

# God användbarhet med Scrum

En studie av ISO 9241-anpassad systemutveckling

Kandidatuppsats, 15 högskolepoäng, INFK01 i Informatik

*Framlagd:* Juni, 2009

*Författare:* Daniel Acs  
Anna Rengemo

*Handledare:* Claus Persson

*Examinator:* Eric Wallin  
Lars Fernebro

<b>Titel:</b>	God användbarhet med Scrum En studie av ISO 9241-anpassad systemutveckling
<b>Författare:</b>	Daniel Acs Anna Rengemo
<b>Utgivare:</b>	Institutionen för informatik
<b>Handledare:</b>	Claus Persson
<b>Examinator:</b>	Eric Wallin Lars Fernebro
<b>Publiceringsår:</b>	2009
<b>Uppsattstyp:</b>	Kandidatuppsats
<b>Språk:</b>	Svenska
<b>Nyckelord:</b>	ISO 9241, Scrum, användbarhet, informatik, systemutveckling,.

## **Abstract**

Användbarhet är en viktig aspekt vid utvecklandet av ett system. Många anser att Agila systemutvecklingsmetoder inte stödjer användbar systemutveckling. Bakgrunden till denna studie är därför att undersöka hur den populära agila metoden Scrum hanterar användbarhetsfrågor. Som utgångspunkt för vår studie har vi använt ISO 9241 kapitel 11 – *Guidance on usability*.

Genom intervjuer med systemutvecklare som har erfarenhet av Scrum har vi samlat in den empiri som vår uppsats grundas på. Vi har valt utvecklare med varierad bakgrund som arbetar inom olika organisationer för att få en så bred bild som möjligt av hur användbarhetsproblem hanteras ute i verkligheten.

Våra slutsatser är att Scrum idag inte lever upp till de krav som ställs i ISO 9241 beträffande användbarhet. Dock ser vi att genom viss modifikation av metoderna som används i Scrum, går det att säkerställa god användbarhet enligt den standard som ställs i ISO 9241.

## Förord

Först och främst vill vi tacka Claus Persson för vägledning och stöd under arbetet med denna uppsats. Vi vill även tacka Klas Skogmar, Jonas Wallander och Michael Lindberg som ville ställa upp och bidra med erfarenheter och information från verkligheten, utan dem vore denna uppsats omöjlig att färdigställa. Vi tackar även Patrik Andersson för idéer och feedback. Slutligen ett stort tack till våra familjer och vänner för allt stöd under terminens gång.

*Daniel Acs och Anna Rengemo*

# Innehållsförteckning

1 Inledning .....	1
1.1 Bakgrund.....	1
1.3 Syfte .....	2
1.4 Avgränsning .....	2
1.5 Tillvägagångssätt.....	3
2 Litteraturgenomgång .....	4
2.1 Användbarhet .....	4
2.2 ISO 9241-11 .....	7
2.2.1 Designprocessen enligt ISO9241-11 .....	8
2.2.2 Kriterier och mätbara värden .....	9
2.3 Scrum .....	10
2.4 Koppling mellan Scrum och användbarhet.....	13
2.5 Ramverk.....	14
3 Metod .....	16
3.1 Metodval .....	16
3.2 Intervju .....	17
3.3 Urval av intervjuobjekt .....	19
3.4 Analys .....	19
3.5 Yttre påverkan vid intervju .....	20
3.6 Källkritik.....	20
3.7 Reliabilitet och validitet.....	21
4 Empiri .....	23
4.1 Presentation av informanter .....	23
4.2 Resultat .....	24
5 Analys .....	30
5.1 Användbarhetsproblem .....	30
5.2 Användbarhetsprocesser .....	30
5.3 Användbarhetslösningar.....	32
6 Slutsatser .....	34
6.1 Besvarande av undersökningsfrågor .....	34
6.2 Ytterligare slutsatser .....	35
6.3 Kunskapsbidrag.....	35

Bilagor.....	36
Bilaga 1 - Exempel.....	36
Bilaga 2 - Intervjuguide .....	38
Bilaga 3 – Intervju 1 .....	40
Bilaga 4 – Intervju 2 .....	50
Bilaga 5 - Intervju 3 .....	58
Litteraturförteckning .....	65

# 1 Inledning

*I detta inledande kapitel beskriver vi bakgrunden till vårt uppsatsämne. Vi presenterar även problemställning, syfte och avgränsning. Slutligen beskriver vi kortfattat vårt tillvägagångssätt i arbetet med denna uppsats.*

---

## 1.1 Bakgrund

Att öka effektiviteten och/eller kvaliteten är oftast målsättningen när företagen inför nya informationssystem. För att klara av att leverera detta krävs bra och relevanta funktioner. Men även god användbarhet är viktigt för att arbetsuppgifterna ska kunna utföras på ett effektivt sätt. (Allwood, 1998) Trots att nästan alla system har som syfte att stödja användare i deras behov att utföra specifika uppgifter är mjukvaruutvecklare allt för ofta dåliga på att möta användarnas krav på användbara gränssnitt och bra interaktionsdesign. Agila metoder så som Scrum har som mål att snabbt utveckla system med god kvalitet. Det ska uppnås genom nära kontakt med produktägaren och inte genom analys och designartefakter. (Constantine & Lockwood, 2003) Utvecklingsprojekt med agila metoder är ofta lyckade. Men ingen av de största agila metoderna har processer för användbarhet i metoden (Kane, 2003). Forskning har visat att rena Scrum metoder inte ger tillräckliga möjligheter att bygga prototyper och att undersöka kraven innan utvecklingsprocessen startar (Singh, 2008). Trots att Scrum ger små möjligheter till att bygga prototyper och validera kravspecifikationer från kunden finns det är en stor variation i hur organisationer och företag försöker inkludera användbarhet i sina processer.

ISO 9241 innehåller kriterier för användbarhet inom både hårdvara, mjukvara och processer i utvecklingsprojekt, vilket vi anser gör den relevant att använda i vår undersökning. Det är också en allmänt accepterad standard som ger konkreta kriterier på ett komplext ämne som användbarhet. Nackdelar som poängteras med den är att det ofta förekommer att ISO 9241 standarden är citerad istället för läst. Största anledningen till detta är att standarden är dyr att köpa in, samt svår att få tag på (Travis, 2004). Standarden består av sjutton omfattande delar, av dem är det främst kapitel 11 - *Guidance on usability* som är relevant för vår undersökning.

En fördel med att använda sig av ISO 9241 standarden i vår undersökning anser vi är att det är en vida känd och accepterad teori, som många känner till. Den har också använts i tidigare uppsatser där man på olika sätt undersökt användbarhet. Bland annat ”MDI i mobiltelefoner, en komparativ studie av tre gränssnitt” (Edling Andersson, 2004), ”Intallationsgränssnitt under Linux och Windows XP utifrån ISO 9241” (Nilsson &

Nilsson, 2004) och ”Ett IT-verktygs användbarhet inom hemtjänsten – en utvärdering av IMIS” (Alpadie & Karlsson, 2002). Vi har inte funnit några tidigare undersökningar som undersökt hur Scrum uppfyller ISO 9241s kriterier för användbarhet.

