätt att dokumentera krav genom att använda en lista på funktioner som ska implementeras (Karhapää et al. 2021).

Textbaserade verktyg

Att använda Word, Excel, Wiki page eller Powerpoint för att dokumentera IF krav kan uppfattas som utmanande då dessa verktyg anses vara friliggande ifrån andra mjukvaror vilket kan påverka spårbarheten av IF krav. Det kan bli svårt att komma ihåg att uppdatera ett visst dokument vid ändring av något krav då dessa dokument behöver manuella ändringar (Behutiye et al. 2017). Wiki page är interna dokument där de IF kraven kan dokumenteras. Det kan även användas som ett sätt för utvecklare att skriva antagande om IF krav när det fanns oklarheter gällande de IF kraven, exempelvis när produktägaren har specificerat dem på ett oklart sätt i en kravspekifikation (Alsaqaf, Daneva & Wieringa 2019).

Confluence är en typ av mjukvara med principen av en Wiki page. Confluence används som ett samarbetsverktyg där kunskap, projektrelaterad information och idéer kan delas på ett effektivt sätt mellan medarbetare (Atlassian 2023).

3.3 Faktorer som påverkar kvalitetssäkring av IF krav

Behutiye et al. (2022) framför att uppfattning och dokumentation av IF krav är viktiga faktorer som bidrar till att säkerställa kvalitetsmålen som kunden har, men även ett sätt att se till att varje IF krav når upp till den nivå av kvalitet som är acceptabel. IF krav påverkar mjukvarans kvalitet och om de inte är tillräckligt definierade kan det leda till ökade kostnader och revideringar som i slutändan kan leda till svåra konsekvenser för hela projektet. Författarna iakttar att kvaliteten på IF krav är beroende av vilken kontext det används i. Funktioner som används mer sällan behöver inte ha lika hög nivå av kvalitet som exempelvis en funktion som är säkerhetskritisk och dessutom används dagligen. Det är därför viktigt att ha en full förståelse över helheten av de IF kraven; att se vilka IF krav som hänger ihop, varför de hänger ihop och vilka som eventuellt kan påverka varandra (Behutiye et al. 2022; Karhapää et al. 2021).

Karhapää et al. (2021) poängterar att om kunden är involverad i mjukvaruutvecklingen från början till slut kan denne validera att kvaliteten upprätthålls efter den överenskomna standarden. Validering är något som görs av kunden för att se om dennes behov har uppnåtts, det kan dock medföra mindre fokus på dokumentation av IF krav (Ramesh, Cao & Baskerville 2010), vilket det Agila Manifestet (2001) förespråkar. Behutiye et al. (2022) argumenterar å andra sidan att dokumentation av IF krav hjälper till att definiera vilka kvalitetsmål kunden vill ha och på samma sätt agerar som en guide för hur arbetet ska genomföras.

Å andra sidan beskriver Karhapää et al. (2021) att det finns utmaningar med att ha kontakt med kunden vid insamling av IF krav, då de inte har tillräcklig kunskap om de tekniska detaljerna, de är inte alltid tillgängliga, de kan ha annorlunda prioriteringar än mjukvaruutvecklingsteamet samt att det finns risker för kultur- och språkbarriärer som hindrar tydlig kommunikation. Utmaningarna med att ha kontakt med kunden kan öka risken för omarbete. Insamlingen av IF krav kan vara svår att konkretisera, fånga in och mäta vilket gör dem svåra att definiera och dokumentera. Insamlingen av IF krav görs oftast verbalt men beroende på situation och kunskap kan personen som samlar in kraven bland annat

produktägaren skriva ner dessa för eget bruk för att kunna spara tid och föra en mer detaljerad översikt (ibid).

Behutiye et al. (2022) beskriver att tidsbegränsningar eller tidsbrist kan vara en bidragande orsak till att IF krav skrivs ner otydligt och felaktigt i dokumentationen. Otydlig och felaktig dokumentation kan i sin tur leda till att projektet behöver fler iterationer för att kunna försäkra kvaliteten. Behutiye et al. (2022) belyser att kvalitet tar tid att uppnå och att det krävs att problem hanteras genom hela mjukvaruutvecklingen för att kunna säkerställa att rätt kvalitet levereras till kund. Författarna föreslår att juniora utvecklare och övriga intressenter med begränsad kunskap och erfarenhet av IF krav kan dra nytta av att använda en checklista av alla kvalitetssäkringar som bör göras, eftersom det kan vara svårt att identifiera de IF kraven utan ett helhetsperspektiv och begränsad kunskap om IF krav. Vidare argumenterar författarna att använda ett gemensamt och standardiserat sätt att definiera och se på IF krav som kan hjälpa till vid dokumentationen av dem, vilket i sin tur skulle vara en bidragande faktor till att kvaliteten säkerställs kontinuerligt genom hela mjukvaruutvecklingen av samtliga inblandade (ibid).

3.3.1 Säkerställning av kvalitet enligt svensk standard (SS-EN ISO 9001:2015)

Enligt svensk standard (SS-EN ISO 9001:2015) innebär säkerställning av kvalitet ett strategiskt beslut som syftar till att förbättra, öka förståelsen för intressenternas behov och förväntningar samt öka kundnöjdheten. Det handlar om att ha en kontinuerlig och riskbaserad inställning som främjar en mer hållbar utveckling inom organisationen. Det finns några viktiga principer som styr hur arbetet med kvalitet ska ledas i en organisation.

Ledarskapet spelar en avgörande roll i att främja förbättringar, engagera och stödja både medarbetare och intressenter. Att ha ett kundfokus genom hela organisationen är av stor betydelse, där målet är att se till att kunderna både förstår vad de kan förvänta sig och att deras nöjdhet upprätthålls. Det krävs noggrann planering för att uppnå förbättringar, minska oönskade effekter och förstå de önskvärda effekterna. De kvalitetsmål som fastställs bör vara mätbara, kommunicerade, uppdaterade och relevanta för vad som överenskommit med kunden. Det är viktigt att ha en tydlig plan för vad som ska göras, vem som ansvarar för det, vilka resurser som krävs, när det ska vara klart och hur resultatet ska utvärderas (SS-EN ISO 9001:2015).

Det är även av betydelse att organisationen säkerställer att medarbetarna har rätt kompetens och får lämplig utbildning, erfarenhet och träning för att kunna utföra sina arbetsuppgifter inom organisationen. Om det behövs, bör åtgärder vidtas såsom vidareutbildning, mentorskap eller anställning av mer kompetent personal. Kommunikation är en mycket relevant aspekt inom en organisation, och det bör finnas tydliga riktlinjer för när, med vem, vad, hur och vem som ansvarar för kommunikationen. Det är viktigt att dokumentera informationen på ett tydligt och bestämt sätt, att hålla dokumentationen uppdaterad vid revideringar och att den är tillgänglig när den behövs. Tidigare versioner av dokumentationen kan bevaras, och de bör skyddas på lämpligt sätt (SS-EN ISO 9001:2015).

3.4 Ansvarsroller för att kvalitetssäkra IF krav

Inom den agila mjukvaruutvecklingen förväntas utvecklare att ha kontinuerlig kommunikation med kund för att samla in, utveckla och implementera de IF kraven (Alsaqaf, Daneva & Wieringa 2019).

Behutiye et al. (2022) beskriver att personer med vissa roller, exempelvis produktägare eller projektledare, bör intressera sig för IF krav och ha bra kunskap om dessa, då de måste kunna avgöra om en produkts kvalitetsnivå är uppfylld och därmed säkerställa produktens beredskap. Detta intresse för de IF kraven finns dock inte alltid hos utvecklarna. Vidare menar Behutiye et al. (2022) att dokumentation av IF krav som utförs av chefsroller på högre nivåer kan påverka hur utvecklare på lägre nivåer dokumenterar och implementerar IF krav.

Både mjukvaruutvecklingsteamet och ledningen, såsom produktägare och chefer, bör ha insikt i hur viktigt det är med IF krav. Det är ledningens ansvar att vidta åtgärder som stödjer ökad kunskap om IF krav och dokumentation av dem. Författarna hänvisar till att investera i utbildning om IF krav eller att skapa en guide med riktlinjer att utgå ifrån som beskriver och förklarar de olika kategorierna av IF krav samt hur de kan hanteras och dokumenteras, med fokus på kvalitetssäkring. Faktumet att det råder brist på expertis är en utmaning för att kunna dokumentera och hantera IF krav eftersom projektledare och produktägare inte kan definiera dem tillräckligt väl, inklusive relevanta IF krav såsom säkerhet och användbarhet (ibid). Vidare konstaterar Behutiye et al. (2022) att dessa personer är benägna att glömma av de IF kraven. Av dessa anledningar bör särskilda roller tillsättas för att öka kompetenskravet och kunna kvalitetssäkra de IF kraven, såsom säkerhetsexperter, interaktionsdesigner eller användbarhets-experter. Författarna hänvisar även till att tillsätta ett specifikt team för just hantering och dokumentation av IF krav.

Behutiye et al. (2020b) beskriver att olika roller är ansvariga för att dokumentera de IF kraven såsom produktägaren, projektledaren, utvecklaren, produktchefen, säljteam, utvecklingschefer eller användbarhetsdesigner. Även kunden dokumenterade IF krav och kvalitetsfrågor men gjorde detta i ett annat system än leverantörerna (Behutiye et al. 2020b). Produktägaren har det huvudsakliga ansvaret att dokumentera och hantera specifikationen av IF krav, med hjälp och stöd av övriga inblandade roller. De övriga inblandade medarbetarna, utvecklare, säljteam och analytiker anger sina åsikter om både de FK och IF kraven till produktägaren. Denne analyserar sedan kraven utifrån medarbetarnas åsikter för att kunna förtydliga och specificera kraven. En senare studie av Behutiye et al. (2022) hävdar att dokumentationsproblem kan undvikas om det finns tydlig roller, kommunikation och ansvarsfördelning för hur de IF kraven ska hanteras.

Behutiye et al. (2020b) argumenterar att en utvecklares erfarenhet kan påverka dokumentationen av IF krav. Har utvecklarna erfarenhet, ett samarbetsvilligt tillvägagångssätt att arbeta på samt arbetar i ett mindre företag behövs inte de IF krav specificeras eller dokumenteras. Erfarna utvecklare förväntas ha kännedom om IF krav. Detta kan kopplas till det Behutiye et al. (2022) beskriver att det bör finnas utbildning för de juniora utvecklarna om hur de IF krav ska hanteras, dokumenteras och implementeras men att utvecklarna inte alltid behöver ha en högre nivå av kunskap om de IF krav eftersom förståelsen av IF krav är en kvalifikation som projektledare och produktägare bör fokusera på.

Enligt Ameller et al. (2012) är mjukvaruarkitekten den person som tar design och tekniska beslut i ett mjukvaruutvecklingsprojekt eftersom hen besitter både tekniska kompetens och erfarenhet för att kunna ta arkitekturella beslut som kan påverka hela mjukvarusystemet. På

grund av mjukvaruarkitektens kompetenser anses den vara mest lämpad att samla in och dokumentera de IF krav eftersom bestämmelsen över de IF krav är en iterativ process genom hela mjukvaruutvecklingen. När kunden själv tar fram IF krav agerar mjukvaruarkitekten som en aktiv stödjande part för att kunna förklara betydelsen och definitionen av de IF kraven (ibid).

3.5 Sammanfattning av det teoretiska ramverket

Sammanfattningsvis har IF krav en avgörande roll inom mjukvaruutveckling. Behutiye et al. (2020b) belyser att IF krav innebär de kvalitetsattribut som exempelvis prestanda eller användbarhet av mjukvaran. Forskningen visar att IF krav är svåra att definiera och har en lägre prioritet i jämförelse till FK i mjukvaruutvecklingsprojekt. Faktum är att IF krav bortprioriteras i större utsträckning, särskilt inom det agila projekt. De IF kraven dokumenteras inte alltid som enskilda eller särskilda krav utan är ofta relaterade till FK, vilket kan leda till bortprioritering (Alsaqaf, Daneva & Wieringa 2019).

Det används olika tillvägagångssätt för att dokumentera IF krav såsom UC, US, kravspecifikation, scrum, PB, features och textbaserade artefakter. För att samla in dessa krav används bland andra metoder som möten med kund och intervjuer. Det är oftast projektägaren eller projektledaren som håller kontakt med kunden för att samla in alla krav. Det finns olika utmaningar när det gäller insamling av IF krav (Behutiye et al. 2020b). Behutiye et al. (2020b) har påvisat att kunden oftast saknar kunskap om IF krav och detta leder till kommunikationssvårigheter och misstolkningar. Vidare har Karhapää et al. (2021) påpekat att även utvecklarna, särskilt de juniora utvecklare, har bristande kunskaper när det gäller IF krav. Författarna föreslår involvering av kunden för att öka deras förståelse och därmed påvisa vikten av IF krav i ett mjukvarusystem.

Flera författare har pekat ut det faktum att IF krav hanteras med tysta kunskap. Den tysta kunskapen betraktas som den intuitiva förmåga som särskilt seniora utvecklare besitter. Det har visat sig att inom den agila mjukvaruutvecklingen så förespråkas mindre dokumentation och snabb leverans av fungerande mjukvara prioriteras mest. Flera författare som Behutiye et al. (2017) och Karhapää et al. (2021) anser att det skapar problematik kring dokumentation av IF krav som kan påverka mjukvarans kvalitet. För att mjukvaruutvecklingsprojekt skall lyckas och att kunden ska få mjukvara med rätt kvalitet borde personer som arbetar med kravinsamling och kravdokumentation ha förståelse för de IF krav och behärska kunskap om vikten av IF krav inom mjukvaruutveckling. Medeiros et al. (2018) beskriver att dokumentation av IF krav minimerar kunskapens bortfall. Dessutom underlättar en gemensam dokumentation att ha information om IF krav tillgänglig för alla som arbetar inom mjukvaruutvecklingsprojekt.

Det är viktigt för ett mjukvaruutvecklingsprojekt att utvecklarna förstår vikten av IF krav och att den tysta kunskapen hos framför allt seniora utvecklare dokumenteras. Dokumentation av IF krav bidrar till att kunskapen om IF krav föras vidare och en gemensam förståelse hos utvecklarna och de som arbetar med IF krav skapas. Behutiye, Rodriguez och Oivo (2022) belyser att den viktigaste aspekten för att dokumentera krav är klarheten av dessa krav. Författarna har även påpekat att det behövs expertis inom insamling och dokumentation av IF krav. Behutiye et al. (2022) sammanställer att både mjukvaruutvecklingsteamet och ledningen borde ha kunskap om IF krav och förstå vikten av de inom mjukvaruutveckling. Då det råder brist på expertis för att kunna hantera IF krav borde det fastställas särskilda roller med relevant kompetens för att kunna säkerställa IF kraven.

4 Resultat

I följande kapitel presenteras resultatet av den insamlade empirin från enkätundersökningen. Enkätfrågorna visas och dess svarsalternativ. Se bilaga 1 för en fullständig beskrivning av enkäten med de olika svarsalternativen. Resultaten presenteras både i text och bild på frågor 10-24 där förtydligande behövs. Den univariata analysen genomfördes för att beskriva den procentuella fördelningen av respektive fråga. Respondenterna som deltog i enkätundersökningen arbetar med kravhantering. Deras roller och ansvarsområden varierar. Totalt samlades 43 svar in.

4.1 Univariat analys

Fråga 10: Nedan följer några exempel på krav. Markera det(t) alternativ du anser är icke-funktionella krav

Följande krav användes för att avgöra hur väl respondenterna kunde urskilja IF krav från FK. Se asterix-markering för korrekta svar (som inte angavs i enkäten).

- *Obehöriga ska inte kunna komma in i systemet**
- *Systemet ska uppfattas som seriöst och tillförlitligt**
- *Systemet ska vara kompatibelt med våra andra system**
- *Ska kunna lägga till nya kunder***
- *En användare ska få bekräftelse-mejl efter ett köp***
- *Man måste kunna återanvända koden för att bygga ett nytt system**
- *Systemet ska kunna skapa statistik över sålda produkter***
- *Ska kunna lägga till nya kunder***

* IF krav

** FK

Vissa kategorier var enklare att identifiera än andra. Nedan redovisas ordningen av lätthet att identifiera vardera kategorien:

- 83,3 % av respondenterna angav "*Systemet ska uppfattas som seriöst och tillförlitligt*". Kravet kan kategoriseras som tillgänglighet/pålitlighet (availability/reliability).
- 71,4 % av respondenterna angav "*Man måste kunna återanvända koden för att bygga ett nytt system*". Detta kan kategoriseras som underhållbarhet (maintainability)
- 64,3 % av respondenterna angav "*Obehöriga ska inte kunna komma in i systemet*" som kan kategoriseras som säkerhet (security)
- 57,1 % av respondenterna angav "*Systemet ska vara kompatibelt med våra andra system*" som kan kategoriseras som kompatibilitet/överförbarhet (compatibility/portability)

Följande var anmärkningsvärt: en person (2,3 %) hade fyllt i alla alternativ som IF krav och en annan person (2,3%) hade inte fyllt i något alternativ. 16,2 % av respondenterna hade valt två alternativ varav allas alternativ var IF krav. 23,3% av respondenterna valde endast antingen en eller två av de svaren som flest kunde ange och lämnade resten okryssade.