## 1.2 Problemställning

I takt med att allt fler ställer större krav på systemen vi använder, krävs bättre och mer användarvänliga system för kunderna. För att uppnå dessa användarvänliga system tror vi att det behövs bättre förståelse för hur vi kan använda de systemutvecklingsmetoder som finns på ett optimalt sätt. Vi tror att genom god användbarhet kan man vinna mycket mark i en kundfokuserad miljö. Den senaste tiden har agila systemutvecklingsmetoder blivit allt mer intressanta och framförallt Scrum som idag används i allt större utsträckning. Frågorna vi ska försöka ge svar på i vår uppsats är:

- Hur väl uppfyller Scrum kriterierna för användbarhet som presenteras i ISO 9241?
- Hur kan man gå tillväga för att införa användbarhet enligt ISO 9241 i processen?

## 1.3 Syfte

Vi vill undersöka möjligheterna att få fram ett system som är anpassat efter slutanvändarnas önskemål då det gäller användbarhet, genom att använda Scrum som systemutvecklingsmetod. Samtidigt vill vi se om det är möjligt att skapa ett system som uppfyller kriterierna som ställs i ISO 9241 standarden. Finns de nödvändiga processerna redan i metoden eller behöver vi förändra de processer som finns för att detta ska vara möjligt?

## 1.4 Avgränsning

Vi kommer att avgränsa vårt arbete till att endast innefatta användbarhetskriterierna som finns i delar av ISO 9241 standarden. Alla delar är inte relevanta för vår undersökning utan främst kommer vi att använda oss av kapitel 11 - *Guidance on usability*. Målsättningen är att jämföra processerna i metoden mot rekommendationerna som ISO 9241 ger.

Det finns givetvis många olika systemutvecklingsmetoder där man skulle kunna undersöka hur ISO 9241 kriterier uppfylls. Vi har avgränsat oss till att endast inrikta oss på Scrummetoden. Det är en agil metod som används allt mer just nu.

Vi kommer att utföra intervjuer med personer som har erfarenhet av arbete med scrum. Eftersom det är processerna i metoden vi vill undersöka tror vi att de kan ge bäst insikt. Hade vi haft mer tid hade det varit intressant att följa ett eller flera projekt från början till slut. Men det är inte möjligt i detta fall.

## 1.5 Tillvägagångssätt

För att skapa en bakgrundsförståelse av vad man menar med användbarhet har vi valt att göra en litteraturstudie över vad begreppet ”användbarhet” innebär. Vidare avser vi att skaffa oss en generell överblick över ISO 9241 kriterierna. Vi kommer även att teoretiskt undersöka hur Scrummetoden är uppbyggd och hur processerna inom Scrum fungerar. Tanken är att vi ska ha en god bakgrundsförståelse innan vi går ut och gör den empiriska undersökningen. Genom bredare förkunskaper ökar möjligheterna till att hålla öppna frågor och att föra en djupare diskussion med intervjuobjekten (Jacobsen, 2002).

För att komma fram till svar inom vårt valda område, vill vi göra kvalitativa intervjuer med få aktiva systemutvecklare som använder sig av Scrummetoden i sina processer. Genom intervjuerna kan vi systematiskt försöka ta reda på hur systemutvecklarna ser på användbarhet i systemen de skapar samt hur de går tillväga för att säkerställa god användbarhet åt slutanvändaren.

Den empiriska undersökningen vi gör ute hos systemutvecklaren kommer att jämföras med den teori vi finner inom Scrum och ISO 9241. Utifrån denna jämförelse kan vi undersöka vilka processer som är inriktade mot en bättre användbarhet, samt hur man eventuellt skulle kunna förbättra dessa processer i Scrummetoden.



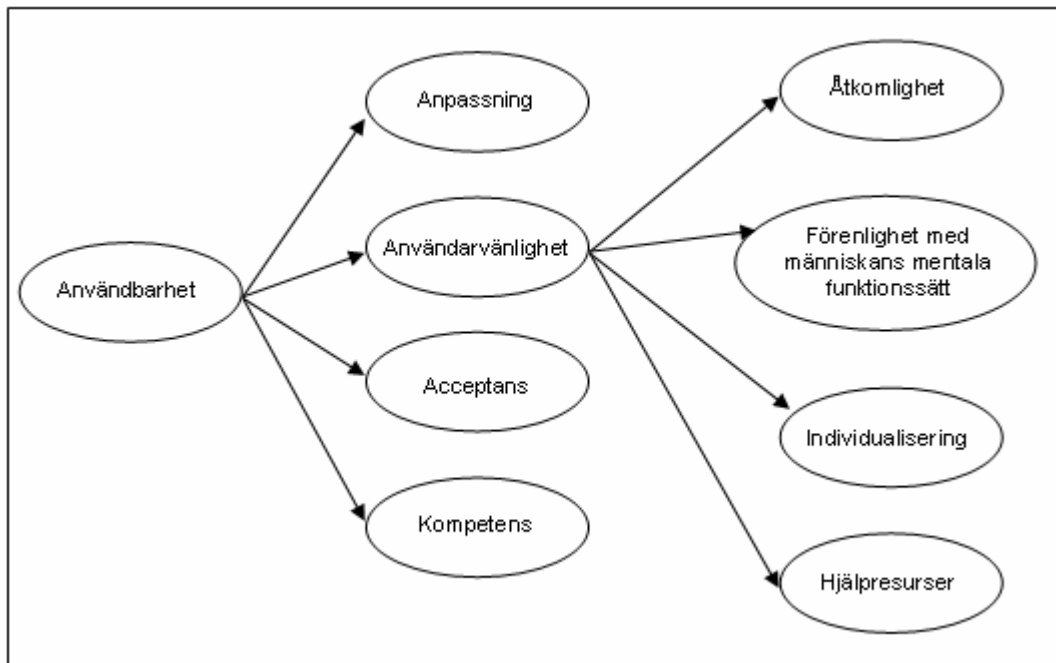
## 2 Litteraturgenomgång

*I detta kapitel presenterar vi tidigare forskning och principer för användbarhet, ISO 9241 och utvecklingsmetoden Scrum. Dessa utgör grunden för vår uppsats.*

---

### 2.1 Användbarhet

Varför är användbarhet viktigt? Syftet med att utföra arbete med hjälp av datorer är oftast att höja kvaliteten och/eller effektiviteten på arbetet. Det vill säga att öka produktiviteten. För att uppnå det krävs inte bara tillräcklig funktionalitet utan också god användbarhet som gör att arbetsuppgifterna kan genomföras effektivt. Minsta möjliga tid ska gå till att fundera ut hur programmet fungerar. Systemet ska vara lätt att förstå och ge tillräcklig information för att kunna användas effektivt. Det är också viktigt att användarna känner tillfredsställelse av systemet och uppskattar att använda det. (Allwood, 1998) I de flesta företag ligger vinsten i att öka produktiviteten hos de anställda. I exempelvis e-handelsföretag kan god användbarhet däremot betyda skillnaden mellan att den potentiella kunden lämnar sidan eller registrerar ett köp (Kane, 2003). Att åstadkomma ett användbart system är komplext eftersom det inte finns någon allmän enighet bakom begreppen och oftast ingen klar definition i beställningen av ett användarvänligt system (Gulliksen & Göransson, 2002). För att säkerställa att systemen som utvecklas är användbara krävs processer under systemutvecklingen som kartlägger användarnas motivation och arbetsmiljö (Lee & McCrickard, 2007).



Figur 2.1 Användbarhetsaspekter (Allwood, 1998)

Enligt Allwood (1998) kan användbarhet ses som en kombination av flera aspekter. Anpassning, användarvänlighet, acceptans och kompetens.

*Anpassning* innebär att systemet och dess funktioner är anpassade till användaren. Funktionerna ska följa uppgiftens struktur på ett logiskt sätt. Det kan vara lätt att uppnå i system där arbetsuppgifterna är tydligt strukturerade, men det är inte alltid självklart. Vissa program kan användas för många olika uppgifter som inte alltid är klart specificerade i förväg. (Allwood, 1998)

*Användaracceptans* ses av många som den viktigaste aspekten av användbarhet. Det innebär att användarna har motivation att använda systemet och är välvilligt inställda till det. Om de inte är det så är risken stor att användarna inte använder systemet eller inte lär sig det ordentligt utan använder det på ett sätt som tar onödigt lång tid eller genererar fel. God användaracceptans måste säkerställas redan i utvecklings- och implementeringsfasen. (Allwood, 1998)

Ytterligare en aspekt av användbarhet är *användarkompetens*. Det betyder att användarna ska ha fått tillräckligt med utbildning för att ha kompetensen som krävs för att använda systemet och utföra arbetet. Beroende på typen av program behöver användarna olika mycket utbildning. Utifrån graden av komplexitet utformas utbildningsmaterialet och utbildningen. Hänsyn måste också tas till användarnas förkunskaper. Datorvana, motivation och inlärningssätt påverkar förmågan att ta till sig utbildningen. (Allwood, 1998)

*Användarvänlighet* är ännu en del av användbarheten. Användarvänligheten består i sin tur av flera olika delar. Den ena är åtkomlighet. Åtkomlighet handlar om användarens tillgång till systemet och hur lätt det är att komma åt de funktioner som användaren vill använda. Det kan också handla om att ha tillgång till en viktig server eller stordator och att svarstiderna inte är för långa. (Allwood, 1998)

Systemet ska också vara kompatibelt med sättet som användarna tänker kring en arbetsuppgift. Användaren ska inte behöva klura ut hur systemet vill att saker ska göras, utan systemet ska stödja användarens sätt att fungera mentalt och vara logiskt uppbyggt. För att leva upp till det, ska systemet inte kräva att användaren behöver hantera stora mängder information på en gång. Systemet ska ge stöd för minnet genom den information som visas på skärmen. (Allwood, 1998)

Individualisering är ytterligare en aspekt under begreppet användarvänlighet. Även om användare på många sätt tänker och fungerar liknande så finns det också mycket som skiljer sig mellan olika individer och deras sätt att använda IT-stöd. För att passa varje enskild användare så bra som möjligt behöver systemet kunna passa många olika typer av användare. Det kan exempelvis handla om möjligheten att välja språk eller att anpassa hur information presenteras. Detta kan dock gå till överdrift och leda till dålig interaktionsdesign. Det kan också försvåra användandet om flera individer delar på datorerna, och funktionerna presenteras olika beroende på hur de olika användarna ställt in dem. (Allwood, 1998)

Sista delen av användarvänlighet behandlar kvalitén på den hjälp som finns att tillgå. Det kan vara andra personer, dokumentation, hjälpavsnitt i systemet m.m. Utformningen av hjälpresurser är viktig för inläringen av systemet och hur det upplevs. Användare som snabbt och lätt kan få hjälp med sina problem får en positiv upplevelse av systemet. Hjälpen kan ta egna initiativ genom tex. tips som visas när programmet startar eller bara aktiveras på användarens initiativ som ett hjälpavsnitt. (Allwood, 1998)

Användbarhet är ett stort och komplext område. Gulliksen & Göransson (2002) har försökt sammanställa konkreta, mätbara termer för användbarhet utifrån Nielsen (1993), Gudin (1992) och Dix et al. (1998). De kom fram till att inspiration till mätbara mål för ett användbart system kan vara att det ska vara:

- Lätt att lära
- Effektivt att använda
- Lätt att komma ihåg
- Ge få fel
- Subjektivt tilltalande (Gulliksen & Göransson, 2002)

En metod för att uppnå god användbarhet är scenariobaserad design. Först utförs en grundlig behovsanalys. Efter det följer en fas med iterationer mellan utveckling och

utvärdering. Genom representationer av designen utformas scenarion som användare får använda för att lösa uppgifter. Det används i kombination med *claims* som bygger på designkunskap som ska fånga positiva och negativa egenskaper hos designelementen. Syftet är att underlätta utvecklandet av effektiva och organiserade gränssnitt som stödjer användarna i deras arbetsuppgifter. (Lee & McCrickard, 2007)

Utvärdering av användbarhet bedrivs gärna tillsammans med slutanvändarna. Några metoder är *walkthroughs* och kontrollerade studier i laboratorium. Att utforma, utföra och analysera utvärderingarna tar tid. Det gör att det är svårt att utföra under de korta tidsperioder, som till exempel sprintarna i Scrum utgör. (Lee & McCrickard, 2007)

## 2.2 ISO 9241-11

ISO 9241 - *Ergonomic requirements for office work with visual display terminals* är en vida känd standard där riktlinjer för många typer av frågor om arbetsmiljö och användbarhet samlats i sjutton kapitel. Vissa länder har även valt att lagstadga denna standard. Trots det, är det få som har kunskap om innehållet. Det beror på att standarden är dyr att köpa in och svåråtkomlig. Dessutom är den väldigt omfattande. (Travis, 2004)

I kapitel 11 - *Guidance on usability* behandlas användbarhet. ISO standarden förklarar fördelarna med att mäta prestation och tillfredsställelse. (ISO 9241) Standarden behandlar alla faser i utvecklingen: analys av möjligheter, att skapa användarupplevelse, bygga användningskontexten, följa upp användningen samt att förbättra systemet. Den ger principer och rekommendationer för hur användbarhet ska säkerställas. Syftet är att utforma system som är *ändamålsenliga, effektiva och tillfredsställande* för specifika personer som ska uppnå specifika mål. (Travis, 2004)

De tre nyckelorden är viktiga i standarden:

- Ändamålsenlighet (eng effectiveness) beskriver hur väl ett mål eller en uppgift är avklarat.
- Effektivitet (eng efficiency) är hur stor ansträngning som krävs för att uppnå målet. Bättre effektivitet innebär mindre ansträngning.
- Tillfredsställelse (eng satisfaction) innebär hur positiva känslor och hur stor tillfredsställelse som användaren upplever av att använda produkten. (ISO 9241)

Dessa kriterier kan jämföras med ovan nämnda kriterier för god användbarhet. Se tabell 2.1.