Sammanfattningsvis hade 91 % av respondenterna mellan 1-4 rätt men angav samtidigt även ett eller flera felaktiga svarsalternativ. 21 % av respondenterna hade svarat exakt rätt på frågan om vilka de IF kraven var av alternativen. Dessa respondenter var:

- Respondent 11, 51-65 år magisterexamen i orelaterat ämne, applikationskonsult med över 11 års erfarenhet, uppfattade IF krav som otydliga.
- Respondent 17, 31-40 år, kandidatexamen i data/informatik, kravanalytiker/BA, 8-10 års erfarenhet, uppfattade IF krav som väldigt tydliga.
- Respondent 19, 51-65 år, kandidatexamen i data/informatik och ledarskap, projektledare, 2-4 års erfarenhet i nuvarande roll, uppfattade IF krav som väldigt tydliga.
- Respondent 28, 31-40 år, masterexamen, data/informatik och ingenjör, CTO, 5-7 år i nuvarande roll, uppfattade IF krav som väldigt tydliga.
- Respondent 29, 31-40 år, kandidatexamen, data/informatik, mjukvaruansvarig, 0-1 år i nuvarande roll, uppfattade IF krav som väldigt tydliga.
- Respondent 31, 51-65 år, självlärd eller kurser inom data/it, kravanalytiker/projektledare med över 11 års erfarenhet, uppfattade IF krav som väldigt tydliga.
- Respondent 36, 51-65 år, självlärd eller kurser inom data/informatik, verksamhetsutvecklare med mindre än 1 år i nuvarande roll, uppfattade IF krav som väldigt tydliga.
- Respondent 40, 40-51 år, kandidat i data/IT, kravanalytiker, BA, 5-7 års erfarenhet, uppfattade IF krav som väldigt tydliga.
- Respondent 43, 51-65 år, master inom data/IT, arkitekt med över 11 års erfarenhet, , uppfattade IF krav som väldigt otydliga.

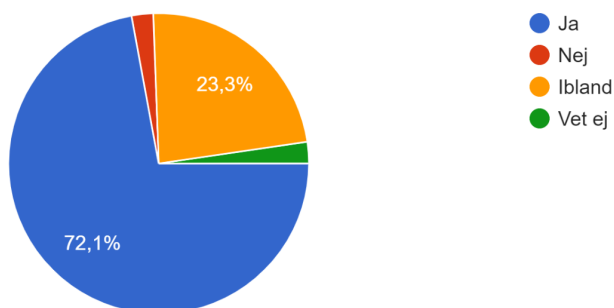
Majoriteten (55%) av de som svarat alla rätt var mellan 51-65 år gamla och hade roller såsom arkitekt eller applikationskonsult med över 11 års erfarenhet, eller kortare erfarenhet i nuvarande roll som verksamhetsutvecklare eller projektledare, vilka båda är yrken som kräver senioritet eller gedigen erfarenhet för att kunna anta.

33% var 31-40 år gamla men hade större ansvarsroller såsom mjukvaruansvarig på ett stort företag, CTO på ett medelstort företag, samt kravanalytiker/BA med 8-10 års erfarenhet.

Ingen var under 31 eller över 65 år. Utbildningsnivån verkar inte spela någon roll då det varierade mellan självlärd/kurser till masterexamen. Ingen hade endast gymnasieexamen. Dock var samtliga utom 1 respondent utbildade inom data/informatik, ibland kombinerat med andra ämnen också.

2 av de 5 respondenter som aldrig tidigare hade jobbat med krav ingick i dessa 21% som kunde felfritt identifiera de IF kraven, därför verkar inte tidigare erfarenhet ha någon relevans för hur väl man kan identifiera IF krav.

Fråga 11: Dokumenterar ni krav i det företag du arbetar på idag?



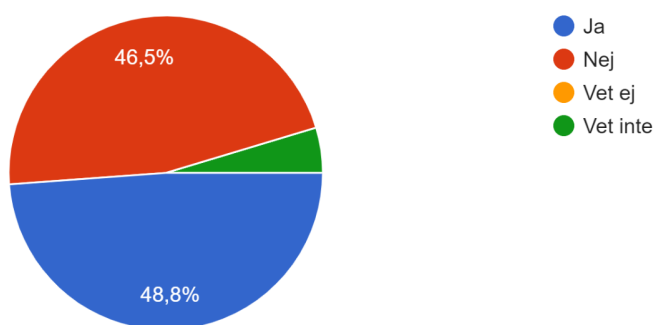
Figur 9: Cirkeldiagram över procentuell fördelning över dokumentation av krav inom respondenternas nuvarande arbetsplats

Det framgår av figur 9 att 72,1 % av respondenterna dokumenterar krav idag medan 23,3 % framför att de gör det ibland. Vidare hade 2,3 % angett att de inte gör det och 2,3 % framförde att de inte vet om de dokumenterar krav eller inte.

Den andel (2,3 %) som svarade "vet ej" motsvarar en respondent. Denna respondent har angett att hen är självlärd eller tagit enstaka kurser, jobbar som applikationskonsult och har jobbat i över 11 år på ett stort företag. Hen vet inte om de tidigare arbetat med IF krav, tycker det är lite otydligt med IF krav, samt är neutral både i frågan om IF krav är viktiga och om det är viktigt att dokumentera IF krav.

Sammanfattningsvis visar resultatet på denna fråga att majoriteten av respondenterna, vilket var 72,1 % av samtliga respondenter, utför dokumentation av krav på deras nuvarande arbetsplats.

Fråga 12: Definierar ni vilka krav som är funktionella och icke-funktionella vid dokumentation?

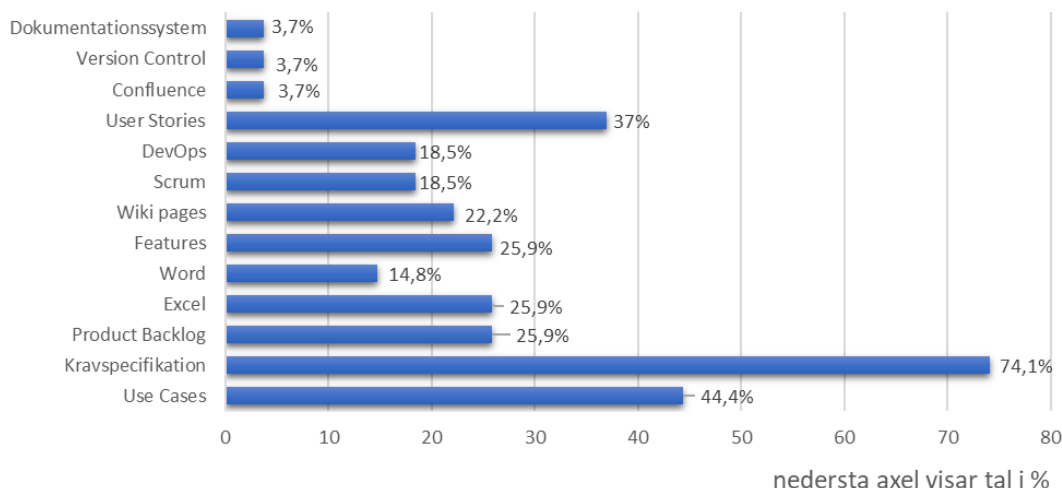


Figur 10: Cirkeldiagram över procentuell fördelning över respondenternas medvetenhet om icke-funktionella krav

Det framgår av figur 10 att 48,8 % av respondenterna säger på och definierar vilka krav som är FK och IF krav vid dokumentation. 46,5 % definierar inte vilka krav som är FK och IF krav vid dokumentation och samlar därmed alla krav som krav. 4,7 % svarade "vet ej". Detta visar att en nästan lika stor andel av respondenterna som antingen definierar eller inte de IF

krav, dock visade det sig att majoriteten av respondenterna definierar de IF krav när de dokumenterar alla krav.

Fråga 13: Om du svarade ja på föregående fråga, hur dokumenterar ni icke-funktionella krav idag? (Flera val är möjliga)



Figur 11: Stapeldiagram över procentuell fördelning av vilka tillvägagångssätt som används för att dokumentera IF krav

Det framgår i svaren på frågan att endast 62,8 % av respondenterna har svarat på frågan, vilket tyder på att 37,2 % inte har svarat på frågan.

Av dessa 62,8 % (27 st) som besvarat frågan har:

- 7,4 % har svarat sju olika sätt att dokumentera
- 7,4 % har svarat sex olika sätt att dokumentera
- 11,1 % har svarat fem olika sätt att dokumentera
- 11,1 % har svarat fyra olika sätt att dokumentera
- 18,5 % har svarat tre olika sätt att dokumentera
- 22,2 % har svarat två olika sätt att dokumentera
- 22,2 % har svarat ett olika sätt att dokumentera

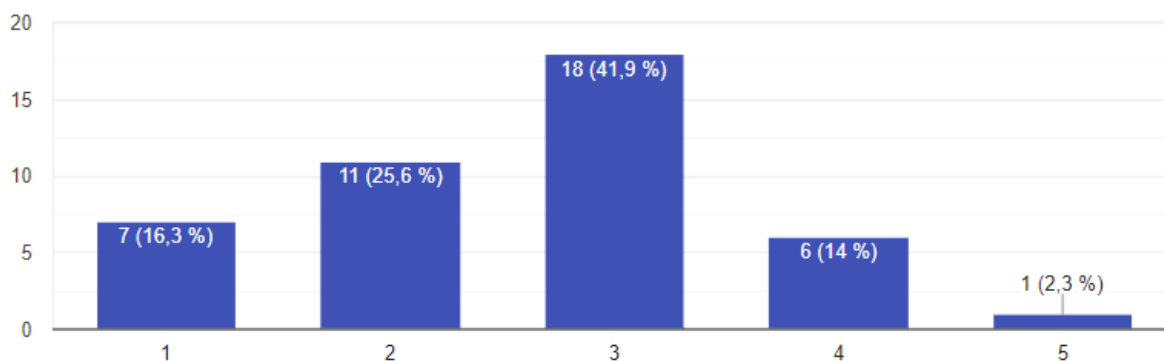
Det framgår av figur 11 att 74,1 % använder kravspecifikationen som sitt tillvägagångssätt för att dokumentera IF krav följt av use cases på 44,4 % och user stories på 37 %. Backlog, Excel, features har vardera 25,9 %. 22,2 % använder wiki pages medan 18,5 % vardera använder Scrum och DevOps. 14,8 % använder Word och sedan finns det enstaka som använder andra dokumentationssystem, såsom Confluence och version control. Medelvärde för hur många olika tillvägagångssätt som används är tre stycken. Sammanfattningsvis använder respondenterna olika kombinationer av tillvägagångssätt för att dokumentera IF krav.

Fråga 14: Är ovanstående svar er standard för att dokumentera icke-funktionella krav, eller varierar era tillvägagångssätt från projekt till projekt?

Enligt fråga 14 angav 71,4 % av respondenterna att tillvägagångssätten kunde variera från projekt till projekt medan 28,6 % angav att det är deras standard-tillvägagångssätt att dokumentera IF krav. 18,6 % av respondenter besvarade inte frågan. Detta visar att

majoriteten av respondenterna saknar standard för att dokumentera IF krav och att deras sätt att dokumentera dessa varierar vanligtvis från projekt till projekt.

Fråga 15: Till vilken grad är det tydligt vad som ska implementeras från de icke-funktionella kraven?

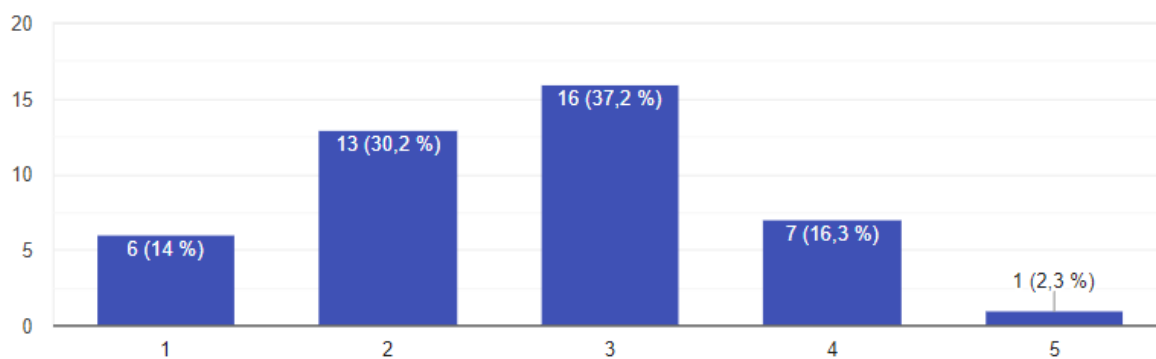


Figur 12: Stapeldiagram över procentuell fördelning av hur tydligt det är att utläsa vad som ska implementeras ifrån de IF krav. Skala 1 står för "Väldigt tydligt" och skala 5 står för "Väldigt otydligt".

Det framgår av figur 12 att 16,3 % ansåg att det är väldigt tydligt vad som ska implementeras från de IF kraven, medan endast 2,3 % ansåg tvärtom att det är väldigt otydligt. 14 % av respondenterna svarade att det är otydligt och 25,6 % angav tvärtom att det är tydligt. Den största gruppen, 41,9 %, har antingen en neutral åsikt eller en delad uppfattning.

Majoriteten svarade en 3:a på skalan vilket betyder att de antingen har en neutral åsikt i frågan eller en delad åsikt – det är inte helt otydligt för dem, men inte helt tydligt heller. Därefter har de flesta av respondenterna svarat antingen en 1:a eller 2:a, vilket är antingen tydlig eller väldigt tydlig förståelse. Sammanlagt har 41,9 % svarat att det är tydligt vad som ska implementeras, vilket är exakt samma andel som angivit en neutral åsikt. Minoriteten har angett 4:a eller 5:a (16,3 %), vilket indikerar att minoriteten av respondenterna ansåg att det är otydligt kring vad som ska implementeras från IF krav.

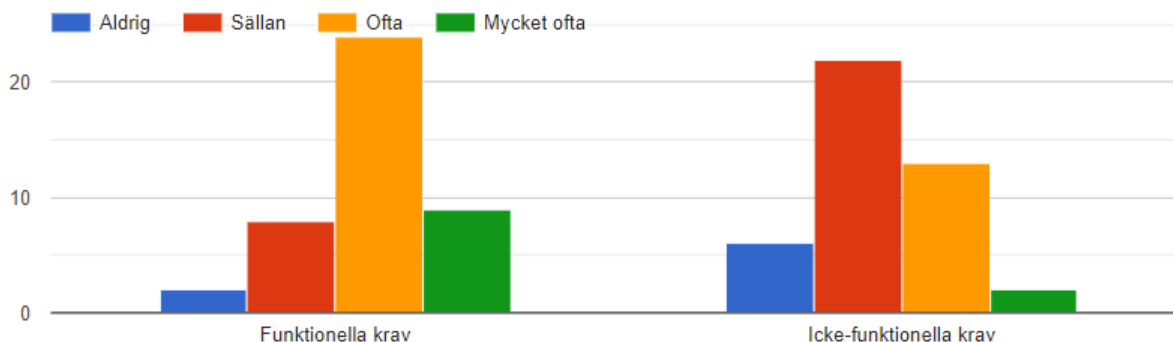
Fråga 16: Till vilken grad är det lätt/svårt för dig att dokumentera vad som ska implementeras från de icke-funktionella kraven?



Figur 13: Stapeldiagram över procentuell fördelning av hur lätt/svårt det är att dokumentera vad som ska implementeras ifrån de IF krav. Skalan visar 1= Väldigt svårt och 5 = Väldigt lätt

Det framgår av figur 13 att 14 % anser att det är väldigt lätt att dokumentera vad som ska implementeras från de IF krav (skala 1) medan endast 2,3 % anser att det är väldigt svårt (skala 5). 30,2 % angav att det är lätt (skala 2); och 37,2% angav en neutral eller delad åsikt (skala 3). 16,3 % angav att är svårt att dokumentera vad som ska implementeras från de IF kraven (skala 4). Detta visar att 44,2 % har det lätt gällande dokumentation av IF krav och 18,6 % har det svårt att dokumentera IF krav.

Fråga 17: Hur ofta händer det att ni reviderar kraven (tillsammans med kund) innan ni går vidare till implementationsfasen?



Figur 14: Stapeldiagram över procentuell fördelning av hur frekvent revidering av både FK och IF krav sker

Det framgår av figur 14 att de FK reviderades:

- Aldrig = 2,4 %
- Sällan = 18,6 %
- Ofta = 55,8 %
- Mycket ofta = 20,9 %

De IF krav reviderades:

- Aldrig = 13,9 %
- Sällan = 51,2 %
- Ofta = 30,2 %
- Mycket ofta = 2,4 %

Sammanfattningsvis visar resultaten att FK reviderades ofta medan IF krav reviderades sällan.

Fråga 18: Vad utgår du ifrån när du implementerar icke-funktionella krav? (När du väl sitter och ska börja implementera). Ange det svaret som är vanligaste utgångspunkten för dig. Om frågan ej är applicerbar på din roll kan du skippa frågan.

Det framgår av fråga 18 att flera av respondenterna har svarat med eget alternativ om vad de utgår ifrån när de ska implementera de IF kraven. 13,9 % (6 personer) valde att inte besvara frågan alls. För att enklare kunna överblicka svaren trots många individuella svar, har de grupperats enligt följande:

- 1) Baserat på skriftlig information - 50 %
 - Ett dokument med alla krav som jag följer och kontrollerar. (12)
 - Ett dokument. (7)
 - Checklista kopplat till varje uppdrag. (1)
 - Beror på krav, men måste ett feature ut snabbt kanske man kan tumma lite på test coverage etc. (1)
- 2) Baserat på muntlig/teaminformation - 16 %
 - Implementerar inte kraven själv, säkerställer att teamet gör det. (1)
 - Gruppmöte med vårt team. (4)
 - Gruppmöte med vårt team som bestämmer vad som ska göras, sen gör jag det. (2)
- 3) Kombination av muntligt och skriftligt - 7%
 - En kombination av muntligt teamarbete och skriftlig dokumentation i features eller kravspecifikation. (1)
 - Både skriftligt och verbalt. (1)
 - Både dokument och gruppmöte. (1)
- 4) Baserat individuell kännedom och minne - 12 %
 - Många olika parametrar. Kvalitetssystem, egna erfarenheter i teamet etc. (1)
 - Jag minns informationen/jag har kraven i huvudet och utgår från det när jag implementerar. (2)
 - Jag har det i huvudet. (2)
- 5) Ej besvarad
Sex personer (13,9 %) besvarade inte frågan alls, andra gav svar som inte besvarade frågan. Här ingår svar från respondent 11 "Startpunkten är att kravnivån för ändringar på och tillägg till ett ERP-system är homogenitet." (1)

Sammanfattningsvis visar resultatet av denna fråga att majoriteten av respondenterna har angett att de utgår ifrån dokumentationen vid implementering av IF krav. Det är även intressant att veta hur respondenternas senioritet påverkar vad de utgår ifrån. Detta presenteras i figur 16 där respondenterna har grupperats efter dess antagna senioritet, som analyserades i avsnitt 2.4.2 men inte är definitiv eller bekräftad, utan en sammansatt uppfattning av deras erfarenhet, kompetens och utbildning.