Tabell 2.1 Jämförelse mellan ovan nämnda kriterier för användbarhet och ISO 9241-11.  
(Gulliksen & Göransson, 2002)

ISO 9241-11	Användbarhet
Effektivitet	Lätt att lära Effektivt att använda
Ändamålsenlighet	Lätt att komma ihåg Ge få fel
Tillfredsställelse	Subjektivt tilltalande

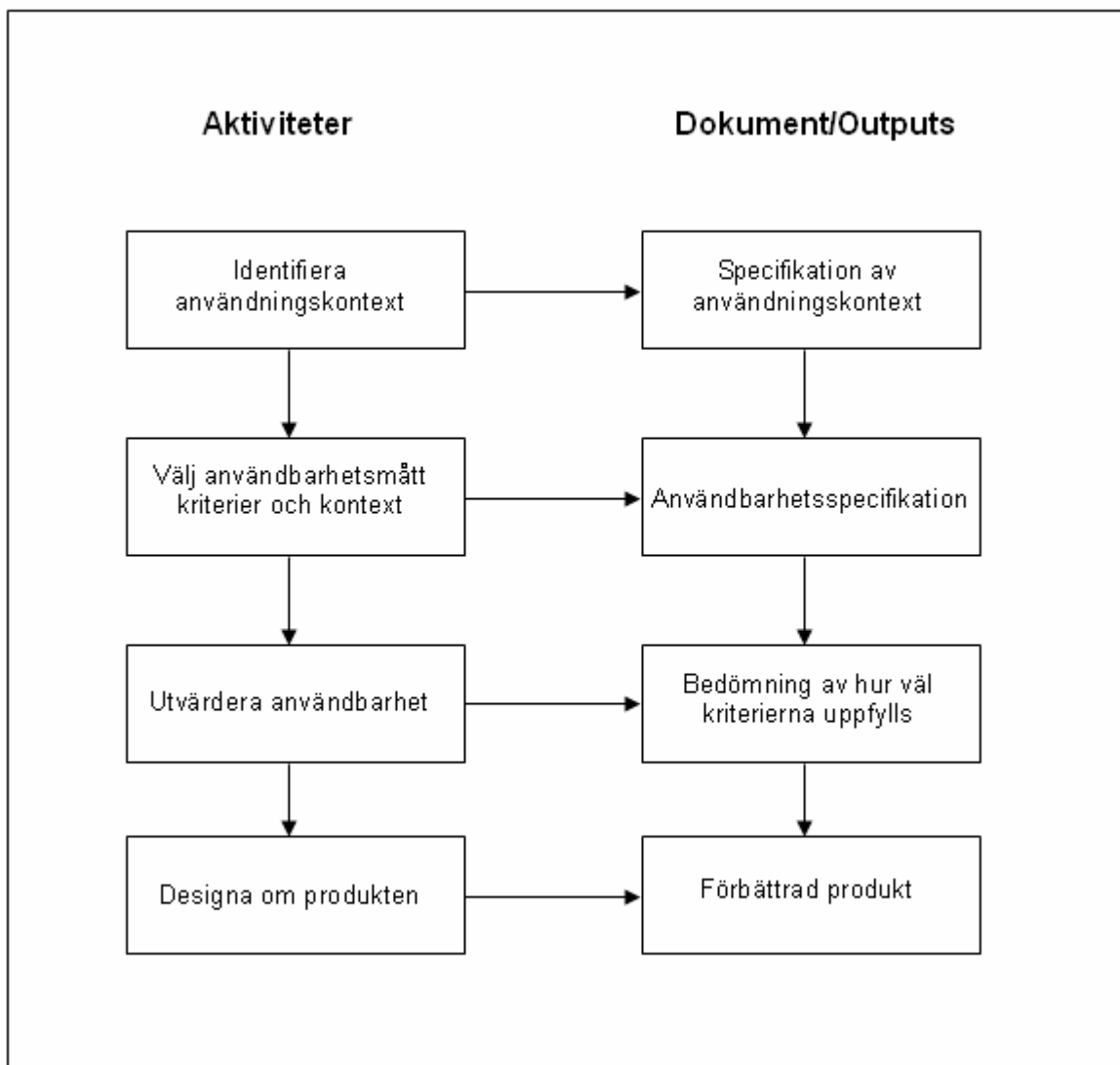
### 2.2.1 Designprocessen enligt ISO9241-11

Den del av standarden som är viktigast för vårt arbete är den som berör specifikation och utvärdering av användbarhet under designprocessen. Genom definitionen av användbarhet som presenteras i standarden kan utvecklingsteamet uppnå en gemensam förståelse för vad användbarhet är och vilka aspekter som är viktiga i det aktuella projektet. (ISO 9241)

I början av processen ska en specifikation göras av kontexten i vilken systemet kommer att användas. I kontexten ingår vilken typ av användare som ska använda systemet, till vilka uppgifter och mål som det ska användas och i vilken miljö uppgifterna utförs. Den analysen ligger sedan till grund för mer detaljerade användbarhetspecifikationer. (ISO 9241) Se exempel på attribut som kan vara relevanta vid kontextanalys i bilaga 1.

Innan utvecklingen påbörjas bör utvecklarna identifiera vilka användbarhetskrav som ska mötas av systemet. Vilka kritiska delar av användbarheten som ska mätas, vilka mätpunkter som ska användas för att mäta ändamålsenlighet, effektivitet och tillfredsställelse samt vilka värden som är godkända för dessa kriterier. ISO 9241-11 kan användas som ett ramverk för att ta fram mätbara kriterier för att säkerställa användbarhet i systemet. Dessa kriterier kan sedan kontrolleras i olika stadier av utvecklingen. Genom att jämföra resultaten med de önskade kraven kan utvecklarna fatta beslut om designen behöver förändras. Med den grundläggande analysen av utvecklingskontexten som stöd kan nyckelanvändare väljas för användartester. Även de uppgifter och den miljö som testerna ska ha kan utvecklas utifrån den. (ISO 9241)

Processen visas tydligare i figur 2.2 nedan.



Figur 2.2. Modell över användbarhetssäkringsprocessen enligt ISO 9241.

### 2.2.2 Kriterier och mätbara värden

Utifrån analysen av användningskontexten ska mätbara värden väljas som är relevanta för användbarheten i den specifika kontexten. Mätpunkter ska väljas för vart och ett av användbarhetskriterierna ändamålsenlighet, effektivitet och tillfredsställelse. Kriterier för vad som räknas som godtagbara värden för varje mätpunkt ska också specificeras. Utifrån det kan utvecklarna testa användbarheten under projektets gång. Om systemet inte når upp till de satta kriterierna behöver designen utvecklas vidare. (ISO 9241-11)

Exempel på mätvärden kan vara:

### **Ändamålsenlighet**

- Procent av mål som uppnåtts
- Procent av användare som lyckats med en uppgift
- Genomsnittlig riktighet på genomförda uppgifter

### **Effektivitet**

- Tid för att genomföra en uppgift
- Antal uppgifter utförda på en viss tid
- Kostnad för att utföra en uppgift

### **Tillfredsställelse**

- En betygsskala för upplevd tillfredsställelse
- Hur ofta användarna använder det
- Hur ofta användarna klagar (ISO 9241-11)

Utifrån den aktuella verksamheten sätts minimumkrav och målkrav för de valda mätvärdena. Det kan handla om procent av användare som uppnår ett visst mål eller genomsnittliga värden för de användare som använt systemet eller liknande.