	Junior	Mid-level	Senior
Skriftligt	41,6 %	80 %	63,1 %
Muntligt	16,6 %	0 %	26,3 %
Skriftligt+muntligt	16,6 %	0 %	5,2 %
Kännedom/minne	25 %	20 %	5,2 %
Totalt:	33%	14%	53%

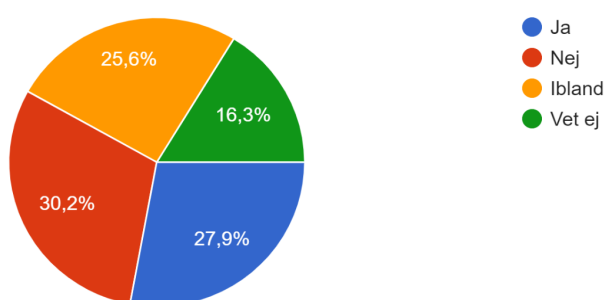
Figur 15: Procentuell fördelning av hur många respondenter av vardera seniorfördelningen svarat. Totalt svarade 12 av 14 juniorer, fem av sju mid-levels och 19 av 22 seniorer. Procentandel har angetts baserat på de respondenter som svarat.

Enligt figur 15 är den grupp som till störst del förlitar sig på sin egen kännedom eller minne de mer juniora respondenterna (25%), följt av en närliggande andel mid-level respondenter (20 %) men de seniora respondenterna förlitar sig allra minst på sin egna kännedom eller minne (5,2 %).

En tydlig majoritet av de seniora respondenterna (63,1 %) förlitar sig på skriftliga källor att utgå ifrån när de ska implementera IF krav. Om man dessutom adderar andelen som svarat en kombination av skriftligt och muntligt är den totala andelen 68,3 % av de seniora respondenterna som förlitar sig på antingen bara skriftligt eller både skriftligt och muntligt. För att göra samma analys av de juniora respondenterna är det sammanlagt 58,2 % som förlitar sig på antingen skriftligt eller skriftligt och muntligt. De juniora förlitar sig på det skriftliga minst av alla grupper, inklusive mid-level respondenterna som endast svarat skriftligt (80%) eller egen kännedom (20%).

Samtliga grupper har allra oftast svarat att de förlitar sig på skriftliga källor.

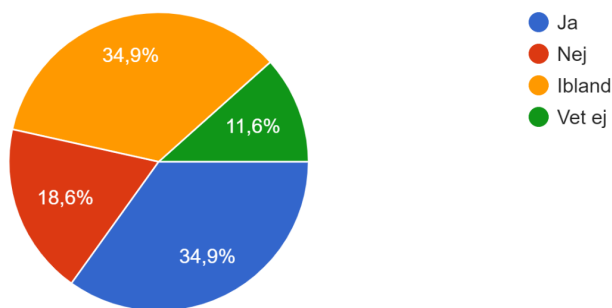
Fråga 19: Har någon ett översiktligt ansvar specifikt för dokumentation av de icke-funktionella kraven?



Figur 16: Cirkeldiagram över procentuell fördelning över respondenternas ansvar för dokumentation av icke-funktionella krav

Det framgår av figur 16 att 27,9 % av respondenterna angav att någon har ett specifikt översiktligt ansvar för dokumentationen av de IF krav, medan 30,2 % av respondenterna angav att ingen har ett översiktligt ansvar för det, vilket är den största svarsgruppen. 25,6 % angav att någon har det ibland; och 16,3 % vet inte om någon har det. Detta visar att majoriteten av respondenterna (72,1 %) inte tilldelar något specifikt ansvar för dokumentation av IF krav per standard, det vill säga de som angett nej, ibland eller vet ej.

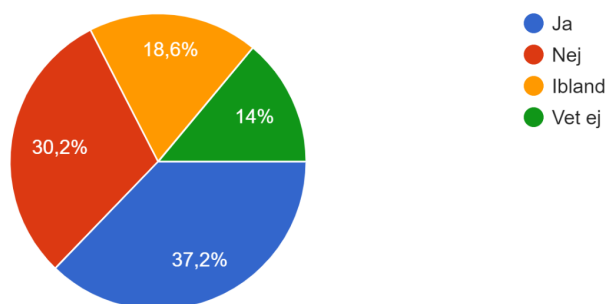
Fråga 20: Fördelar ni ansvaret för icke-funktionella krav mellan olika personer?



Figur 17: Cirkeldiagram över procentuell fördelning över respondenternas ansvar för icke-funktionella krav

De framgår av figur 17 att 34,9 % fördelar ansvar för de IF krav mellan olika personer medan 18,6 % inte gör det. 34,9 % gör det ibland och 11,6 % vet inte om fördelning av ansvar sker eller inte. Detta visar att lika stor andel av respondenterna angett att de antingen fördelar ansvar mellan olika personer för IF krav (34,9 %) eller gör det ibland (34,9 %).

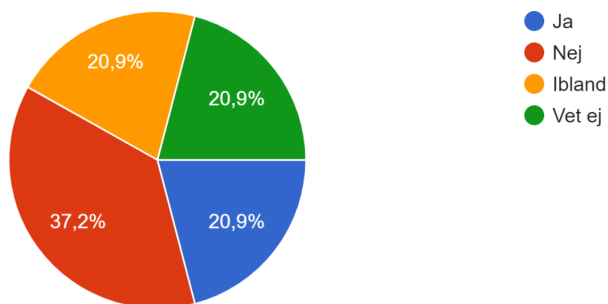
Fråga 21: Grupperar ni de icke-funktionella kraven efter vilken kategori de tillhör? Tex. alla krav som rör säkerhet grupperas för sig, användbarhetskrav för sig, osv.



Figur 18: Cirkeldiagram över procentuell fördelning över respondenternas kategorisering av icke-funktionella krav

Det framgår av figur 18 att 37,2 % av respondenterna grupperar sina IF krav efter vilken kategori de tillhör medan 30,2 % gör det inte. 18,6 % angav att de gör det ibland och 14 % angav att de inte visste om de grupperar sina IF krav eller inte. Detta visar att majoriteten av respondenterna kategoriserar IF krav efter tillhörande kategori medan en mindre del (30,2 %) inte gör kategorisering av IF krav. Dock var det 32,6 % som antingen inte visste vare sig de gjorde det, eller också gjorde det ibland. Denna svarsgrupp var den näst största.

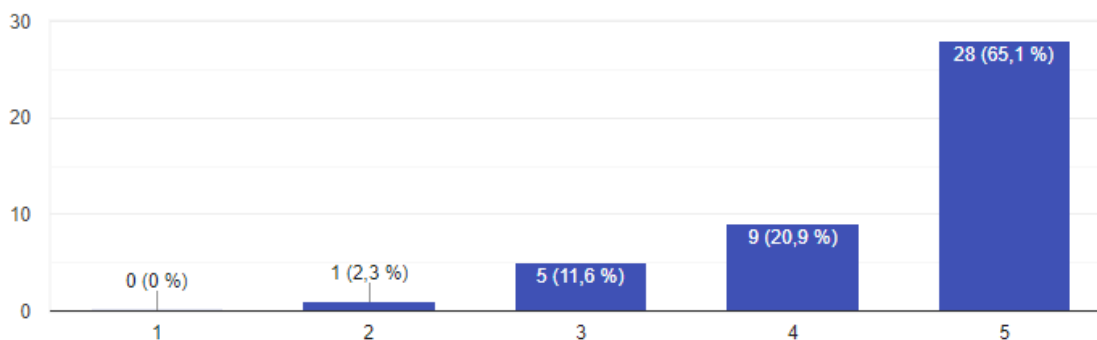
Fråga 22: Har ni något tillvägagångssätt för att säkerställa att alla har samma uppfattning/bild av de icke-funktionella kraven? Tex se till att alla implementerar samma nivå av användbarhet.



Figur 19: Cirkeldiagram över procentuell fördelning över respondenternas sätt att säkerställa uppfattning av icke-funktionella krav

Det framgår av figur 19 att 20,9 % har ett tillvägagångssätt för att säkerställa att de har samma uppfattning eller bild av de IF krav medan 37,2 % beskriver att de inte har det. 20,9 % framför att de har det ibland och 20,9 % vet inte om de har något tillvägagångssätt eller inte. Detta visar att majoriteten av respondenterna (37,2 %) inte har ett tillvägagångssätt för att säkerställa att alla har samma uppfattning om IF krav. Dessutom har lika stora andelar av respondenterna (20,9 %) angett att de antingen vet inte om de har ett tillvägagångssätt eller har det ibland eller har tillvägagångssättet för att alla har samma uppfattning om IF krav.

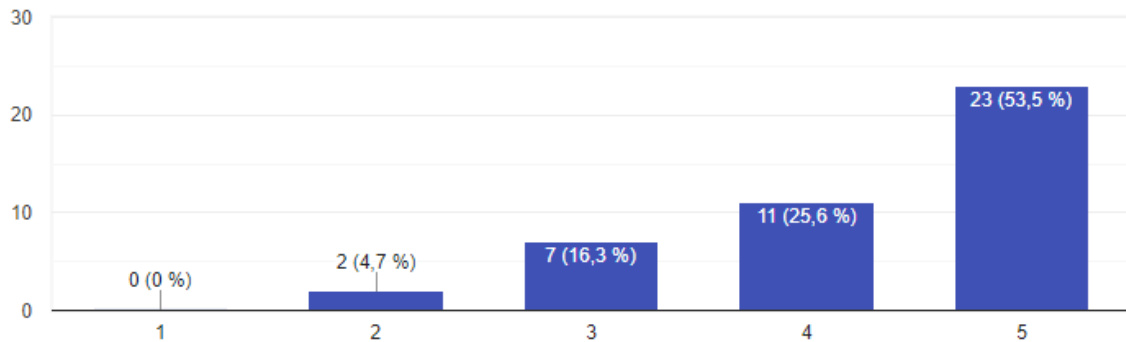
Fråga 23: Anser du att icke-funktionella krav är viktiga för mjukvaruutvecklingsprojekt?



Figur 20: Stapeldiagram över procentuell fördelning om respondenten anser att IF krav är viktiga för mjukvaruutvecklingsprojekt. Skalan visar 1 = Inte alls viktigt och 5 = Väldigt viktigt

Det framgår av figur 20 att 65,1 % av respondenterna ansåg att IF krav är väldigt viktiga för ett mjukvaruutvecklingsprojekt och har satt 5 på en 1-5 skala. 20,9 % har satt en 4 på skalan, 11,6 % har satt en 3 på skalan och 2,3 % har satt en 2 på skalan. Ingen av respondenterna har svarat "inte alls viktigt" det vill säga 1 på skalan. Detta visar att majoriteten av respondenterna ansåg att IF krav är viktiga för mjukvaruutvecklingsprojekt.

Fråga 24: Anser du att dokumentation av icke-funktionella krav är viktigt för mjukvaruutveckling?



Figur 21: Stapeldiagram över procentuell fördelning om respondenten anser att dokumentationen av IF krav är viktigt för mjukvaruutvecklingsprojekt. Skalan visar 1 = Inte alls viktigt och 5 = Världigt viktigt

Det framgår av figur 21 att 53,5 % anser att dokumentation av IF krav är viktig för ett mjukvaruutvecklingsprojekt och angav 5:a på en 1-5 skala. 25,6 % angav en 4:a på skalan, 16,3 % angav en 3:a på skalan och 4,7 % angav en 2 på skalan. Ingen av respondenterna svarade "inte alls viktigt", det vill säga 1 på skalan. Detta visar att majoriteten av respondenterna anser att dokumentationen av IF krav är viktigt för mjukvaruutveckling.

5 Analys och diskussion

I följande kapitel presenteras analys av resultatet från enkätundersökningen (se kapitel 4) tillsammans med de teoretiska utgångspunkterna (se kapitel 3) i fyra teman med underrubriker.

5.1 Utmaningar med att identifiera IF krav

Utifrån fråga 10 i enkätundersökningen identifierades tre intressanta svarsgrupper: 2,3 % av respondenterna valde att ange samtliga alternativ som IF krav, vilket var intresseväckande då forskning föreslår att IF krav ofta missuppfattas eller felaktigt kategoriseras som FK (Eckhardt, Vogelsang & Fernandez 2016). Svaret kan ha berott på att respondenterna i enkätundersökningen saknar gedigen kunskap eller kännedom om definitionen av IF krav. Detta ledde till att respondenterna chansade på att ange alla alternativ istället för att välja de rätta alternativen. En annan orsak till respondenternas svarsval kan vara att de vanligtvis inte separerar IF krav från FK vid dokumentationen och därmed uppfattade att alla krav tillhör en och samma kategori. Detta säger delvis någonting om okunskap och svårigheter att definiera IF krav, men även om bristfällig uppfattning av IF krav.

Enligt Eckhardt, Vogelsang och Fernandez (2016) är en av utmaningarna med IF krav att kunna särskilja dem från FK. Enligt enkätundersökningens fråga 10 var det endast 21 % (9 respondenter) som kunde urskilja IF krav vilket tyder på att det råder kunskapsbrist vad gäller definitionen av IF krav och distinktionen från FK. Detta i sin tur påverkar förmågan att dokumentera och prioritera de IF kraven eftersom det är en av de mest väsentliga förutsättningarna: utan tydlig distinktion och definition kan IF krav inte hanteras och dokumenteras.

Jarzewowicz och Weichbroth (2021) argumenterar att om det finns gedigen kunskap om IF krav i en mogen organisation så förbises inte de IF kraven och dokumentationen av dessa. För att avgöra huruvida respondenterna arbetar i mogna organisationer reviderades fråga 7: "Hur stort är företaget du arbetar på idag?", då det kan antas att större företag har en högre mognadsgrad eftersom ett större företag kräver mer strategiskt arbete och därav ökad mognad. Enligt enkätundersökningen arbetar 88,4 % respondenter på ett stort eller ett medelstort företag, vilket tyder på att de har rätt förutsättningar för att kunna implementera praxis av att dokumentera IF krav

Den tredje intressanta svarsgruppen motsvarade 16,2 % av respondenterna och hade endast angett två alternativ som IF krav (av fyra möjliga IF krav), vilka var korrekta. De kunde enbart identifiera två av fyra IF krav. Alla fyra IF krav var av olika typer: tillgänglighet/pålitlighet, underhållbarhet, säkerhet och kompatibilitet/överförbarhet. Att respondenterna endast kunde urskilja två och inte fyra typer av IF krav skulle kunna tyda på att det finns begränsad kunskap om IF krav som grundar sig i att endast vissa IF krav dokumenteras, vilket Behutiye et al. (2022) påvisar som bristfälliga faktorer. Karhapää et al. (2021) argumenterar att säkerhets- och användbarhetskrav ofta förbises då Scrum används som tillvägagångssätt för att dokumentera IF krav. Enligt fråga 13 av olika tillvägagångssättet för dokumentation av IF krav använder 18,5 % av respondenterna Scrum som tillvägagångssätt, 25,9 % angav att de använder backlogs; 25,9 % att de använder features och 3,7 % att de använder Confluence, och dessa alla ingår i Scrum. Nackdelarna med

product backlogs (PB) är att IF krav inte sär-definieras från FK, vilket Behutiye, Rodriguez och Ovio (2022) anser att de borde.

Eftersom det anses vara en utmaning att urskilja IF krav från FK (Eckhardt, Vogelsang & Fernandez 2016) skulle resultatet från enkätundersökningen kunna tyda på att respondenterna endast dokumenterade vissa typer av IF krav per praxis och inte alla, då alla ej kunde identifieras. Enligt fråga 10 var de två enklaste kategorierna att identifiera tillgänglighet/pålitlighet (83,3 %) och underhållbarhet (71,4 %), medan de två svåraste att identifiera var säkerhet (64,3 %) och kompatibilitet/överförbarhet (57,1 %). Det är därför möjligt att IF krav som rör kategori säkerhet och kompatibilitet/överförbarhet dokumenteras inte till lika stor grad som IF krav som rör kategori tillgänglighet/pålitlighet och underhållbarhet.

En ständig utmaning med att kvalitetssäkra IF krav är tidsbrist, vilket är svårt att undgå i projekt. Stress i samband med närliggande deadlines och tidsbrist kan leda till slarvfel i dokumentationen (Alsaqaf, Daneva & Wieringa 2019). Felaktig eller bristfällig dokumentation, som till exempel att misslyckas att identifiera samtliga IF krav, leder i sin tur till felaktig implementering av krav. Som en följd av detta måste systemet genomgå ytterligare iterationer för att kunna kvalitetssäkras (Behutiye et al. 2022).