## **2.3 Scrum**

Agila utvecklingsmetoder ger ett nytt sätt att leda och planera systemutvecklingsprojekt (Moe, et.al. 2008). Agile kommer från engelskan som betyder smidig eller vrig. Gustavsson (2007) anser att en av de största fördelarna med agila metoder är möjligheten att kunna slutföra ett projekt. Det är något som har visat sig svårt i många andra sammanhang. Agila metoder lyckas med denna bedrift då utvecklarna arbetar för att löpande leverera mindre delmål. (Gustavsson, 2007) Genom att med agila metoder planera och leverera små delar av projektet, ges möjlighet att uppleva en del av slutresultatet i ett tidigt skede. Därefter kan mer specifika krav ställas på resten av projektet och det finns möjlighet att stanna upp och utvärdera, och om det skulle behövas, finns möjlighet att ändra projektet under vägen (Gustavsson, 2007). Några av de största aktörerna inom agila metoder är Scrum, XP (Extreme Programming) och DSDM (Dynamic Systems Development Method). Vi kommer dock att avgränsa oss och fokusera vår teori på Scrum i denna uppsats. Den grundläggande filosofin bakom de agila metoderna är att arbeta i korta cykler där färdig funktionalitet levereras i slutet av varje cykel. Utvecklarna ska enbart fokusera på det mest nödvändiga, prioritering är A och O. Problemet ska beskrivas kortfattat och ingen onödig tid ska läggas på analys och design. Istället ska det finnas tät kommunikation mellan utvecklare och användare och tester ska ske löpande, delvis för varje enskild del och delvis för hela systemet. De traditionella metodernas dokumentation byts ut mot personlig kommunikation och ett minimum av dokumentation. (Constantine, 2001)

Scrum består av processer som underlättar systemutvecklarens arbete. Ett Scrum projekt är uppbyggt av tre roller, en produktägare, en Scrummaster och så själva arbetsgruppen. Schwaber (2002) ger följande förklaringar på de olika rollerna:

- Produktägaren är den person som ska sammanställa de förändringar som produkten som utvecklas ska genomgå. Produktägaren måste även prioritera bland de möjliga funktionaliteterna som följer med den färdiga produkten.
- Scrum-master ska coacha hela utvecklingsteamet, för att upprätthålla effektivitet och se till att projektet driver framåt mot de uppsatta målen.
- Arbetsgruppen består av en användare samt de utvecklare som arbetar i projektet. Tillsammans är de 5-9 personer.

Arbetsgruppen för en diskussion med produktägaren om hur målen ska sättas och hur sprintarna ska utformas. Vidare planerar de tillsammans hur funktionaliteten på produkten ska brytas ner på detaljnivå och delegeras till arbetsgruppen som arbetsuppgift. Hela gruppen är självorganiserande och medlemmarna har ett gemensamt ansvar för att resultat ska uppnås.

I början av ett projekt samlar produktägaren in alla krav och önskemål som ska ligga till grund för produkten som ska utvecklas (Schwaber, 2002). När man till slut lyckats definiera vilka mål som föreligger, styckas dessa ner i mindre delar. Schwaber (2002) poängterar vikten av att varje delmål både ska skapa ett affärsvärde, och kunna levereras som delmål.

Produktägaren prioriterar och bestämmer i vilken ordning som förändringar och funktioner ska genomföras och levereras (Singh, 2008). Schwaber (2002) lyfter fram att detta resulterar i en att-göra-lista. Då man startar en ny sprint låser produktägaren de första punkterna på listan och kallar Scrumteamet till möte.

Då alla är överrens om vilka uppgifter som ska utföras och vilken tidsåtgång man ska utgå ifrån, släpper produktägaren över ansvaret till Scrumteamet. De jobbar nu under eget ansvar och strävar efter att arbetet ska vara självorganiserande (Schwaber, 2002). En sprintfas består i huvudsak av 30 kalenderdagar, varav den första brukar avsättas för att skapa en sprint backlog. I figur 2.3 visas en modell av Scrumprocessen. För att avsluta en sprint måste man enligt Schwaber (2002) visa upp en fungerande programvara för en större grupp. Det är viktigt att produktägaren, användaren och olika representanter för företagsledningen är närvarande för att kunna göra en så exakt utvärdering som möjligt (Schwaber, 2002).

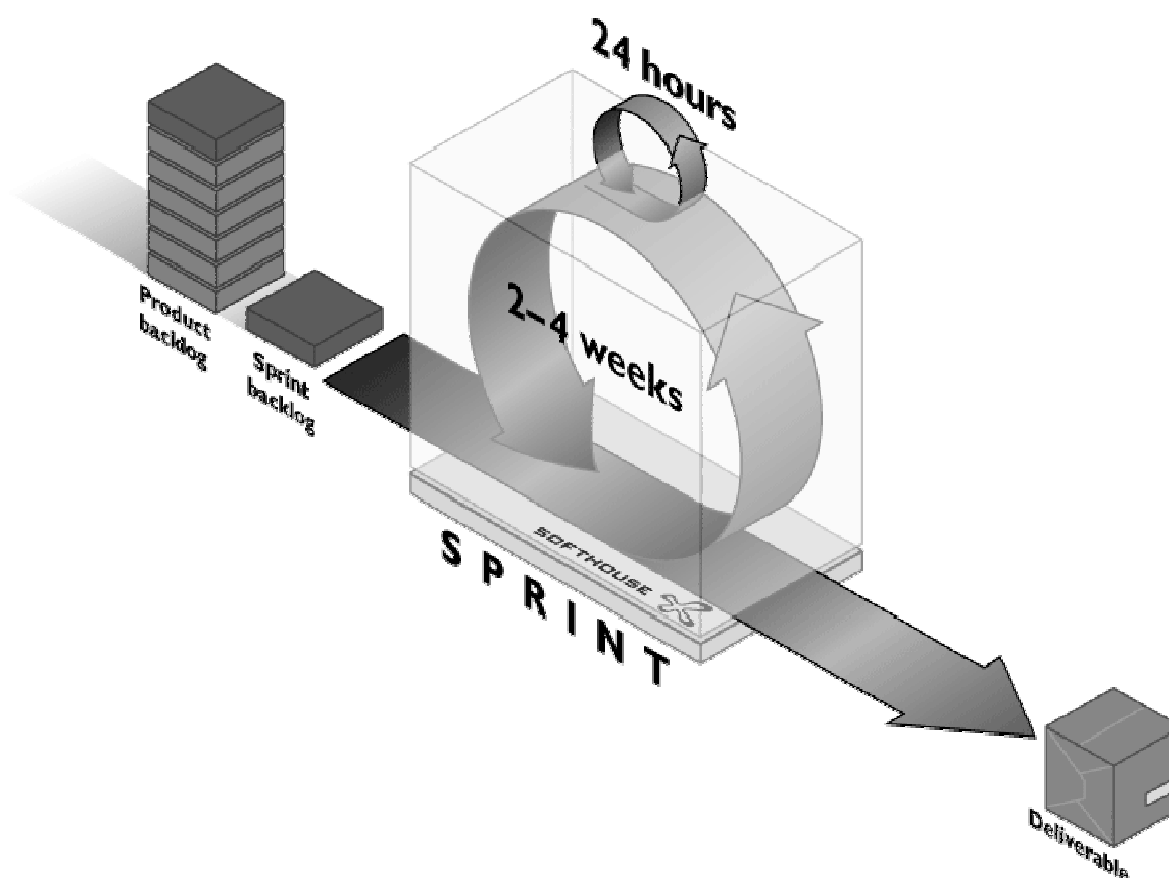


Dagliga möten hålls mellan Scrummastern och arbetsteamet. Avsikten är att försöka eliminera alla hinder som gruppen upplever (Schwaber, 2002). För att uppnå detta ska alla deltagarna i gruppen försöka svara på följande frågor av liknande karaktär:

- Vad har du gjort sedan förra mötet?
- Vad tänker du göra inför kommande möte?
- Vilka moment anser du hindrar dig att utföra det planerade arbetet?

Målet med frågorna är att försöka ge deltagarna full insyn i hur hela projektet fortskrider, samt vara till underlag för eventuell problemlösning.

Fullt funktionella och korrekt utvecklade system kan tas fram, men det finns en risk att dessa system kan vara svåra att hantera och använda. Lee och McCrickard (2007) skriver att då man inför användbarhetstänkande i processen, får utvecklarna en djupare förståelse av hur användarna arbetar, och hur de kan realisera målen med användningen av systemet på ett effektivt sätt. Agila utövare har börjat utforska olika sätt att införa användbarhet i de agila metoderna (Lee, McCrickard, 2007).



Figur 2.3 Modell av Scrum. (<http://www.softhouse.se/Uploades/scrum-diagram.gif>, 2009-05-10)

## 2.4 Koppling mellan Scrum och användbarhet

*"It is not a weak point, it is an absence"*

Så beskriver Alistair Cockburn bristen på användbarhetsdesign i de agila metoderna för Larry Constantine. (Constantine, 2001) Det är måhända en överdrift, men problemet diskuteras i flera artiklar. Det finns vissa likheter mellan det agila arbetssättet som används i Scrum och praktik för god användbarhet. De är båda cykliska i sitt arbetssätt, har stort fokus på kommunikation och koordination inom teamet och de är även människoorienterade. Trots det är de svåra att kombinera. Det agila arbetssättet bygger på att små delar i taget utvecklas i en iterativ process. Det gör att det slutgiltiga systemet är svårt att överblicka. Vilket i sin tur gör det svårt att planera gränssnittet och säkerställa att systemet blir användbart. (Lee & McCrickard, 2007)

Seffah et al. (2005) menar också att en nackdel med agila metoder är just svårigheten att säkerställa god användbarhet. De hänvisar till Armitage (2004) som skriver att de inre strukturerna i ett system kan ändras senare för att bilda en helhet. Men när det kommer till användbarhet krävs en bättre överblick från början. Att i efterhand gå in och ändra gränssnitt innebär att ändra förutsättningarna för användare som redan har lärt sig systemet. Även små ändringar kan ställa till med problem för användarna. Därför måste designen finnas på plats innan programmeringen kan påbörjas enligt Constantine (2001). Det finns också fördelar med agila metoder, de är snabba på att följa med i förändringar. Ett system som inte följer med i organisationens förändrade behov är inte heller användbart. (Seffah et al. 2005)

Det finns olika sätt att försöka komma till rätta med användbarhetsproblemet. Ett sätt är att kombinera ett användbarhetstänk med det agila arbetssättet. Det innebär att teamet först gör gränssnittsdesign och sedan fortsätter arbetet med klassisk Scrum metod. Det medför dock att arbetet blir styrt av gränssnittet. Det strider också mot en av grundtankarna med Scrum, att snabbt komma igång med projektet och att snabbt kunna leverera den första delen av systemet. (Lee & McCrickard, 2007) En annan modell är att använda Scrum parallellt med användbarhetsutveckling. Det ställer stora krav på kommunikation inom projektet och det finns risk för att Scrumteamet och användbarhetsteamet har olika visioner för systemet. (Lee & McCrickard, 2007)

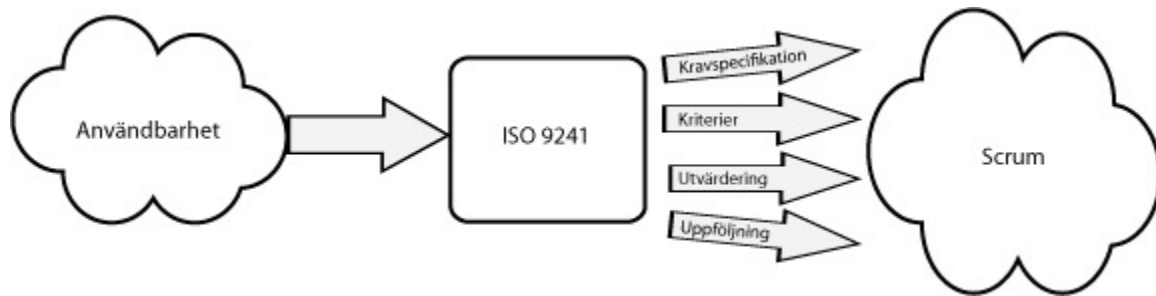
Singh skriver i sin artikel från 2008 att vissa organisationer infört processer i utvecklingen där några personer ägnar sig åt användbarhetsutveckling i ett par sprintar i förväg. På så sätt har man redan en bild av hur användbarheten skall vara utformad när man börjar med utvecklingen av funktionaliteten. Den modellen har Singh utvecklat till en metod hon kallar U-Scrum. Den innebär bland annat att det ska finnas en extra produktägare med fokus på användbarhet och dessutom någon expert på användbarhet i Scrumteamet. Aston & Meszaros (2006) förespråkar en iterativ prototypingmetod som de testat med lyckat resultat i ett projekt. Constantine (2001) beskriver en metod med indexeringskort för att främja användbarhet. Kort sagt, det finns många förslag på metoder för att införa användbarhetsprocesser i utvecklingsarbetet.

Det är dock vanligt förekommande att många utvecklare bedömer att kostnaden för att utveckla ett användarvänligt system är högre än att snabbt få fram ett system. En intressant observation som framkommer är att användbarheten som systemutvecklaren på egen hand skapar egentligen inte är mycket lättare att processera än om han skulle välja att utveckla efter slutanvändarens önskemål om användbarhet (Singh, 2008). Singh tror att största anledningen att systemutvecklaren bedömer kostnaden att bli högre för ett användarvänligt system, är att systemutvecklare saknar erfarenhet och kunskap om hur man ska utveckla och implementera ett system ur ett användarvänligt perspektiv. Utvecklarna väljer därför att gå på vad de tror är säkrare metoder för att framställa en fungerande system. Hennes artikel jämför två olika modeller av Scrum ifråga om användbarhet, U-Scrum och Scrum. Undersökningen har skett genom observation av två grupper, där den ena är starkt fokuserad mot traditionell systemutveckling med relevanta funktioner. Den andra gruppen har fokuserat mer på att utveckla ett användarvänligt system. Det visade sig att gruppen som var användarvänligt inriktad hade mer social kontakt med slutanvändaren. En undersökning som Singh (2008) utförde under arbetet med sin artikel, visade att av ett dussin produktägare hade inte en enda av dem relevant träning eller tillräcklig erfarenhet om processer som riktar sig mot en användarvänlig design eller produkt.