5.1.1 (Tyst) kunskap om IF krav

Ett av de mest omtalade ämnen i litteraturstudien var den tysta kunskapen som är en katalysator i agila mjukvaruutvecklingsprojekt, men hämmande för dokumentationen av IF krav då den tysta kunskapen leder till eller är orsaken bakom mindre dokumentation (Alsaqaf, Daneva & Wieringa 2019; Behutiye et al. 2022; Karhapää et al. 2021; Jarzębowicz & Weichbroth 2021). Det är inte kunskap och erfarenhet av IF krav per se som är problematiskt för dokumentation. Tvärtom är det en förutsättning att ha både erfarenhet och mycket god förståelse för att kunna kvalitetssäkra IF krav, enligt den svenska standarden (SS-EN ISO 9001:2015). Problemet ligger i att det förekommer mer tillit till den tysta kunskapen som mer seniora utvecklare besitter som då kan ersätta dokumentation. Detta är främst dokumenterat i agila mjukvaruutvecklingsprojekt där en vanlig missuppfattning är att minimal dokumentation innebär desto mindre, desto bättre, vilket resulterar i nästintill obefintlig dokumentation (Behutiye et al. 2017).

Behutiye et al. (2022) beskriver att i agila mjukvaruutveckling ligger fokus på att snabbt leverera en fungerande mjukvara istället för att ha en omfattande dokumentation. Problematiken uppstår när kunskapen om IF krav inte går att förmedla, transportera och bearbeta eftersom det inte går att verifiera kvaliteten och därmed bortfaller kvalitetssäkringen. Den tysta kunskapen är per definition svårdefinierad och därför svår att förmedla. Juniora utvecklare befinner sig oftast i behov av riktlinjer och saknar denna tysta kunskap. Som en följd av detta lämnas juniora utvecklare till att skapa sina egna uppfattningar, vilket resulterar i att verifiering blir nästan omöjlig. En annan vanligt förekommande problematik med de IF krav är dess tvetydighet, vilket kan härledas från den tysta kunskapen.

Ramesh, Cao och Baskerville (2010) beskriver att de IF kraven är vagt definierade och är svåra att definiera. 51,2 % av respondenterna i enkätundersökning anser att de har en mycket god förståelse för definitionen av IF krav, 20,9 % anser sig ha god förståelse (fråga 8). Bland respondenterna angav 83,7 % att de har erfarenhet av att jobba med IF krav (fråga 9). Det

tyder på att många anser sig ha en etablerad förståelse för IF krav i de svenska mjukvaruutvecklingsprojekt utifrån deras erfarenhet med IF krav. Detta bör inte förväxlas med den gedigen kunskap och erfarenhet som den svenska standarden kräver. Det var nämligen endast 21 % som kunde identifiera och särskilja IF krav från FK enligt fråga 10. Detta anses vara problematiskt med tanke på att det finns risk för att felaktiga tillvägagångssätt etablerats. Majoriteten (79 %) av respondenterna i fråga 10 inte kunde identifiera IF krav – men 83,7 % har erfarenhet av att arbeta med dem enligt fråga 9. Detta kan vara ett resultat av att det förekommer mer tillit till den tysta kunskapen över dokumentation i den agila mjukvaruutvecklingen, då det lämnar utrymme för egna tolkningar och uppfattningar som inte går att verifieras.

Lika viktigt att påpeka att majoriteten av respondenterna, vilka utgör 83,7 %, har erfarenhet av att arbeta med IF krav (fråga 9) och dessutom har god kännedom om IF krav, som utgör 72,1 % av respondenterna i enkätundersökningen (fråga 8). Detta leder till ett antagande om att det framträder en god kännedom om IF krav inom den svenska mjukvaruutvecklingsbranschen och att det förekommer att personer som arbetar med IF krav besitter en god erfarenhet. Vad som stärker ovanstående argument är att 72,1 % av respondenterna dokumenterade krav (fråga 11), var det enbart 48,8 % som definierade IF krav från FK (fråga 12), det vill säga 35 % av samtliga respondenter. Mindre än hälften för kunskapen om IF krav vidare genom dokumentation och resterande 65 % förlitar sig på den tysta kunskapen i svenska mjukvaruutvecklingsprojekt.

Karhapää et al. (2021) menar att insamlingen av IF krav oftast görs verbalt men beroende på situation och kunskap kan personen som samlar in kraven, bland annat produktägaren, skriva ner dessa för eget bruk. Det är alltid riskabelt att förlita sig på en persons subjektiva uppfattning då det är nästan omöjligt att vara opartisk. Alsaqaf, Daneva och Wieringa (2019) argumenterar att det finns risk att personen som prioriterar de IF kraven lägger sin personliga åsikt i vad som ska prioriteras och bortprioriterar de IF krav. Om dokumentation saknas och det förekommer mer tillit till den tysta kunskapen, kan ingen verifiera eller kontrollera beslut som tas vid prioriteringen.

Fråga 22 i enkätundersökningen syftar till att illustrera om svenska mjukvaruutvecklingsprojekt har något tillvägagångssätt för att säkerställa samma uppfattning om de IF kraven. Respondenternas svar visar att endast 20,9 % har det medan 37,2 % svarar att de inte har det, 20,9 % svarar ibland och 20,9 % svarar vet inte. Detta påvisar att många verksamheter saknar något sätt att säkerställa samma uppfattning av de IF kraven, vilket enligt Behutiye et al. (2022) är en viktig faktor för att kunna specificera och kvalitetssäkra dokumentationen av IF krav. Finns inte rätt uppfattning eller förståelse kan medarbetare enligt Alsaqaf, Daneva och Wieringa (2019) göra egna antaganden och den tysta kunskapen framhävs viktigare än att dokumentera.

Idag dokumenterar 72,1 % av respondenterna krav enligt fråga 11, vilket belyser att det finns en benägenhet att dokumentera krav i svenska mjukvaruutvecklingsprojekt. Enligt Medeiros et al. (2018) kan benägenheten att dokumentera bidra till att kunskapsbortfall minimeras. Behutiye et al. (2020b) framför att resultatet av att inte dokumentera kan leda till försämrad kvalitet, ökade underhållskostnader och omarbetningar, vilket kan ses som teknisk skuld. Följaktligen blir det påtagligt att det finns behov av standardiseringar för dokumentation av IF krav. För att lyckas med dokumentationen behövs gedigen kunskap, som dessutom kan minska risken för teknisk skuld och blir därför ekonomiskt försvarbar.

Fråga 14 i enkätundersökningen visar att 28,6 % av respondenterna följer en standard när de dokumenterar de IF kraven medan 71,4 % säger att de varierar från projekt till projekt. Medeiros et al. (2018) anser att en gemensam dokumentation kan underlätta för involverade i projektet att ta det av ny och uppdaterad information från en källa, vilket även ökar medvetenheten av helhetsperspektivet och en mer komplett förståelse och mindre redundans av de IF kraven. Behutiye et al. (2020b) argumenterar att genom att ha ett väldefinierat tillvägagångssätt kan kvalitet säkerställas medan Behutiye, Rodríguez och Oivo (2022) anser att riktlinjer en form av instruktionsmanual kan användas för att hjälpa till med dokumentationen av IF krav. Enligt Behutiye et al. (2022) kan de mer oerfarna utvecklarna främjas genom att förse dem med kunskap och tillvägagångssätt för hur de IF kraven ska dokumenteras och implementeras. De juniora utvecklarna saknar generellt riktlinjer för hur de IF kraven ska dokumenteras då de seniora utvecklarna litar på sin tysta kunskap, som per definition, är svår att förmedla vidare (Alsaqaf, Daneva & Wieringa 2019). Endast hälften (51,3 %) av respondenterna i fråga 18 använder sig av skriftlig dokumentation när de ska implementera de IF kraven och endast en av respondenterna använder sig av en checklista som är kopplat till varje uppdrag hen medverkade i. Dessutom har fyra av respondenterna svarat att de kombinerar olika tillvägagångssätt när de dokumenterar.

Behutiye, Rodríguez och Oivo (2022) argumenterar att det behövs flera etablerade riktlinjer för att dokumentera IF krav på grund dokumentationsbristen som råder. Orsaken till att de IF kraven inte dokumenteras visar på att de IF kraven anses vara underförstådda vilket 10,8 % av respondenterna medger att de har informationen om de IF kraven i huvudet enligt fråga 18. Att ha informationen i huvudet, det vill säga den tysta kunskapen kan medföra att viktig kunskap och tillvägagångssätt för hur de IF kraven ska implementeras inte förs vidare (Alsaqaf, Daneva & Wieringa 2019; Karhapää et al. 2021). Å andra sidan argumenterar Karhapää et al. (2021) att utvecklarna behöver förstå "varför" ett krav ska implementeras och varför det skapar värde för kunden.

Utifrån fråga 15 besvarade endast 16,3 % av respondenterna att det var väldigt tydligt vad som ska implementeras från de IF kraven och 25,6 % angav att det var tydligt, dessa två svarsgrupper (sammanlagt 41,9 %) var lika stor som den svarsgrupp som angav en neutral eller delad åsikt (41,9 %). Av alla svarsgrupper var dessa tre de mest förekommande (de som angav 1:a, 2:a eller 3:a). Dock kvarstår faktumet att mindre än hälften (41,9 %) av samtliga ansåg det vara antingen tydligt eller väldigt tydligt. Det råder därmed en majoritet av otydlighet (58,2 %), inklusive svarsalternativen 3, 4 och 5 där antingen osäkerhet eller klar otydlighet råder. Detta kan tyda på att dokumentationen är bristfällig eller inte existerar och det genererar svårigheter att förstå "varför" ett krav ska implementeras. Att inte ha tydlighet eller klarhet om de IF kraven kan leda till feltolkningar (Behutiye, Rodríguez & Oivo 2022).

5.2 Dokumentationens betydelse för IF krav

Det förekommer flera olika tillvägagångssätt för att dokumentera IF krav. Fråga 13 i enkätundersökningen framför att medelvärdet på antal olika tillvägagångssätt som används är tre. Respondenternas svar visar att de använder allt ifrån ett till sju olika sätt att dokumentera IF krav vilket kan kopplas till det Behutiye et al. (2020b) beskriver som en flexibel ansats där dokumentationen är beroende av kontext och mål. Dock påpekar Medeiros et al. (2018) att vid användning av flera olika tillvägagångssätt kan det ha en negativ inverkan på tilliten eftersom utvecklarna har svårt att veta om exempelvis kravspecifikationen är uppdaterad eller inte. 55,5 % av respondenterna i enkätundersökningen använder mellan tre till sju olika tillvägagångssätt och 14,8 % använder sex till sju olika tillvägagångssätt för att dokumentera

de IF kraven. Desto fler olika dokument och tillvägagångssätt som används, desto svårare blir det att hålla ordning på vilken information som är korrekt eller uppdaterad. Dessutom är det viktigt att notera att det finns risker med att information försvinner mellan dokument då det oftast glöms av. Detta kan vidare kopplas till bristen på ansvarstagande för IF krav (se avsnitt 5.3.1).

Fråga 13 visar att 37,2 % inte använder någon specifik tillvägagångssätt idag för att dokumentera de IF kraven vilket kan kopplas till det Medeiros et al. (2018) hänvisar till att det finns begränsade tillvägagångssätt för att kunna beskriva de IF kraven. De som inte dokumenterar de IF kraven saknar eventuellt ett tillvägagångssätt eller besitter mindre kunskap i hur de IF kraven ska dokumenteras. Det kan även handla om kommunikationsklyftor i ett team som Behutiye et al. (2022) påpekar. Dessa klyftor kan handla om att det förekommer diskussioner i mindre utsträckning om de IF kraven, olika tolkningar och förståelse av de IF kraven. Genom att ha öppna diskussioner om IF krav i mjukvaruutvecklingsteamet kan det minimera missförstånd och förbättra dokumentationen för att få en bättre gemensam syn på vad som ska dokumenteras och slutligen implementeras (ibid).

5.2.1 Tillvägagångssätt för att dokumentera IF krav

Fråga 17 belyser hur ofta revideringar av kraven görs tillsammans med kunden. De FK revideras "ofta" av respondenterna medan de IF kraven revideras mer "sällan" och i vissa fall "ofta" också (fråga 17). Att revidera krav "ofta" sätter stort fokus på att spårbarheten blir korrekt, det vill säga att varje krav behöver manuellt uppdateras i varje dokument och artefakt som används. Detta ses som problematiskt enligt Behutiye et al. (2017) är när textbaserade programvara används såsom Word, Excel och Wiki-page eftersom dessa oftast är friliggande från andra mjukvaror. 25,9 % av respondenterna svarade på fråga 13 att de använde Excel, vilket är det fjärde största tillvägagångssättet som används för att dokumentera IF krav, medan 14,8 % använder Word enligt samma fråga.

Jarzebowicz och Weichbroth (2021) visar på att det finns brist på verktyg för att stödja samarbete med kunden medan Medeiros et al. (2018) visar på att en effektiv delning av dokumentation till kunden är viktig för att kunna säkerställa komplett förståelse och rätta uppdaterade krav. Vid revideringar som sker ofta kan utvecklare och andra medlemmar i ett team enkelt glömma bort att ändra olika dokument och det är lätt hänt att krav missas att uppdateras om flertalet olika tillvägagångssätt och dokument används. Redundans av likartade krav inom dokumentation kan öka vilket leder till ökad komplexitet för utvecklarna enligt Medeiros et al. (2018).

Det som blir viktigt enligt Behutiye, Rodríguez och Oivo (2022) är klarheten på IF krav. Finns inte rätt definition och indelning av krav i kategorier kan detta resultera med feltolkningar i dokumentationen. Fråga 12 belyser om respondenterna definierar vilka krav som är FK och vilka som är IF krav vid dokumentationen. Här råder det oenighet mellan respondenterna. 48,8 % säger att de definierar krav medan 46,5 % säger att de inte gör det. Att inte definiera de olika kraven kan leda till att dokumentationen blir otydlig, enligt Glinz (2007). Att dessutom beskriva kraven i olika artefakter blir det svårt att hålla ordning på allt vid eventuella revideringar.

Kunden är en viktig del av dokumentationen och bör vara närvarande, tillgänglig, samarbetsvillig och kunnig för att kunna framföra de "rätta" IF kraven. Eftersom den agila

mjukvaruutvecklingen förespråkar kontinuerligt samarbete med kunden (Ramesh, Cao, & Baskerville 2010) är det inte konstigt att respondenterna som deltagit i enkätundersökningen “ofta” reviderar FK. Det som är intressant är att respondenterna “sällan” reviderar de IF kraven. Detta visar på att det kan finnas ett samband med att kunden mer försummar de IF kraven, då mer fokus läggs på FK. Dessutom påstår Ramesh, Cao, Baskerville (2010) att kunden saknar insikt och har bristande kunskaper om de IF kraven då dessa är mer “osynliga” vilket också Behutiye et al. (2020b) anser.

Orsaken till varför de FK revideras av respondenterna i enkätundersökningen i större omfattning och de IF kraven mer sällan kan beror på vad Alsaqaf, Daneva och Wieringa (2019) argumenterar för, att de IF kraven inte prioriteras i högre grad och till och med bortprioriteras speciellt vid tidsbrist. Kommunikation spelar en väsentlig del av revideringar och insamling av IF krav. För att minska revideringar av krav behöver kunden specificera IF krav och dokumentera dem i början av ett projekt. Om kunden eller den ansvarige för dokumentationen från början hade specificerat sina IF krav på ett mer konkret sätt hade de IF kraven inte behövts revideras. Men detta är det som blir problematiskt. Kunden är inte alltid tillgänglig när dokumentation av IF sker och varje tillvägagångssätt vare sig det handlar om kravspecifikationen, US eller UC kräver kundens syn, input, förståelse och närvaro.

Ramesh, Cao och Baskerville (2010) beskriver att målet med kravspecifikationen är att få fram kundens behov och krav på det som ska utvecklas. Dock anses kravspecifikationen vara en olämplig form inom den agila mjukvaruutvecklingen, då det är svårt att få fullständiga krav nedskrivna, då revideringar sker. Eftersom FK revideras “ofta” och IF krav både “ofta” (30,2 %) och “sällan” (51,2 %) enligt respondenternas svar på fråga 17, kan denna form vara problematiskt eftersom den oftast är skriven i ett naturligt textbaserat språk. Oftast blir kravspecifikationen lång, enligt Medeiros et al. (2018) när företagen är kunddrivna. Dock kan en lång kravspecifikation resultera i redundans och otydligheter till de som ska implementera kraven. Detta kan kopplas till fråga 16 som tar upp hur svårt eller lätt det är att dokumentera vad som ska implementeras från de IF kraven. 44,2 % av respondenterna svarade att det var “väldigt lätt” eller “lätt” medan 37,2 % hade en mer neutral åsikt om hur lätt/svårt det var medan 18,6 % ansåg att det var “svårt” eller “mycket svårt”.

Medeiros et al. (2018) beskriver att kvaliteten på kravspecifikationen kan förbättras genom att mer tydliggöra och specificera de IF kraven så att utvecklarna har något mer specifikt att gå på vid implementering. Eftersom kravspecifikationen enligt Ramesh, Cao och Baskerville (2010) inte alltid är komplett vilket kan resultera i att utvecklarna behöver göra egna tolkningar av kraven i kravspecifikationen på grund av tidsbrist. Detta kan leda till mer omarbetningar av det som implementerades om kraven behöver revideras ofta.

Enligt fråga 13 var UC det nästa vanligaste tillvägagångssättet (44,4 % av respondenterna använde det) för att dokumentera IF krav. Dock framgick det av de artiklarna som granskades i litteraturstudien att endast användning av UC inte var ett av det vanligaste sättet att dokumentera utan att personer som dokumenterade använde sig mer av US vilket var det tredje vanligaste sättet som visade att 37 % av respondenterna använder. Alsaqaf, Daneva och Wieringa (2019) beskriver att US är det mest förekommande tillvägagångssättet att dokumentera. Att använda både US och UC är enkla sätt att beskriva IF krav textuellt för att underlätta förståelse av dessa för kunden (Jarzębowicz & Weichbroth 2021; Medeiros et al. 2018). Dock anser Medeiros et al. (2018) att UC inte är ett optimalt sätt att specificera de IF kraven då det saknar teknisk anknytning och är mer anpassade för dokumentation av FK. En likartad uppfattning har Karhapää et al. (2021) som framför att US kan vara problematiskt att

använda för att specificera de IF kraven men att dessa krav bör ses som ett tillägg till de FK. Dessutom anser Medeiros et al. (2018) att US saknar teknisk jargong.