## 2.5 Ramverk

Utifrån kriterierna för användbarhet vill vi undersöka hur väl Scrum som systemutvecklingsmetod lever upp till kraven på användbarhet. Användbarhet är ett relativt luddigt begrepp, med många olika delar som ofta bedöms subjektivt av individen som använder systemet. Att försöka komma på egna jämförelsepunkter mellan användbarhet och Scrum hade varit både svårt, tidskrävande och svårt för oss att motivera. Genom att använda oss av ISO 9241 som ett ramverk för användbarhet så får vi en etablerad struktur att jobba efter.

ISO 9241 ger konkreta råd på hur god användbarhet ska kunna säkras. Genom att standarden har specifika kriterier för hur kravspecifikationer, kriterier, utvärdering och uppföljning bör se ut har vi kunnat identifiera mätpunkter för att jämföra hur arbetet i verkliga scrum-projekt ser ut jämfört med hur det borde se ut enligt ISO 9241. ISO 9241 är också en vedertagen standard vilket gör att de kriterier som nämns där bör vara relevanta för användbarheten. Det är alltså i praktiken ISO 9241 som utgör vårt ramverk. De aktiviteter för utvecklingsprocessen som rekommenderas av standarden, och som vi kommer att utgå ifrån är främst kravspecifikation, kriterier, utvärdering och uppföljning enligt modellen som också presenteras i figur 2.2.



*Figur 2.4 Modell av vårt ramverk.*

Figur 2.4 visar vårt ramverk. Användbarhet illustreras av ett moln för att tydliggöra att det är stort och komplext. Genom att använda ISO 9241 översätts det luddiga användbarhetsbegreppet till något mer konkret. Här finns tydliga ramar för hur processer bör utformas för att säkerställa god användbarhet. Genom att ta de kriterier som är intressanta för vårt arbete kan vi jämföra dem med processerna i Scrum. På så vis hoppas vi kunna besvara våra frågeställningar.

## 3 Metod

*I detta kapitel beskriver hur vi har gått tillväga i vår process att skapa denna uppsats. Vi förklarar hur vi utfört vår empiriska undersökning samt på vilket sätt vi gjort vårt urval av informanter. I kapitlet behandlar vi också giltigheten i uppsatsen samt dess trovärdighet.*

---

### 3.1 Metodval

Många hävdar att en *kvantitativ* undersökning endast mäter undersökarens förståelse för en viss företeelse. Denna åsikt baseras på att undersökaren själv definierar frågor och svarsalternativ. (Jacobsen, 2002) För att undvika denna fallgrop var vår ansats istället att utföra en *kvalitativ* undersökning. Jacobsen (2002) beskriver den kvalitativa metoden som att man utför observationer där man kan se vad forskningsobjekten gör, samt att man låter dem tala med egna ord. Vi har lagt tonvikten vid en kvalitativ undersökning, där vi utfört vårt fältarbete genom att använda oss av intervjuer med öppna frågor som inbjuder till diskussion. Det gav oss möjlighet att vara flexibla och anpassa vår datainsamling till den verklighet som råder vid visst tillfälle.

Vid frågor kring användbarhet är tolkningar ofta subjektiva och individuella. I sådana fall anser Jacobsen (2002) att kvalitativa undersökningar är att föredra. De ger en större flexibilitet samtidigt som de ger en bättre uppfattning av hur andra uppfattar verkligheten. Det kan endast uppnås genom intensiva och långvariga intervjuer. Kvalitativa metoder understryker ofta att närheten är ett viktigt element då undersökaren vill förstå andra människors uppfattning av verkligheten (Jacobsen, 2002). Nackdelen med att ha en för stark närhet kan vara att undersökaren mister sin förmåga att vara kritisk. Undersökaren förlorar därmed sin förmåga att vara analytisk och hålla en objektiv distans till det som undersöks (Jacobsen, 2002). ). Jacobsen (2002) hänvisar till Repstad (1993) som skriver att man istället för att välja en inriktning ska försöka växla mellan de båda idealen. Det är viktigt att bibehålla närheten för att lättare förstå det som ska undersökas, men samtidigt måste man hålla distansen för att använda förståelsen på ett bredare perspektiv. Vi försökte därför växla mellan dessa båda ideal då vi utförde den empiriska datainsamlingen.

Det finns inga krav på att tidigare kunskaper ska finnas för att man ska kunna formulera en frågeställning, till skillnad mot då en hypotes ska formuleras (Backman, 1998). En hypotes bygger istället på någon form av påstående. Vi har inte funnit någon tidigare forskning kring det vi valt att undersöka, vilket Backman (1998) anser vara ett kriterium för att en

frågeställning ska kunna tas fram. Backman (1998) förklarar att en formulering av forskningsfråga kan ske då ett visst forskningsområde är okänt, eller vid avsaknad av specifik kunskap inom området.

Avgränsningar kan göras explicit eller implicit. Forskaren bör så långt som möjligt göra explicita val, då de bygger på medvetna avgränsningar (Jacobsen, 2002). För att göra en så bra avgränsning som möjligt behöver också det som inte avses undersökas nämnas samt varför detta har valts bort. För att på bästa sätt göra det här anser Jacobsen (2002) att problemställningen måste analyseras grundligt. Inte förrän vi är medvetna om vad vi vill undersöka är vi klara över vad vi väljer att inte undersöka. Utifrån det har vi försökt avgränsa vårt arbete. Jacobsen (2002) skriver att den första fasen i ett arbete är att formulera en problemställning och att man måste anpassa undersökningens upplägg till problemställningen. Vår forskningsfråga kräver att vi förstår hur man använder Scrum i praktiken samt vad användbarhet går ut på.

### 3.2 Intervju

En öppen och individuell intervju kännetecknas enligt Jacobsen (2002) av att samtalet sker som i en vanlig dialog. Den data som fångas upp av intervjuaren kommer i form av berättelser. En stark fördel med öppna och individuella intervjuer som Jacobsen (2002) lyfter fram är att det ger en möjlighet att höra individens personliga tolkning av ett visst fenomen. Det ger klarhet i hur individen tänker och uppfattar de förhållanden som råder. Jacobsen (2002) lyfter fram olika tillfällen då öppna intervjuer är mest lämpliga:

- När få enheter ska undersökas
- Vid intresse av vad den enskilda individen har att säga
- När individens tolkning vid ett speciellt fenomen är intressant

Risken med öppna intervjuer är att det resulterar i för mycket information, så att det blir svårt att överblicka den. Jacobsen (2002) poängterar också att intervjuer ofta successivt ger mindre information för varje ny intervju som utförs. Till slut uppnås en mättnadspunkt då en ny intervju inte ger någon ny information. Av den anledningen och på grund av vår begränsade tid var vår avsikt att hålla oss till mellan 4-5 intervjuobjekt. Tyvärr lyckades vi inte genomföra mer än tre.