Att kunna specificera de IF kraven på ett strukturerat sätt så att både kunden och utvecklaren förstår dess innebörd är en stor utmaning (Behutiye et al. 2020b). Följaktligen är det viktigt att vid val av tillvägagångssätt för att dokumentera IF krav ha mottagaren i åtanke för att kunna avgöra hur väl dokumentationen skall genomföras. För att underlätta för utvecklaren att förstå IF krav behöver kraven struktureras och anpassas för att sedan kunna implementera dessa. Mottagaren av dokumentationen behöver ha förståelse för IF krav, därför är det viktigt att strukturera och förtydliga dessa i dokumentationen.

Enligt fråga 21 i enkätundersökningen, utgör 37,2 % av respondenterna de som grupperar IF krav efter deras tillhörande kategori. Grupperingar av IF krav görs inom svensk mjukvaruutveckling. Däremot visade enkätundersökningen att 30,2 % av respondenterna inte gör grupperingen av IF krav. Resterande 18,6 % gör det ibland och 14 % vet inte varken de gör det eller ej. Detta visar på att det görs delvis eller sporadiskt och påvisar att det saknas standardiseringar vid dokumentation av IF krav, vilket bekräftas av fråga 14 i enkätundersökningen där 71,4 % av respondenterna anger att de inte har någon standard för att dokumentera IF krav. Grupperingen av IF krav kommer att underlätta för utvecklarna att kunna implementera dessa krav för att göra kunden tillfredsställd med mjukvaran. Detta påvisade Medeiros et al. (2018) som anser att kvaliteten på kravspecifikation kan förbättras genom att tydliggöra och specificera de IF krav som i sin tur kommer att underlätta implementeringsfasen för utvecklare. Karhapää et al. (2021) argumenterar att bristande kunskaper kan finnas hos utvecklarna, specifikt hos juniora utvecklare.

5.2.2 Kombination av dokument

Det är viktigt att utvecklarna har en översiktlig förståelse för helheten av kraven, såväl IF krav som FK, då det gör att utvecklarna kan se vilka krav de ska fokusera på och vilka krav som hänger ihop (Alsaqaf, Daneva & Wieringa 2019). Enligt Ramesh, Cao och Baskerville (2010) kan kravspecifikationen ge en översiktlig förståelse av samtliga krav. Enligt fråga 13 i enkätundersökningen var det 74,1 % av respondenterna som använde kravspecifikation vid dokumentation av krav — en klar majoritet.

PB erbjuder också ett översiktligt perspektiv, men det ingår i Scrum-ramverket. Scrum-ramverket ökar sannolikheten att icke-funktionella säkerhets- och användbarhetskrav missas att dokumenteras enligt Karhapää et al. (2021). Vidare delas inte alltid kraven upp per FK och IF krav i PB och kategorier definieras inte, vilket gör det svårare att kontrollera vare sig just säkerhets- och användbarhetskrav faktiskt har dokumenterats. Kravspecifikation är alltså det bättre alternativet för att skapa just en översiktlig bild av kraven, då man delar upp kraven per FK och IF krav.

Samtidigt som utvecklarna måste ha en översiktlig förståelse för kraven behöver de även förstå dem på ett mer djupgående sätt. Dokumentationen måste erbjuda en förståelse för motivationen bakom de IF kraven, *“the why”* bakom varje krav med andra ord, som Karhapää et al. (2021) argumenterar är viktigt. Detta kan underlätta och tydliggöra hur de bör bli implementerade, eftersom det finns många olika sätt att göra det på. Detta är ett sätt att minska tvetydigheten från dokumentation till implementering och förbättrar därmed mjukvarans kvalitet. Eftersom kravspecifikationen erbjuder ett översiktligt perspektiv kan det

vara en bra idé att komplementera det med ett annat tillvägagångssätt som i sin tur beskriver kraven mer djupgående.

US är ett tillvägagångssätt för att dokumentera IF krav som specificerar informationen på djupet genom att definiera vem, vad, för vad och/eller varför kravet ska implementeras (Karhapää et al. 2021) och är därför ideal för behovet att förstå motivationen bakom kraven. Dessa ska utformas i nära samarbete med kunden enligt Behutiye, Rodriguez och Oivo (2022) då de ska specificeras utifrån användarens perspektiv och mål vilket betyder att utvecklarna måste vara involverade i framtagningen av dokumentation av krav tillsammans med kunderna.

Av respondenterna var det 37 % enligt fråga 13 som angav att de använder US som tillvägagångssätt idag, vilket var det tredje mest använda tillvägagångssättet efter UC (44,4 %) och kravspecifikation (74,1 %). UC anses dock vara mer lämpat för FK, även om det fungerar att använda för att dokumentera IF krav enligt Larman (2005). Detta skulle innebära att IF krav anges som begränsningar på FK, vilket Alsaqaf, Daneva och Wieringa (2019) har påvisat kan ha en negativ inverkan på implementationen av IF krav. Det anses inte vara fördelaktigt att använda UC för att dokumentera IF krav som många idag gör, utan istället använda sig av det näst-mest använda tillvägagångssättet US. Dock förbises ofta viktiga IF krav som säkerhet och användbarhet i US, därför bör US kompletteras med en annan dokumentationsform som förtydligar kategorier av IF krav. På så sätt kan avsaknad av vissa kategorier lättare tydliggöras.

Enligt fråga 10 var just säkerhet en av de kategorier som var svårast att identifiera, vilket kan betyda att de typer av IF krav också löper risk att förbises om endast US används som tillvägagångssätt. Ett tillvägagångssätt som lämnar plats åt att specificera kategorier av IF krav är kravspecifikationen, även om den inte "tvingar" att ange kategori, såsom ett system kan göra. För att säkerställa att kombinationen av kravspecifikation och US fungerar hade det krävts att det införs som standard, där kategorisering av IF krav alltid görs i kravspecifikationen. För att minska risken att informationen i kravspecifikationen inte uppdateras mellan kan det dessutom vara fördelaktigt att integrera de två dokumenten i ett och samma system, så att uppdateringar sker automatiskt.

Det är värt att notera att kravspecifikation är ett dokument ur vattenfallsmodellen medan US är från den agila metoden. Ingen av respondenterna svarade att de använde just denna kombination, men enligt en analys av respektive dokumentationsmall är det just dessa två som tillsammans kompenserar varandra för att skapa så komplett dokumentationsform som möjligt. Vidare forskning bör göras i att kombinera vattenfalls- och agila metoder.

5.3 Faktorer som påverkar kvalitetssäkring av IF krav

Kvalitet är något som tar tid (Karhapää et al. 2021) och som resulterar i ett ständigt hållbart förbättringsarbete gällande att ökad kundnöjdhet, ökad förståelse för intressenternas behov och förväntningar samt effektivare planering. För att nå kvalitetsmålen behöver det finnas tillvägagångssätt för vad som ska göras, vem som är ansvarig, vilka resurser krävs, när det ska vara klart och hur resultatet ska kunna utvärderas (SS-EN ISO 9001:2015). Eftersom kvalitet ska byggas in från början (Eriksson 2008) bör det finnas tydlig ansvarsfördelning för hur de IF kraven ska hanteras och dokumenteras (Behutiye et al. 2022), redan från början.

5.3.1 Ansvar som faller mellan stolarna

Fråga 19 nyanserar om *någon* har ett översiktligt ansvar för att dokumentera de IF kraven, vilket 27,9 % respondenterna angett ja på; 25,6 % angett att någon ibland har det; 30,2 % angett att ingen har det och 16,3 % vet inte om någon har det. Det är en relativt jämn svarsfördelning som påvisar att det inte finns någon standard och eftersom majoriteten svarade antingen "nej", "ibland" eller "vet inte". Det tyder på att det råder en viss tvetydighet huruvida *om* någon har ansvar och dessutom *vem* som har ansvar över de IF kraven och dokumentationen av dessa. Både Karhapää et al. (2021) och Behutiye et al. (2020b) beskriver liknande situationer från deras kvalitativa undersökningar och drog slutsatsen att olika personer ansvarar för dokumentationen av IF krav, även om produktägaren ofta har det översiktliga ansvaret att prioritera dem, så anger flera medarbetare sina åsikter som sedan används som underlag för beslut om hur de IF ska prioriteras. Detta belyser problemet att det varken finns standardiseringar, riktlinjer eller experter för området. Det anses också vara problematiskt för kvalitetssäkringen då Behutiye et al. (2022) påvisar att det krävs tydliga ansvarsroller över vem som dokumenterar de IF kraven för att undvika brister i dokumentationen. Vidare är detta en indikation på att majoriteten av svenska mjukvaruutvecklingsföretag inte uppnår den svenska standarden för kvalitetssäkring helt då det enligt den svenska standarden (SS-EN ISO 9001:2015) bör finnas någon som har ansvar över hur, när och vad som ska ske, samt för när ett projekt ska vara klart och hur resultatet ska se ut.

Karhapää et al. (2021) argumenterar att prioritering av de IF kraven är en utmanande process och konsekvensen blir att utvecklingarna fokuserar på fel krav. Det krävs gedigen kunskap om IF krav för att kunna prioritera dem, med betoning på kvalitet över kvantitet i ansvarsrollen.

Enligt fråga 20 indikerar enkätundersökningen att majoriteten av respondenterna (65,1 %) inte har något tydligt sätt att fördela det översiktliga ansvaret för de IF kraven, det vill säga de som svarade nej, ibland, eller vet ej. Ett antagande kan då göras att expertkompetens saknas av majoriteten av mjukvaruutvecklingsteamerna, eftersom denne annars hade erhållit det översiktliga ansvaret. Detta leder oss i sin tur till att ifrågasätta huruvida de då kan säkerställa att dokumentationen uppfyller en viss nivå som anses godtycklig enligt svensk standard. Enligt den svenska standarden (SS-EN ISO 9001:2015) krävs det att dokumentationen ska innehålla tydlig information samt att den uppdateras kontinuerligt vid revideringar, vilket inte kan säkerställas utan att en person som är ansvarig för det ser till att detta sker enligt Alsaqaf, Daneva och Wieringa (2019). Då de påvisar klyftor och tvetydighet när det gäller ansvarstagande för IF krav som resulterar i att de IF kraven bortprioriteras helt och medarbetare fokuserar endast på just det som de är ansvariga för. Saknas en ansvarig person, riskerar IF kraven att bortprioriteras.

5.3.2 Både kunskap och entusiasm är nödvändiga

Enligt denna studies litteraturstudie är kunskap om IF krav en förutsättning för att kunna ansvara för dem; att kunna identifiera dem och därmed konstatera och dokumentera dem. Behutiye et al. (2022) beskriver att det bör finnas ett intresse, gedigen kunskap och vilja/entusiasm för de IF kraven för att kunna avgöra om en produkts kvalitetsnivå är uppfylld och säkerställa produktens beredskap. Behutiye, Rodríguez och Oivo (2022) framför att det är viktigt att ha *förståelse* för hur viktiga de IF kraven är och att det är ledningens ansvar att se till att öka medvetenheten om vikten av de IF kraven och att de behöver dokumenteras. Enligt

fråga 23, anser 86 % av respondenterna att det är “viktigt” eller “mycket viktigt” med IF krav i ett mjukvaruutvecklingsprojekt, parallellt indikerar fråga 24 att samma andel (86 %) anser att *dokumentation* av IF krav är “viktig” eller “mycket viktig”. Detta tyder på att de flesta anställda inom svenska mjukvaruutvecklingsprojekt förstår att IF krav är viktiga och dokumentation av dem är viktiga enligt fråga 24 och 23.

Paradoxalt nog och kanske den viktigaste insikten med denna studie är att endast 21 % av respondenterna hade full förståelse för vad ett IF krav är (fråga 10), medan 91 % hade mellan 1-4 rätt av 4 möjliga på fråga 10, som gav respondenten alternativ på FK och IF krav. Detta kan betyda att cirka 65,1 % av respondenterna enligt fråga 23 anser att IF krav är viktiga men saknar kunskap om dem. Om dessa personer då är med och prioriterar eller fattar beslut gällande IF krav, riskerar de att prioritera och implementera fel krav (Karhapää et al. 2021). Även om det finns en viss kunskap om IF krav behöver den vidareutvecklas för att ett samarbete mellan medarbetare ska fungera, vilket resultaten av enkätundersökningen visar att de flesta gör.

Den svenska standarden för kvalitetssäkring rekommenderar att vidareutveckla befintlig kompetens eller anlita mer kompetens (SS-EN ISO 9001:2015) då en organisation måste kunna säkerställa att deras medarbetare är kompetenta och har lämplig utbildning relaterade till deras arbetsuppgifter. Fråga 3 behandlar respondenternas utbildningsområde där 92 % har en utbildning inom ett tekniskt område och fråga 4 framför att 76,8 % av respondenterna har en eftergymnasial utbildning. Detta tyder på att majoriteten av respondenterna har adekvat utbildning inom rätt område och att kompetent personal finns inom den svenska mjukvaruutvecklingsbranschen i idag.

Enligt respondenternas svar på fråga 11 i enkätundersökningen var det bara en respondent som framförde att han inte dokumenterade krav i företaget han arbetade på idag. Denna respondent har ingen adekvat utbildning utan är självlärd eller har enskilda kurser, finner IF krav otydliga och anser varken eller att de är viktiga eller att dokumentera dem men har mer än elva års erfarenhet som applikationskonsult på ett stort företag. Okunskap om IF krav leder till mindre dokumentation. Detta kan kopplas till den tysta kunskapen som denna person besitter. Behutiye et al. (2017) förklarar att mindre dokumentation av IF krav förekommer när den tysta kunskapen prioriteras.

5.3.3 Kunden påverkar kvaliteten

Att ha dokumentation med dålig specificering av krav kan leda till omarbete, vilket i sin tur leder till att mer tid går åt av ett projekt på att revidera som slutligen kan påverka kundnöjdheten. Enligt svensk standard (SS-EN ISO 9001:2015) är ökad kundnöjdhet och förståelse något som ska eftersträvas i hela organisationen. Detta kan kopplas till det Behutiye et al. (2020b) beskriver att kunden oftast saknar insikt och tillräcklig kunskap om IF krav och att det enligt Karhapää et al. (2021) viktigt att skapa medvetenhet om de IF kraven hos kunden.

Ameller et al. (2012) anser att det är av yttersta vikt att hjälpa kunden och vara en stödjande part när kunden är de som tar fram de IF kraven. För att kunna stödja kunden i detta behövs erfarenhet och kompetens av arbetsroller såsom en mjukvaruarkitekt som har förmåga att se helhet gällande både design och arkitekturella aspekter av ett mjukvarusystem. Av fråga 5 i enkätundersökningen framgår det att endast 7 % arbetar som arkitekter idag. Detta visar att

det finns ett behov av kompetensutveckling och att flera utbildade mjukvaruarkitekter behövs inom svensk mjukvaruutvecklingsbranschen.

En annan arkitektroll som kan främja dokumentation och kvalitetssäkring av IF krav är lösningsarkitekt. Det kan ske att kunden "över-specificerar" de IF krav till den grad att det behövdes nå konsensus mellan kunden och leverantören i sådana frågor då alla krav inte kan vara tekniskt eller ekonomiskt genomförbara för mjukvaruutvecklingsteamet (Ramesh, Cao & Baskerville 2010). Besluten som fattas av kunden måste alltså vara tekniskt, ekonomiskt och strategiskt informerade, vilket kräver assistans. Även om det är främjande att utvecklarna själva har direktkontakt med kunden (Ramesh, Cao & Baskerville 2010; Karhapää et al. 2021) saknar de förmåga att assistera verksamhetsstrategiska beslut som exempelvis en lösningsarkitekt kan göra, som dessutom kan översätta tekniska termer från utvecklarna till kunderna och kunna tolka kundens behov till tekniska specifikationer.

5.4 Ansvarsroller för att kvalitetssäkra IF krav

Behutiye et al. (2022) anser att de befintliga riktlinjer och modeller som finns för dokumentation av krav inte är anpassade för olika faktorer, exempelvis tidsbrist, medvetenhet om IF krav och kommunikationsluckor. Detta leder till att riktlinjer för att dokumentera krav borde vara mer anpassade till olika faktorer som Behutiye et al. (2022) angett. Problematiken med verifikation av den tysta kunskapen återstår i den svenska mjukvaruutvecklingsbranschen. Detta omöjliggör kvalitetssäkring.

Enligt Behutiye et al. (2022) bör produktägare och projektledare intressera sig för IF krav och ha bra kunskap om dessa, då deras arbetsuppgift innefattar att avgöra om en produkts kvalitetsnivå är uppfylld. Enligt enkätundersökningen anger personer i denna yrkesgrupp att de har bra kunskap om dem. Samtliga respondenter som var projektledare och/eller produkt/serviceägare angav att definitionen av IF krav var "väldigt tydligt" (1/5) för dem, med undantag för en produktägare som angav att det var "tydligt" (2/5). Dock var det endast två utav dem som verkligen kunde definiera dem, enligt fråga 10 i enkätundersökningen.

Behutiye et al. (2022) beskriver att det råder brist på expertis. Det är en utmaning för att kunna dokumentera och hantera IF krav, eftersom projektledare och produktägare inte kan definiera dem tillräckligt väl, inklusive relevanta IF krav såsom säkerhet och användbarhet. Enligt enkätundersökningen kunde 64,3 % av respondenterna identifiera säkerhetskravet. När det gäller att kunna definiera IF krav var det 9 respondenter som klarade av det, varav endast två respondenter var projektledare (respondent 19 och 31). Det betyder att endast 22% av projektledarna och produkt/serviceägarna kunde identifiera IF krav. Bristen på expertis inom de roller som enligt Behutiye et al. (2022), ska säkerställa att kvalitetsnivån är uppfylld betyder att kvaliteten inte kan säkras.

På grund av att det saknas tillräcklig expertkompetens menar Behutiye et al. (2022) att det bör finnas guider med riktlinjer att utgå ifrån som beskriver hur IF krav kan hanteras och dokumenteras, med fokus på kvalitetssäkring. Dock visar enkätundersökningen (fråga 14) att det till stor del saknas standarder för hur just IF krav ska hanteras och dokumenteras, då endast 28,6% angav att de hade ett standardiserat sätt att arbeta med IF krav. 71,4 % av respondenterna angav att tillvägagångssätten för hantering av IF krav varierade från projekt till projekt och 18,6 % av respondenterna besvarade inte frågan.

5.4.1 Senioritet

Baserat på år av erfarenhet i rollen tillsammans med den förväntade ackumulerade kunskapen inom rollen, utbildningsnivån, och hur relevant utbildningsområdet var för rollen, kan slutsatser dras som visar att 44% kan vara seniora, 16% mid-level, och 40% juniora inom sitt yrke (se avsnitt 2.4.2). De varierande nivåerna av senioritet påverkar hur de hanterar dokumentation. Behutiye et al. (2020b) menar att deras erfarenhet påverkar hur de ser på behovet av dokumentation; ju *mer* erfarenhet en utvecklare har, desto *mindre* tenderar de att behöva eller vilja dokumentera sina processer. Som ett resultat är det vanligt att seniora medarbetare besitter en omfattande tyst kunskap och dokumenterar inte de IF kraven utan istället har kunskapen om hur de ska hantera "tyst" i huvudet. Det betyder att juniora medarbetare saknar tillräcklig dokumentation eller anpassade riktlinjer att följa, vilket leder till att de skapar egna tolkningar. Dock visar enkätundersökningen att de seniora respondenterna dokumenterar mest av alla. Det är tvärtom de juniora respondenterna som förlitar sig på att ha informationen i huvudet istället för att diskutera eller dokumentera den, vilket kan bero på att de saknar riktlinjer för hur de ska dokumentera de IF kraven.

Behutiye et al. (2022) beskriver att dokumentation av IF krav som utförs av chefsroller på högre nivåer kan påverka hur utvecklare på lägre nivåer dokumenterar och implementerar IF krav. Ansvarsrollen bör därför utgöras av en person som arbetar på en högre nivå. En högre nivå kan också tolkas som senioritet, vilket innebär att utbildning, expertis, kompetens och ledarskap vägs in (Se fråga 18).

5.4.2 Respondenterna som kunde identifiera IF krav

Endast 21% av respondenterna kunde felfritt identifiera alla IF krav i fråga 10 i enkätundersökningen. Av de totala 9 respondenterna var 8 seniora och en mid-level. Inga var juniora, vilket indikerar att personen som kan ansvara för kvalitetssäkringen av IF krav bör vara en mer senior medarbetare. Majoriteten (55%) av respondenterna var mellan 51-65 år gamla och hade roller såsom arkitekt eller applikationskonsult med över 11 års erfarenhet, eller kortare erfarenhet i nuvarande roll som verksamhetsutvecklare eller projektledare, vilka båda är yrken som kräver gedigen erfarenhet av yrket och ledarskap för att kunna anta.

33% var 31-40 år gamla men hade större ansvarsroller såsom mjukvaruansvarig på ett stort företag, CTO på ett medelstort företag, samt kravanalytiker/BA med 8-10 års erfarenhet. Ingen var under 31 eller över 65 år.

Utbildningsnivån visade ingen relevans i frågan då respondenterna hade varierade examen – från självlärd/kurser till masterexamen. Ingen hade dock gymnasieexamen. Däremot var samtliga utom en respondent (som hade +11 års erfarenhet) utbildade inom data/informatik, ibland kombinerat med andra ämnen också. Det kan därför anses viktigt att ha varit utbildad inom rätt ämne, kombinerat med lång erfarenhet av yrket.

2 av de 5 respondenter som aldrig tidigare hade jobbat med krav ingick i dessa 21% som kunde felfritt identifiera de IF kraven, därför verkar inte tidigare erfarenhet av IF krav ha någon relevans för hur väl man kan identifiera IF krav.

Arkitektrollen

Enligt enkätundersökningen var det en enda respondent som stack ut, som svarade att han uppfattar IF krav som väldigt otydliga. Denna person var högutbildad (master) och hade arbetat som arkitekt i över 11 år. Respondenten svarade dessutom att han anser att både IF krav i sig och dokumentationen av dem är viktiga (4 av 5 där en 5:a var "väldigt viktig") i fråga 23 och 24. Hans svar skulle kunna tolkas som att han förstår den otydliga naturen av IF krav som förbises av många och som slutligen leder till dess bortprioritering. Hans svar indikerar dock att det är viktigt att de inte bortprioriteras och att de därför måste dokumenteras på något sätt. Hans förståelse för IF krav bottnar även i att han felfritt kunde identifiera dem och särskilja dem från FK i fråga 10. Kunskapen om dem kan bero på hans utbildning eller hans långa yrkesvana (+11 år), eller kombinationen av både och.

Det fanns en till arkitekt bland respondenterna, som dessutom var produktägare och systemutvecklare. Hon hade arbetat i sin nuvarande roll i 8-10 år och hade en kandidatexamen inom data/informatik (respondent 25). Hon angav att IF krav var väldigt tydliga, men kunde inte utan problem särskilja dem från FK. Hon angav tex att kravet "ska kunna lägga till nya kunder" var ett IF krav, vilket i själva verket är ett FK. Slutligen angav hon en trea av fem möjliga på både fråga 23 och 24 som löd: "Anser du att icke-funktionella krav är viktiga för mjukvaruutvecklingsprojekt?" Respektive "Anser du att dokumentation av icke-funktionella krav är viktigt för mjukvaruutveckling?".

Enligt Behutiye et al. (2022) är produktägaren den som bör engagera sig för IF krav, men enligt enkätundersökningen var det ingen produktägare som kunde identifiera de IF kraven. Enligt Ameller et al. (2012) å andra sidan, är mjukvaruarkitekten den som kan samla in och dokumentera IF krav bäst, då det är en iterativ process som kräver förståelse för hela mjukvaruutvecklingen och inte expertis i delar av den. Vidare ingår det i mjukvaruarkitektens roll att kunna förklara och definiera IF krav för kunden (ibid), vilket enbart den arkitekten med högre utbildning och längre erfarenhet kunde göra (respondent 43). Eftersom kundens förståelse för IF krav är avgörande, är det dessutom gynnande att den ansvariga personen själv förstår hur svårt det är att tydligt förklara IF krav, vilket kan leda till ett större engagemang att godtyckligt sätt förklara för kunden.

5.4.3 Kommunikation

Fråga 20 i enkätundersökningen behandlar frågan om huruvida ansvaret för de IF kraven fördelas mellan *olika* personer. På denna fråga svarade 34,9 % av respondenterna ja; att ansvaret för de IF krav fördelas mellan olika personer, 34,9 % svarade ibland, 18,6 % svarade nej; inte fördelas alls och 11,6 % svarade vet ej om ansvaret fördelas mellan olika personer. Majoriteten fördelar ut ansvar till olika personer antingen alltid eller ibland vilket förstärker ovanstående iakttagelse att praxis inom svenska mjukvaruutvecklingsprojekt är att inte ha en person som ansvarar för de IF kraven utan de samarbetar för att få en helhetsbild. Detta antagande stärks i studien av Behutiye et al. (2020b) och Karhapää et al. (2021). Detta riskerar dock att försaka kvaliteten om att kommunikationen samtidigt inte kan hållas till en viss nivå.

Karhapää et al. (2021) beskriver att det ibland är flera olika personer som dokumenterar de IF kraven i en kravspecifikation, samt flera olika som implementerade de IF kraven och att detta ofta sker utan tydlig kommunikation mellan parterna. Detta kan resultera i felaktig implementering av utvecklare på grund av otillförlitlig dokumentation och att omarbete

behöver göras. En likartad uppfattning framförde Medeiros et al. (2018), då flertalet olika tillvägagångssätt för att dokumentera IF krav kan skapa en negativ inverkan på tilliten till kravspecifikationen. Som tidigare nämnt ökar risken för misskommunikation om flera personer dokumenterar krav. Dessutom påverkar detta dokumentationens kvalitet, då brister och oklarheter kan förekomma. Det återstår att via en god kommunikation lösa dessa brister och oklarheter. Samtidigt finns det alltid rum för individuella uppfattningar hur tydlig dokumentationen än är, vilket understryker antagandet om att tydlig och informativ kommunikation mellan inblandade parter är en viktig aspekt för att kunna lyckas med dokumentationen. Detta antagande stärker Behutiye et al. (2022) som poängterar att gemensamma diskussioner bland teammedlemmar om IF krav minimerar missförstånd och därför förbättrar dokumentation av dessa.

5.5 Sammanfattning av analys och diskussion

Samarbete med kunden är väsentligt, eftersom deras närvaro, samarbete och kunskap bidrar till att presentera och samla in korrekta IF krav. Däremot visade enkätundersökningen att IF krav inte revideras lika ofta som FK, vilket tyder på att kunder är mer måna om att försumma IF krav jämfört med FK. Detta görs möjligen på grund av IF kravens "osynliga" karaktär och kundernas bristande insikt och förståelse av dessa krav som rapporteras i litteraturstudien av författarna Alsaqaf, Wieringa och Daneva (2019). Prioriteringen av FK framför IF krav och tidsbegränsningar bidrar ytterligare till lägre frekvens av revidering av IF krav.

Det har konstaterats enligt studiens enkätundersökning att dokumentationen inom svenska mjukvaruutvecklingsföretag inte alltid är komplett, vilket leder till att utvecklare behöver göra sina egna antaganden utan någon form av systematisk korrekthet eller standardisering. Medeiros et al. (2018) understryker att genom att tydligt och mer specifikt beskriva IF krav inom dokumentationen, kommer utvecklare att få mer specifik information att utgå ifrån under implementeringen. Kommunikationsluckor inom teamet och olika tolkningar av IF krav kan också bidra till utmaningar i dokumentationen.

Enligt studiens enkätundersökning var UC det näst vanligaste tillvägagångssättet för att dokumentera IF krav (44,4 %). Dock visar litteraturstudien att användningen av UC inte var ett av de vanligaste sätten att dokumentera, utan att de som dokumenterade använde US oftast, vilket var det tredje vanligaste tillvägagångssättet med 37 % av respondenterna i enkätundersökningen. Användning av både US och UC kan underlätta förståelsen om IF krav för kunden. Medeiros et al. (2018) anser dock att UC inte är ett optimalt sätt att specificera IF krav, då tillvägagångssättet saknar teknisk koppling och är mer anpassat för dokumentation av FK. Utmaningen med US är att de inte beskriver interaktionen mellan olika IF krav, som kan resultera i att ett helhetsperspektiv saknas. Det krävs även erfarenhet för att skapa US. Dock anses det vara mer fördelaktigt att använda US än UC. Enligt vår studie framkom det att det är en stor utmaning att kunna specificera IF krav på ett strukturerat sätt, så att både kunden och utvecklaren förstår dess innebörd och vikt för mjukvaran.

Det är viktigt att utvecklarna har en generell förståelse för både FK och IF krav enligt Alsaqaf, Daneva och Wieringa (2019). Enligt Ramesh, Cao och Baskerville (2010) kan kravspecifikationen ge en överblick över alla krav. Enligt enkäten var kravspecifikation det vanligaste tillvägagångssättet för att dokumentera IF krav som 74,1 % av respondenterna angav att de använde. Det kan dock vara en bra idé att komplettera den med ett annat tillvägagångssätt som i sin tur beskriver kraven mer i detalj och som dessutom kan erbjuda en djupare förståelse för kunden som saknar tekniskt fackspråk. Således är en kombination av

kravspecifikation och US enligt enkätundersökning och analys anses vara det bästa dokumentationsalternativet.

För att säkerställa att kombinationen av kravspecifikation och US fungerar, bör det införas som en standard att kategorisering av IF-krav alltid görs i kravspecifikationen då det kan främja att inga krav förbises. För att minska risken att informationen i kravspecifikationen inte uppdateras mellan olika dokument kan det också vara fördelaktigt att integrera de båda dokumenten i samma system, så att uppdateringar sker automatiskt.

Beroendet av seniora utvecklarens tysta kunskap ersätter dokumentation, vilket gör verifiering och kvalitetssäkring utmanande. Otydligheten i de IF kraven förvärrar problemet ytterligare. Medan många respondenter hävdar att de har erfarenhet av IF krav, är det bara en liten andel som korrekt identifierar och särskiljer dem från FK. Detta indikerar ett gap mellan upplevd och faktisk förståelse. Enligt enkätundersökningen anser 86 % av respondenterna att IF krav är viktiga och att dokumentation av dem är viktig vilket påvisar att tendensen att dokumentera IF krav finns inom svensk mjukvaruutveckling är positiv. Dock har endast 21 % av respondenterna från de resterande som har en fullständig förståelse för vad IF krav är, vilket kan leda till felaktiga prioriteringar och implementering av krav.

Enligt enkätundersökningen saknas tydligt ansvar för dokumentationen av de IF kraven i svenska mjukvaruutvecklingsprojekt. Majoriteten av svaren indikerar osäkerhet kring ansvaret, vilket tyder på bristande standardisering och riktlinjer. Vidare påvisar enkätundersökningen att ansvaret för IF krav i mjukvaruutvecklingsföretag är ofta fördelat på olika individer, vilket Karhapää et al. (2021) och Behutiye et al. (2020b) bekräftar men tillägger att produktägaren ofta har det övergripande ansvaret. Detta bristande ansvarssystem påverkar kvalitetssäkringen och indikerar att många företag inte uppfyller den svenska standarden för kvalitetssäkring enligt SS-EN ISO 9001:2015. En tydlig ansvarsroll behövs för att undvika brister i dokumentationen och förhindra nedprioritering av IF krav.

Sammanfattningsvis behövs följande kvalifikationer för att kunna kvalitetssäkra IF krav:

- En tydlig ansvarsfördelning och ansvarstagande
- Kunskap om de IF kraven; kunna identifiera dem och särskilja dem från FK samt förståelse för dess inverkan på mjukvaran
- Engagemang till IF krav
- Kunna förklara de IF kraven till kunder på ett sätt så att de förstår och kan specificera de IF kraven
- Kommunikation med kund för att säkerställa att de blir nöjda och är engagerade i produktens IF krav
- Teknisk kompetens för att effektivt kunna kommunicera de IF kraven till utvecklare så att de får till sig den tekniskt tunga information de behöver
- Behärska strategiska/verksamhets kunskaper exempelvis besitta kunskaper inom ledarskap som strategisk kommunikation, verksamhetsutveckling och verksamhetsledning.
- Tidigare erfarenhet av att jobba med IF krav är inte nödvändigtvis meriterande, inte heller utbildningsnivån personen har. Däremot spelar inriktningen på studierna roll och ska helst ha varit data/informatik.
- Senioritet är viktigt för att kunna identifiera IF krav.

5.6 Sammanfattning av nya upptäckter

I följande avsnitt presenteras sammanfattning av nya upptäckter som tillkommit till denna studie utifrån enkätundersökningen i förhållande till det teoretiska ramverket.

Perspektiv på den svenska mjukvaruutvecklingsbranschen

Denna studie fokuserar på dokumentation av IF krav inom svenska mjukvaruutvecklingsprojekt. I den kvalitativa litteraturanalysen förekommer inga beskrivningar om hur dokumentation, ansvar, kvalitetssäkring, tillvägagångssätt eller uppfattningar av IF krav ser ut inom svenska mjukvaruutvecklingsprojekt. Denna studie bidrar till insikt och kunskap om hur dessa områden ser ut inom den svenska mjukvaruutvecklingsbranschen. Det råder både skillnader och likadana upptäckter i både studiens enkätundersökning och i litteraturen som granskats i denna studie.

Enligt enkätundersökningen anser en betydande majoritet inom svensk mjukvaruutveckling att IF krav är viktiga eller mycket viktiga och är positivt inställda till att dokumentera dem, vilket är motsatsen till vad tidigare forskning föreslagit (Jarzebowicz & Weichbroth 2021). Svenska mjukvaruutvecklingsprojekt har en mer positiv attityd till IF krav och dokumentation av dessa.

Andra tillvägagångssätt i Sverige för att dokumentera IF krav än vad tidigare forskning föreslagit

Enligt Alsaqaf, Daneva och Wieringa (2019) är user stories det vanligaste tillvägagångssättet att dokumentera IF krav men enligt vår enkätundersökning är use cases det vanligaste sättet att dokumentera IF krav i svenska mjukvaruutvecklingsprojekt. Det som även skiljer litteraturen från enkätundersökningen i denna studie är att dokumentation av IF krav förekommer i högre grad. Flera olika sätt att dokumentera krav förekommer, vilket även anammats av respondenterna, dock medger Behutiye et al. (2022) att använda flera olika tillvägagångssätt (fråga 13) inte är optimalt då det blir svårare att hålla ordning. Svenska mjukvaruutvecklingsprojekt använder olika sätt att dokumentera från projekt till projekt och har i endast 28,6 % ett standardiserat sätt att dokumentera IF krav på. Flertalet av respondenterna använder sig av textbaserade tillvägagångssättet att dokumentera IF krav såsom Word eller Excel. Dessa tillvägagångssätt betraktas inte som pålitliga enligt litteraturen eftersom de är friliggande ifrån annan mjukvara och uppdateringar kan lätt glömmas bort. Dock anses dessa tillvägagångssätt vara effektiva när kommunikation och effektiv delning med kunden ska hanteras eftersom respondenterna i enkätundersökningen oftast reviderar FK med kunden.

Hur väl IF krav kan kvalitetssäkras i Sverige

För att kunna säkra kvalitet behöver det finnas någon med översiktligt ansvar för att dokumentera de IF kraven. Inom svenska mjukvaruutvecklingsprojekt råder det tvetydigheter om någon har det och vem som har det. Behutiye et al. (2022) beskriver att olika personer kan ha ansvar över de IF kraven och dokumentation av dessa men att det är viktigt med tydliga ansvar för att undvika brister i kommunikationen. Endast 27,9 % av respondenterna anger att de har något med översiktligt ansvar över dokumentationen av IF kraven och därmed inte når upp till den svenska standarden för kvalitetssäkring då det enligt den svenska

standarden (SS-EN ISO 9001:2015) bör finnas någon som har ansvar över hur, när och vad som ska ske.

Enligt Karhapää et al. (2021) är det vanligt att de mer seniora medarbetarna tenderar att förbise dokumentationen och istället förlita sig på sin tysta kunskap, detta i sin tur leder till att de juniora medarbetare som behöver beslutsstöd i form av dokumentation, saknar detta. Dock visar enkätundersökningen att de seniora och mid-level medarbetarna är de som dokumenterar till störst grad och de juniora dokumenterar minst. De juniora är i sin tur de som förlitar sig mest på att de har informationen i huvudet och varken behöver dokumentation eller muntligt stöd.

6 Slutsatser

I följande kapitel presenteras studiens slutsatser både utifrån den kvalitativa litteraturstudien samt den kvantitativa enkätundersökningen. Utifrån den framförda analysen och diskussionen (se kapitel 5) har följande utgångspunkterna identifierats relaterat till forskningsfrågorna. Därefter presenteras en generalisering av de resultat som framkommit i studien om IF krav inom svenska mjukvaruutvecklingsprojekt. Syftet med generaliseringen är att dra bredare slutsatser och reflektera över studiens relevans för liknande sammanhang och framtida tillämpningar. I det följande och sista avsnittet presenteras förslag till framtida forskning.

Slutsatser från den kvalitativa litteraturstudien

Baserat på den genomförda litteraturstudien har vi identifierat flera viktiga teman IF krav i mjukvaruutvecklingsprojekt. De teman som framkommit inkluderar utmaningar med att identifiera IF krav, dokumentationens betydelse för IF krav, faktorer som påverkar kvalitetssäkring av IF krav samt ansvarsroller för att kvalitetssäkra IF krav. Dessa teman har varit vägledande för utformningen och tolkningen av våra egna forskningsresultat.

Den kvalitativa litteratursanalysen visade att IF krav är en viktig del för att lyckas med ett mjukvaruutvecklingsprojekt. De IF kraven är oftast vaga och svåra att definiera och bortprioriteras av såväl kunder som utvecklare. Utvecklarna besitter tyst kunskap om de IF kraven vilket medför att det förekommer mindre dokumentation av dem. Detta gör det svårt för nya utvecklare att sätta sig in i arbetet. Det förekommer olika tillvägagångssätt för att dokumentera IF krav vilket kan medföra att det kan bli svårt att hålla ordning på hur och vart de IF kraven har blivit dokumenterade, vilket i sin tur kan leda till missförstånd och kommunikationssvårigheter inom utvecklingen.

Slutligen beskriver flera författare att dokumentationen av IF krav är begränsad och utmanande både i agila och traditionella mjukvaruutvecklingsprojekt. Den agila metoden förespråkar mindre och mer informell dokumentation medan den traditionella metoden förespråkar att använda olika tillvägagångssätt för att dokumentera krav. Detta för att kunna möta olika behov hos olika intressenter eftersom ett mer flexibelt tillvägagångssätt kan medföra en bredare variation beroende på mål och kontext med mjukvaruutveckling projektet.

I litteraturstudien beskriver flera författare att den som bör ansvara för att kvalitetssäkra de IF kraven bör ha intresse och kunskap om dem. Det är av ytterst vikt att även ledningen såsom chefer och produktägaren inser hur viktigt de IF krav är för mjukvaruutvecklingsprojekt. Orsaken är att de IF kraven är en väsentlig del av att mjukvaruutvecklingsprojekt ska lyckas eller inte. Forskningen visar att det saknas kunskap och tydliga ansvarsområden för att dokumentera IF krav idag. Det behövs särskilda roller för att öka kompetensbehovet såsom säkerhetsexperter, interaktionsdesignare eller användbarhets-experter. Olika roller ansvarar för att dokumentera IF krav och för att överbrygga dessa utmaningar bör det finnas tydliga roller och ansvarsfördelningar för hur de IF kraven ska hanteras och dokumenteras. Ansvarsroller bör ligga på en person som tar både design och tekniska beslut i beaktning men som även besitter både teknisk kompetens och erfarenhet för att kunna ta arkitekturella beslut. Enligt litteraturstudien är denna person mjukvaru- eller lösningsarkitekt.

Slutsatser från den kvantitativa studien

Resultaten från enkätundersökningen visar att respondenterna ofta inte skiljer mellan IF krav och FK i sin dokumentation, vilket tyder på bristande förståelse och svårighet att definiera IF krav. Utmaningen att inte skilja IF krav från FK påverkar förmågan att effektivt dokumentera och prioritera IF krav.

När det kommer till kunskap och förståelse för IF krav avslöjar enkätundersökningen tre huvudsakliga grupper av respondenter baserat på deras förmåga att identifiera och särskilja IF krav. Denna information ger en djupare förståelse för den generella kompetensnivån inom området samt betonar vikten av att adressera kunskapsbrister för att förbättra kvalitetssäkring och dokumentation av IF krav i mjukvaruprojekt. Resultaten antyder att en mer standardiserad och utförlig utbildning och vägledning om IF krav kan vara nödvändig för att förbättra förmågan att identifiera och dokumentera dessa krav korrekt.

Enkätundersökningen visar också att vissa av respondenterna saknar gedigen kunskap om IF krav, trots att de arbetar i större organisationer som borde ge nödvändiga förutsättningar för effektiv dokumentation. Frånvaron av en gemensam uppfattning om IF krav inom organisationer bidrar ytterligare till utmaningarna med att specificera och kvalitetssäkra deras dokumentation. Dessutom visar enkätundersökningen att respondenterna använder flera olika tillvägagångssätt för att dokumentera IF krav, då resultatet visade att i genomsnitt användes tre olika tillvägagångssätt. Användningen av flera olika tillvägagångssätt indikerar att det finns en potentiell utmaning i att hålla reda på information, vilket ökar risken för att information går förlorad eller glöms bort mellan olika dokument.

Slutligen visade studiens enkätundersökning att det behövs ytterligare kunskapsutveckling för att främja samarbete mellan anställda. Beroendet på tyst kunskap introducerar subjektivitet och riskerar också felaktig prioritering.

Fråga 19 undersökte ansvarsfördelningen för dokumentation av IF krav, där varierande ansvar förekom: 27,9% angav översiktligt ansvar, 25,6% ibland, 30,2% inget ansvar, och 16,3% visste inte. Brist på standardisering och liknande situationer rapporterades även av andra studier. Enligt fråga 20 saknade majoriteten (65,1%) tydlig fördelning av översiktligt ansvar, antyder kunskapsbrist. Detta är problematiskt för att säkerställa dokumentationskvalitet enligt svensk standard. Även om majoriteten ansåg IF krav viktiga (86% enligt fråga 23), hade endast 21% full förståelse för dessa krav (fråga 10), vilket pekar på kunskapsklyfta. Det kan leda till felaktig kravprioritering och implementering. Trots att det finns en majoritet (92%) av teknisk utbildning och kompetens (frågor 3 och 4) för att kunna kvalitetssäkra IF krav, var det en minoritet (21%) som påvisade förståelse för IF krav.

Sammantaget finns det ett behov av bättre kunskapsspridning, flera förbättrade riktlinjer och förbättrade verifieringsprocesser för att dokumentera IF krav.

Utifrån analys och diskussion bör ansvarsrollen utföras av någon med lång erfarenhet av yrket, personen behöver inte ha erfarenhet av IF krav men bör ha studerat data/informatik. Vidare är det enligt Behutiye et al. (2022) viktigt att personen har insikt i hur viktigt det är med IF krav. Personen behöver aktivt kunna stödja och förklara betydelsen och definitionen av IF kraven för kunden. I de agila projekten förväntas utvecklaren ha kontinuerlig kommunikation med kunden för att samla in, utveckla och implementera kraven, vilket den ansvarige bör överse för att säkerställa att kvaliteten upprätthålls. Därmed bör den ha mycket

god teknisk förståelse för mjukvaran och kunna fatta arkitekturella beslut samtidigt som den bör kunna översätta de tekniska termerna och ändringar till kunden. Att hålla kunden involverad är viktigt för kvalitetssäkringen.

Vår slutsats är att en arkitekt med lång yrkeserfarenhet och högre utbildning (minst en master), eller en mer senior person i företaget, kan vara den mest kompetenta personen att samla in och dokumentera de IF kraven. En mjukvaruarkitekt kan förstå det övergripande mjukvarusystemet och fatta arkitekturella beslut i kommunikation med systemutvecklaren som bör ha kontakt med kunden. Mjukvaruutvecklaren kan agera medlare och teknisk översättare mellan kund och utvecklare. Denne kan stödja kundens beslut genom att på ett godtyckligt sätt informera och förklara vikten av IF krav, då kundens beslut är avgörande. Vidare är kunden central när det gäller att samla in och dokumentera IF krav. Alternativt skulle en lösningsarkitekt kunna hjälpa till att nå en samsyn mellan kund och leverantör och säkerställa tekniskt och ekonomiskt genomförbara krav. Deras expertis i att översätta tekniska termer och förstå kundernas behov är också värdefull.

6.1 Generalisering

Enligt Jacobsen (2017) är det viktigt att tillräckligt många respondenter deltagit i den extensiva undersökningen för att den ska bli generaliserbar, dock *kan* generaliseringar göras även utifrån färre enheter. Baserat på den extensiva undersökningsdesignen har en statistisk generalisering kunnat göras (se avsnitt 2.5.2), vilket betyder att de enheter som undersökts kan generaliseras till andra enheter som inte undersökts, på en empirisk nivå (ibid).

Populationsgeneralisering

Då studien utgjorts av en mixed-method ingår både den kvantitativa och kvalitativa undersökningen i generaliseringen. Vanligtvis är det svårt att generalisera utifrån kvalitativa undersökningar enligt Jacobsen (2017), men då samstämmighet mellan delar av det teoretiska ramverket och enkätundersökningen kunde påvisas så ökar istället tillförlitligheten och generaliserbarheten till andra situationer eller populationer. Å andra sidan fanns det variabler som skilde sig från teorin, dessa variabler påvisar i sin tur hur den svenska populationen skiljer sig från en större population.

De generaliseringar som kunde göras gäller för populationen i svenska mjukvaruutvecklingsprojekt som hanterar krav, där hälften är kvinnor, hälften män. Enligt SCB stod kvinnor för 30% av arbetskraften inom IT-branschen år 2021 (TechSverige u.å.), vilket skiljer sig från genusfördelningen i enkäten (51% kvinnor; 49% män). Skillnaden kan bero på att fler kvinnor jobbar idag med krav än män inom mjukvarubranschen och ville därför till större grad besvara enkäten, eller också att kvinnor inom mjukvarubranschen är mer benägna att besvara enkäter från studenter. Detta är ett resultat från urvalet som diskuterades i avsnitt 2.4.1. Enkäten visade även att icke-binära var tydligt underrepresenterade (0 av 43 respondenter), vilket är ett välkänt faktum inom tech-branschen (AnitaB.Org 2022).

Åldersfördelningen var jämnt fördelad mellan 26-65 år för kvinnor, med något fler i åldersgruppen 40-51 år. De flesta männen är mellan 31-40 år gamla. Gruppen av män mellan 31-40 år (48%) är mer än dubbelt så stor som någon annan åldersgrupp, inklusive kvinnornas åldersgrupper. Inga personer är under 26 år gamla och fåtal (<5%) är över 66 år gamla.

Statistisk generalisering

Resultaten från studien visade på vikten av dokumentation för att säkerställa kvaliteten av IF krav inom mjukvaruutvecklingsprojekt. Dokumentationens roll som ett verktyg för att minimera kunskapsförlust och underlätta samarbete mellan teammedlemmar visade sig vara avgörande för att undvika missförstånd och felaktiga tolkningar av kraven. Denna insikt är inte bara relevant för svenska mjukvaruutvecklingsprojekt utan kan också vara tillämplig på liknande arbetsmiljöer där dokumentation spelar en nyckelroll i kravhantering och projektkommunikation.

En annan slutsats som kan generaliseras är det identifierade kompetensbehovet för dokumentation av IF krav. Studien påvisade att bristen på en gemensam förståelse och tydliga ansvarsområden för IF krav skapade utmaningar för kvalitetssäkringen av IF kraven. Därmed pekar resultaten på behovet av att tillsätta särskilda roller, som exempelvis seniora mjukvaru- eller lösningssarkitekter, för att stärka kompetensen inom dokumentation av IF krav. Denna insikt kan vara relevant inte bara för svenska mjukvaruutvecklingsprojekt utan också för organisationer inom olika branscher som strävar efter att förbättra sin kravhanteringsprocess och säkerställa hög kvalitet på sina mjukvaruprodukter.

Resultaten av den kvantitativa undersökningen visade ett intressant samband mellan senioritet och förmågan att identifiera IF krav felfritt. Samtliga respondenter som kunde identifiera IF krav utan fel ansågs seniora inom sina företag. Denna observation ger en indikation på att erfarenhet och yrkesmässig mognad kan spela en betydande roll när det gäller att säkerställa korrekt tolkning och hantering av IF krav. Det är värt att notera att denna aspekt inte diskuterades i den kvalitativa undersökningen, där istället fokus låg på de seniorernas förmåga att besitta tyst kunskap om IF krav hade en negativ påverkan för de juniora utvecklarna, då de saknade dokumentation. Den statistiska generaliseringen antyder därför ett potentiellt samband mellan kompetensnivå och precision i tolkningen av dessa krav, vilket ger ytterligare djup till förståelsen av vad som krävs för att kvalitetssäkra IF krav.

En särskild aspekt som framkom under studien var utmaningarna och möjligheterna med agil dokumentation av IF krav. Både bristen på formell dokumentation och den flexibla ansatsen för dokumentation kan ha både positiva och negativa konsekvenser för kvalitetssäkringen av de IF kraven. Denna insikt ger en djupare förståelse för de unika utmaningarna inför IF krav som agila utvecklingsmetoder kan generera och ger möjligheter för ytterligare forskning och undersökningar inom detta område.

6.2 Framtida forskning

Forskning visar att uppfattning av IF krav kan vara otydlig och att det förekommer mindre dokumentation av IF krav på grund av den tysta kunskapen som finns. Mer forskning behövs inom dessa områden. Den tysta kunskapen behöver på något sätt dokumenteras för att inte gå förlorad. Det hade varit intressant att göra kvalitativa intervjuer med medarbetare inom mjukvaruutveckling, speciellt med seniora utvecklare för att se hur de dokumenterar och för vidare sin tysta kunskap gällande IF krav.

Behutiye är den författare som har bedrivit mycket forskning kring IF krav och dess dokumentation och har även framfört förslag på riktlinjer för hur dokumentationen IF krav ska ske effektivt. Dessa riktlinjer hade varit intressant att implementera inom svenska mjukvaruutvecklingsprojekt eftersom det förekommer mindre standardiserade sätt att

dokumentera IF krav inom svensk mjukvaruutveckling. Det behövs mer forskning om hur ett standardiserat tillvägagångssätt för att dokumentera och kvalitetssäkra IF krav kan se ut. Mer forskning skulle kunna ta fram ett ramverk för just den disciplinen.

Det saknas forskning som påvisar vilken kunskap som behövs för att kunna fatta bättre beslut, samla in och dokumentera IF krav och dessutom förstå dess natur bättre så att de på ett mer effektivare sätt kan identifieras. Majoriteten av respondenterna i denna studie har IT-relaterad utbildning men hade ändå bristande kunskaper om IF krav så eventuell fortsatt forskning kan innebära vilken akademisk kunskap kan behövas tillföras till kvalitetssäkring av IF krav? Finns det tillräckligt med kunskap om de IF kraven i de svenska universiteten idag? För att juniora utvecklare ska kunna skapa nya banor som inte lagts av de seniora utvecklarna som besitter tyst kunskap är det viktigt att de själva är tillräckligt utbildade innan de kommer ut i arbetslivet. Samtidigt måste det bli en attraktiv kunskap som rekryterare förstår är värdefull.

Mycket av forskningen problematiserar relationen mellan kund och dokumentation och flera författare menar att det behövs ett effektivt sätt att *dela* dokumentation av IF krav med kund. Vi har delvis undersökt området men på grund av avgränsningar har vi inte studerat ämnet på djupet, dock menar vi att mer forskning behövs. De hade varit intressant att se hur svenska mjukvaruutvecklingsföretag kommunicerar dokumentation av IF krav sinsemellan och om kunden är involverad i en mindre eller större omfattning, vilket den agila systemutvecklingsmetoden förespråkar men som nuvarande forskning visar på är bristfällig.

Enligt forskning kan tidsbegränsningar och tidsbrist påverka dokumentationen genom att otydliga eller ospecificerade krav dokumenteras. Ett annat intressant forskningsområde är om det skulle kunna finnas en automatiserad process för att dokumentera och hantera IF krav. Baserat på våra slutsatser i denna studie kan det med god sannolikhet tänkas att AI skulle kunna stödja flera processer i hantering av IF krav, men att människans kunskap om IF krav ändå måste prioriteras över AI i utvecklingen. För att kunna lära upp AI hur den ska hantera IF krav måste människan bli bättre på att definiera IF krav först. Därefter finns det goda chanser att en AI kan identifiera nya IF krav som uppkommer mycket snabbare än vad en människa kan göra, vilket skulle stödja processen att dokumentera dem eftersom det kräver identifiering först och främst. Dessutom kan en AI bättre stödja prioriteringen om den är opartisk.

Källförteckning

Agila manifestet (2001). *Manifest för Agil systemutveckling*. Ward Cunningham. <https://agilemanifesto.org/iso/sv/manifesto.html> [2023-02-19]

Ali, A., Khalil, I., Ahmad, I., Parveen, I. & Zaman, U. (2022). Role of Non-functional Requirements in projects' success. *2nd International Conference on Digital Futures and Transformative Technologies (ICoDT2)* | 978-1-6654-9819-7/22/ IEEE | DOI: 10.1109/ICoDT255437.2022.9787463

AltexSoft (2022). *Non-functional Requirements: Examples, Types, How to Approach*. <https://www.altexsoft.com/blog/non-functional-requirements/> [2023-05-12]

Alsaqaf, W., Daneva, M., & Wieringa, R. (2019). Quality requirements challenges in the context of large-scale distributed agile: An empirical study. *Information and software technology*. [Online] 11039–55. DOI: 10.1016/j.infsof.2019.01.009.

Ameller, D., Ayala, C., Cabot, J., & Franch, X. (2012). 'How do software architects consider non-functional requirements: An exploratory study', in *2012 20th IEEE International Requirements Engineering Conference (RE)*. [Online]. 2012 IEEE. pp. 41–50. DOI: 10.1109/RE.2012.6345838

AnitaB.Org – Global Organization for Women and Technology. (2022). *2021 Technical Equity Experience Survey*. <https://anitab.org/wp-content/uploads/2022/05/2021-TechEES.2022.pdf>. [2021-08-07]

Atlassian (2023). *What is Confluence Cloud?* <https://support.atlassian.com/confluence-cloud/docs/what-is-confluence-cloud/> [2023-05-03]

BA Video Series Podcasts (2020). Non-Functional Requirements discussion [podcast], november. <https://open.spotify.com/episode/7MnnWfCBfCcXXoqiBIYe9K?si=32d980a637324057>. [2023-04-08]

Behutiye, W., Karhapää, P., Costal, D., Oivo, M., & Franch, X. (2017). 'Non-functional Requirements Documentation in Agile Software Development: Challenges and Solution Proposal', in *arXiv.org*. [Online]. 2017 Ithaca: Cornell University Library, arXiv.org. p. DOI: 10.1007/978-3-319-69926-4_41

Behutiye, W., Karhapää, P., López, L., Burgués, X., Martínez-Fernández, S., Vollmer, A. M., Rodríguez, P., Franch, X., & Oivo, M. (2020a). Management of quality requirements in agile and rapid software development: A systematic mapping study. *Information and software technology*. [Online] 123106225–. DOI: 10.1016/j.infsof.2019.106225

Behutiye, W., Rodriguez, P., & Oivo, M. (2022). Quality Requirement Documentation Guidelines for Agile Software Development. *IEEE access*. [Online] 1070154–70173. DOI: 10.1109/ACCESS.2022.3187106

- Behutiye, W., Rodríguez, P., Oivo, M., Aaramaa, S., Partanen, J., & Abhervé, A. (2022). Towards optimal quality requirement documentation in agile software development: A multiple case study. *The Journal of systems and software*. [Online] 183111112–. DOI: 10.1016/j.jss.2021.111112
- Behutiye, W., Seppänen, P., Rodríguez, P., & Oivo, M. (2020b). ‘Documentation of Quality Requirements in Agile Software Development’, in *Proceedings of the Evaluation and Assessment in Software Engineering*. [Online]. 2020 Ithaca: ACM. pp. 250–259. DOI: 10.1145/3383219.3383245
- Bjarnason, E., Unterkalmsteiner, M., Borg, M., & Engström, E. (2016). A multi-case study of agile requirements engineering and the use of test cases as requirements. *Information and software technology*. [Online] 7761–79. DOI: 10.1016/j.infsof.2016.03.008
- Browne, G. J., & Ramesh, V. (2002). Improving information requirements determination: a cognitive perspective. *Information & management*. [Online] 39 (8), 625–645. DOI: 10.1016/S0378-7206(02)00014-9
- Chazette, L., Brunotte, W., & Speith, T. (2022). Explainable software systems: from requirements analysis to system evaluation. *Requirements engineering*. [Online] 27 (4), 457–487. DOI: 10.1007/s00766-022-00393-5
- Cronholm, S. & Hjalmarsson, A. (2021). Experiences from Sequential Use of Mixed Methods. *Electronic Journal of Business Research Methods*, 9(2), 87-95.
- Dennis, R., Roth, R. M., & Wixom, B. H. (2010). *Systems analysis and design (5th ed.)*. John Wiley
- Domah, D., & Mitropoulos, F. J. (2015). ‘The NERV methodology: A lightweight process for addressing non-functional requirements in agile software development’, in SoutheastCon 2015. [Online]. 2015 *IEEE*. pp. 1–7. DOI: 10.1109/SECON.2015.7133028
- Eckhardt, J., Vogelsang, V. & Fernandez, D., M. (2016). Are “Non-functional” Requirements really Non-functional?. *IEEE/ACM 38th IEEE International Conference on Software Engineering*. DOI: 10.1145/2884781.2884788
- Ekonomifakta. (2023). *Företagens storlek*.
<https://www.ekonomifakta.se/fakta/foretagande/naringslivet/naringslivets-struktur/>
 [2023-04-25]
- Eliasson, A. (2013). *Kvantitativ metod från början*. (Upplaga 3:3). Lund: Studentlitteratur.
- Elshandidy, H., & Mazen, S. (2013). Agile and traditional requirements engineering: A survey. *International Journal of Scientific & Engineering Research*, 4(9), 473-482. DOI:10.14299/ijser.2013.09.002
- Eriksson, U. (2008). *Test och kvalitetssäkring av IT-system*. 2. uppl. Lund: Studentlitteratur
- Glinz, M. (2007). ‘On Non-Functional Requirements’, in *15th IEEE International Requirements Engineering Conference (RE 2007)*. [Online]. 2007 IEEE. pp. 21–26.

DOI: 10.1109/RE.2007.45

Google Ngrams Viewer (u.å.). *Search in Google Books*. https://books.google.com/ngrams/graph?content=non-functional+requirements%2Cfunctional+requirements%2Crequirements+engineering&year_start=1960&year_end=2019&corpus=en-2019&smoothing=1 [2023-04-19]

Greve, J. (2021). *Uppsats med kvantitativ ansats*. (Upplaga 1:1). Lund: Studentlitteratur.

Heinsohn Business Technology (u.å.) *A Guide to Seniority Levels*. <https://www.us.heinsohn.co/blog/seniority-level/> [2023-08-16]

Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior*. [Online] 51 (Part B), 915–929. DOI: 10.1016/j.chb.2014.10.046

Jacobsen, D.I. (2017). *Hur genomför man undersökningar?: Introduktion till samhällsvetenskapliga metoder*. (Upplaga 2:1). Lund: Studentlitteratur.

Jarzebowicz, A., & Weichbroth, P. (2021). A Qualitative Study on Non-Functional Requirements in Agile Software Development. *IEEE access*. [Online] 940458–40475. DOI: 10.1109/ACCESS.2021.3064424

Karhapää, P., Behutiye, W., Rodríguez, P., Oivo, M., Costal, D., Franch, X., Aaramaa, S., Choraś, M., Partanen, J., & Abherve, A. (2021). Strategies to manage quality requirements in agile software development: a multiple case study. *Empirical software engineering: an international journal*. [Online] 26 (2). DOI: 10.1007/s10664-020-09903-x

Kocak, S. A. (2013). Green software development and design for environmental sustainability. In 11th International Doctoral Symposium an *Empirical Software Engineering (IDOESE 2013)*. Baltimore, Maryland (Vol. 9).

Larman, C. (2005). *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development*. 3. ed. Upper Saddle River, N.J.: Prentice Hall PTR

Martins, F. H., Carvalho de Oliveira Junior, A., Dias Canedo, E., Dias Kosloski, R. A., Ávila Paldês, R., & Costa Oliveira, E. (2019). Design Thinking: Challenges for Software Requirements Elicitation. *Information (Basel)*. [Online] 10 (12), 371–. DOI: 10.3390/info10120371

Medeiros, J., Vasconcelos, A., Silva, C., & Goulão, M. (2018). Quality of software requirements specification in agile projects: A cross-case analysis of six companies. *The Journal of Systems and Software*, 142, 171–194. DOI: 10.1016/j.jss.2018.04.064

Michel, J. B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Google Books Team, Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M. A., & Aiden, E. L. (2011). Quantitative analysis of culture using millions of digitized books. *Science (New York, N.Y.)*, 331(6014), 176–182. DOI: 10.1126/science.1199644

Montgomery, L., Fucci, D., Bouraffa, A., Scholz, L., & Maalej, W. (2022). Empirical research on requirements quality: a systematic mapping study. *Requirements engineering*. [Online] 27 (2), 183–209. DOI: 10.1007/s00766-021-00367-z

Ochodek, M. & Kopczyńska, S. (2018). Perceived importance of agile requirements engineering practices – A survey. *The Journal of systems and software*. [Online] 14329–43. DOI: 10.1016/j.jss.2018.05.012

Patel, R., Davidson, B. (2019). *Forskningsmetodikens grunder*. (Upplaga 5). Lund: Studentlitteratur

Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information systems journal* (Oxford, England). [Online] 20 (5), 449–480. DOI: 10.1111/j.1365-2575.2007.00259.x

Recker, J. (2021). *Scientific Research in Information Systems: A Beginner's Guide*. Springer International Publishing AG.

Riksrevisionen (2011). Statliga IT-projekt som överskrider budget (RiR 2011:5)
GRANSKNINGSRAPPORT 13 JANUARI 2011
<https://www.riksrevisionen.se/rapporter/granskningsrapporter/2011/statliga-it-projekt-som-overskrider-budget.html> [2023-03-28]

Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*. 35(3), ss. 8–13. <https://doi.org/10.1145/1764810.1764814>

Shafiq, M., Qinghua, Z., Akbar, M. A., Khan, A. A., Hussain, S., Amin, F., Khan, A., Soofi, A. A. (2018). Effect of Project Management in Requirements Engineering and Requirements Change Management Processes for Global Software Development. *IEEE Access*, 6(), ss. 25747-25763. DOI: 10.1109/ACCESS.2018.2834473

Shalloway, A., Beaver, G., & Trott, J. (2010). *Lean-agile software development: achieving enterprise agility*. Upper Saddle River, NJ: Addison-Wesley

Sharma, S., Sarkar, D., & Gupta, D. (2012). Agile Processes and Methodologies: A Conceptual Study. *International Journal on Computer Science and Engineering (IJCSE)*. ISSN : 0975-3397

SS-EN ISO 9001:2015. *Ledningssystem för kvalitet - Krav*. Svenska institutet för standarder (SIS).

TechSverige (u.å.). *Statistik kvinnor och män*. <https://www.techsverige.se/> [2023-08-06]

The Standish Group International. (2015). *Chaos Report 2015*. https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf [2023-02-14]

Webster, J. & Watson, R.T. (2002). "Analyzing the past to prepare for the future: Writing a literature review". *MIS Quarterly*, vol. 26, no. 2, pp. 11.

Wohlin, C., Šmite, D., & Moe, N. B. (2015). A general theory of software engineering: Balancing human, social and organizational capitals. *The Journal of Systems and Software*, 109, 229–242. DOI: 10.1016/j.jss.2015.08.009

Bilaga 1 - Enkätfrågor

1. Hur definierar du dig?
 - Kvinna
 - Man
 - Icke-binär
 - Vill ej ange
2. Hur gammal är du?
 - 19-25
 - 26-30
 - 31-40
 - 41-50
 - 51-65
3. Vilken typ av utbildning har du? Ange det alternativ som är närmast din utbildning eller ange ett eget svar. Ange din senaste avklarade utbildning. Kryssa i fler alternativ om inget enstaka alternativ beskriver din senaste avklarade utbildning. Tex: UX-designers kan välja både Data/Informatik och Media, design.
 - Data/Informatik
 - Teknik
 - Ledarskap, administration
 - Media, design
 - Juridik
 - Samhällsvetenskap
 - Ekonomi, affärsutveckling, marknadsföring
 - Annat...
4. Vilken är din senaste avklarade nivå av utbildning?
 - Sjävlärd eller enstaka studier
 - Gymnasium
 - Yrkeshögskola/Yrkesutbildning
 - Kandidatexamen
 - Magisterexamen
 - Masterexamen
 - Doktorsexamen
5. Vilken roll har du idag?
 - Systemutvecklare
 - Kravanalytiker
 - Business Analyst
 - Projektledare
 - Programmerare
 - Arkitekt
 - Applikationskonsult
 - Produktägare
 - Testare
 - UX-designer
 - Annat...
6. Hur länge har du arbetat i din nuvarande roll?
 - 0-1 år
 - 2-4 år
 - 5-7 år

- 8-10 år
 - 11 +
7. Hur stort är företaget du jobbar på idag?
- Stort (Över 250 anställda)
 - Medel (ca 50-249 anställda)
 - Litet (10-49 anställda)
 - Mikroföretag (1-9 anställda)
8. Skulle du säga att du förstår vad man syftar till när man pratar om icke-funktionellt krav?
- 1 - Det är väldigt tydligt för mig
 - 2
 - 3
 - 4
 - 5 - Det är väldigt otydligt för mig
9. Har du tidigare jobbat med icke-funktionella krav?
- Ja
 - Nej
 - Vet ej
10. Nedan följer några exempel på krav. Markera det(t) alternativ du anser är icke-funktionella krav
- Obehöriga ska inte kunna komma in i systemet
 - Systemet ska uppfattas som seriöst och tillförlitligt
 - Systemet ska vara kompatibelt med våra andra system
 - Ska kunna lägga till nya kunder
 - En användare ska få bekräftelse-mejl efter ett köp
 - Man måste kunna återanvända koden för att bygga ett nytt system
 - Systemet ska kunna skapa statistik över sålda produkter
11. Dokumenterar ni krav i det företag du arbetar på idag?
- Ja
 - Nej
 - Ibland
 - Vet ej
12. Definierar ni vilka krav som är funktionella och icke-funktionella vid dokumentation?
- Ja
 - Nej
 - Vet ej
13. Om du svarade ja på föregående fråga, hur dokumenterar ni icke-funktionella krav idag? (Flera val är möjliga)
- Use Cases
 - Kravspecifikation
 - Backlog
 - Excel
 - Word
 - Features
 - Wiki page
 - Skriver på whiteboard eller papper
 - Scrum
 - Devops
 - User Stories
 - Annat...

14. Är ovanstående svar er standard för att dokumentera icke-funktionella krav, eller varierar era tillvägagångssätt från projekt till projekt?
- Det är vår standard
 - Det varierar från projekt till projekt
15. Till vilken grad är det tydligt vad som ska implementeras från de icke-funktionella kraven?
- 1 - Väldigt tydlig
 - 2
 - 3
 - 4
 - 5 - Väldigt otydligt
16. Till vilken grad är det lätt/svårt för dig att dokumentera vad som ska implementeras från de icke-funktionella kraven
- 1 - Väldigt lätt
 - 2
 - 3
 - 4
 - 5 - Väldigt svårt
17. Hur ofta händer det att ni reviderar kraven (tillsammans med kund) innan ni går vidare till implementationsfasen?
- Funktionella krav
 - Icke-funktionella krav
- Alternativ
- Aldrig
 - Sällan
 - Ofta
 - Mycket ofta
18. Vad utgår du ifrån när du implementerar icke-funktionella krav? (När du väl sitter och ska börja implementera). Ange det svaret som är vanligaste utgångspunkten för dig. Om frågan ej är applicerbar på din roll kan du skippa frågan.
- Jag minns informationen - Jag har kraven i huvudet och utgår från det när jag implementerar
 - Skriftlig information - Ett dokument med alla krav som jag följer och kontrollerar
 - Verbal information - Gruppmöte med vårt team som bestämmer vad som ska göras, sen gör jag det
 - Annat...
19. Har någon ett översiktligt ansvar specifikt för dokumentation av de icke-funktionella kraven?
- Ja
 - Nej
 - Ibland
 - Vet ej
20. Fördelar ni ansvaret för icke-funktionella krav mellan olika personer?
- Ja
 - Nej
 - Ibland
 - Vet ej
21. Grupperar ni de icke-funktionella kraven efter vilken kategori de tillhör? Tex. alla krav som rör säkerhet grupperas för sig, användbarhetskrav för sig, osv.

- Ja
 - Nej
 - Ibland
 - Vet ej
22. Har ni något tillvägagångssätt för att säkerställa att alla har samma uppfattning/bild av de icke-funktionella kraven? Tex se till att alla implementerar samma nivå av användbarhet.
- Ja
 - Nej
 - Ibland
 - Vet ej
23. Anser du att icke-funktionella krav är viktiga för mjukvaruutvecklingsprojekt?
- 1- Nej, inte alls viktigt
 - 2
 - 3
 - 4
 - 5 - Ja, väldigt viktiga
24. Anser du att dokumentation av icke-funktionella krav är viktigt för mjukvaruutveckling?
- 1- Nej, inte alls viktigt
 - 2
 - 3
 - 4
 - 5 - Ja, väldigt viktig



HÖGSKOLAN I BORÅS

Besöksadress: Allégatan 1 · Postadress: 501 90 Borås · Tfn: 033-435 40 00 · E-post: registrator@hb.se · Webb: www.hb.se