Vårt mål var att i huvudsak utföra individuella, öppna intervjuer och att använda oss av öppna frågor, men i vissa fall har vi stängda frågor för att vi bara behöver kort bakgrundsinformation. Istället har våra intervjuer gått mot en mera semistrukturerad teknik, vilket innebär att intervjuobjektet får möjlighet att själv utveckla sina svar (Kvale, 1997). Vi har därför öppna frågor som bas för att främja en bredare diskussion. Vi valde att utföra våra intervjuer ansikte mot ansikte, vilket är den vanligaste metoden enligt Jacobsen (2002). Då vi utfört intervjuerna har vi gjort bandupptagningar och samtidigt fört anteckningar.

Vi har delat upp våra frågor i fyra grupper. Enligt Patel och Tebelius (1987) bör man utforma en intervju som en tratt med allmänna breda frågor först för att sedan gå ner på djupare frågor. Det har vi tagit fasta på. Därför har vi först bakgrundsfrågor som ska ge en bild av personen vi intervjuar och dess miljö. Sedan har vi frågor om hur arbetsprocessen fungerar ur intervjuobjektets synvinkel och ett användbarhetsperspektiv. Den tredje frågegruppen handlar om mer specifika processer som är baserade på ISO 9241. Slutligen har vi några avslutande frågor för att fylla ut eventuella luckor i intervjun. I tabellen nedan presenteras våra intervjufrågor. Vi har här utelämnat de inledande och avslutande frågorna. Fullständig intervjuguide finns i bilaga 2.

Tabell 3.1 Tabell över våra intervjufrågor och deras koppling till frågeställningen.

Tema	Fråga
Hur arbetsprocessen ser ut ur ett användbarhetsperspektiv.	<ul style="list-style-type: none"><li>• Hur ser en utvecklingsprocess ut hos er?</li><li>• Vad betyder god användbarhet för dig?</li><li>• Vilka processer använder ni för att uppnå god användbarhet?</li><li>• Vem är det som bestämmer vad som är god användbarhet i era projekt?</li><li>• Har ni någon person som är ansvarig för användbarhet i teamet? (utbildad?)</li><li>• Vilka verktyg använder ni för att uppnå god användbarhet?</li><li>• Hur följer ni upp systemen ni utvecklar? (användbarhet)</li></ul>
ISO 9241 relaterade aktiviteter.	<ul style="list-style-type: none"><li>• Känner du till ISO 9241?</li><li>• Använder ni er av ISO 9241?</li><li>• Hur ser era kravspecifikationer ut?</li><li>• Hur undersöker ni användningskontexten? (användare, miljö, uppgifter?)</li><li>• Identifierar ni mätbara användbarhetskriterier?</li><li>• Hur går ni tillväga för att testa användbarheten under utvecklingen? Utifrån vilka kriterier?</li></ul>

### 3.3 Urval av intervjuobjekt

I vårt val av intervjuobjekt var vår avsikt att finna personer som arbetar i projekt där man använder Scrum. Då en kvalitativ metod används med ett fåtal personer som intervjuobjekt kan det bli problem med representativiteten hos dem (Jacobsen, 2002). Vi var därför tvungna att hitta personer som kunde representera det område vi avsett att utforska. Det är vanligt förekommande att kvalitativa ansatser får problem med den externa giltigheten (Jacobsen, 2002).

Det är svårt att få fram ett representativt urval med för få intervjuobjekt, Jacobsen (2002) anser att det går en maxgräns vid 20 personer. Därefter blir informationen mer kostsam av tid och resurser än vad nyttan av intervjuerna blir. Ett första steg i urvalsprocessen var att försöka få en så bred överblick som möjligt över alla vi ville undersöka. Jacobsen (2002) skriver också att det är viktigt att bestämma antalet enheter vid en given problemställning. Vår målsättning var att hitta personer som jobbar i olika typer av projekt för att få en bredare bild av hur problem med användbarhet lösts vid arbete med Scrum. I gengäld förlorar vi en djupare inblick i de specifika fallen.

Trots ihärdigt arbete genom mailkonversation och telefonsamtal för att hitta intervjuobjekt till vår uppsats, var det endast tre personer som hade tid och möjlighet att ställa upp för en intervju med oss. De tre personerna vi intervjuat har ställt sig väldigt positiva till vår undersökning och bidragit med god och riklig fakta. Vi har även lyckats fånga intervjuobjekt i olika verksamhetsområden, som i sin tur ger oss en bredare bild av hur det verkligen fungerar. För att kompensera det låga antalet intervjuobjekt har vi försökt göra en något djupare dykning i det teoretiska materialet vid arbetet av vår analys.

### 3.4 Analys

Jacobsen (2002) menar att för att få så komplett och korrekt rådata som möjligt bör bandinspelning, eller om det behövs, videoinspelning användas vid intervjuer. Vi anser inte att videoinspelning ger något mervärde i vårt fall utan vi har använt oss av enbart ljudupptagning. Vi förde också anteckningar under intervjuerna. Jacobsen (2002) anser att transkribering är att föredra framför att bara lyssna på intervjuerna och titta i anteckningarna. Detta för att minska risken att missa delar och för att kunna göra anteckningar i utskriften. Vi valde att transkribera i stort sett hela intervjuerna. De inledande och avslutande artighetsfraserna är undantagna och även vissa sidospår där vi helt avvikit från ämnet, främst i slutet av intervjuerna. De ställen där vi utelämnat delar av intervjun är markerade i transkriberingen. Vi har även utelämnat hummanden, stakningar och andra oväsentliga ljud som vi upplever skulle ha försvårat läsningen av transkriberingen. I övrigt har vi försökt hålla det så exakt som möjligt för att ingen



information ska förvrängas eller gå förlorad. Utskrifterna från intervjuerna finns i bilaga 3-5.

För att få en klarare bild över informationen kopplat till vår frågeställning kategoriserade vi datan. Jacobsen (2002) beskriver kategorisering som att samla data i grupper. Syftet är att förenkla och tydliggöra rika data. Om det finns en intervjuguide eller liknande kan den användas för att skapa kategorier. (Jacobsen, 2002) Vi använde vår intervjuguide för att skapa kategorier och sedan gick vi igenom intervjuerna och placerade informationen under relevanta kategorier. Vi försökte skapa en kortfattad men tydlig bild av det som intervjuobjekten beskrivit. Först gjorde vi det för var och en av våra intervjuer och sedan försökte vi hitta likheter och olikheter mellan de olika situationerna.

### 3.5 Yttre påverkan vid intervju

Vid empirisk undersökning finns risken för att de resultat som framkommer påverkas av undersökningen i sig. Undersökningens upplägg kan ge effekter som skapar utfallet (Jacobsen, 2002). Jacobsen (2002) menar att forskaren måste ställa sig frågan hur mycket undersökningen påverkar situationen hos företaget och hur pass korrekt undersökningen verkligen blir. Det gäller att göra så liten påverkan som möjligt på miljön för att få en rättvis bild av det som ska undersökas. Jacobsen (2002) skriver att det starkt hävdas att forskarens personliga värdering alltid påverkar undersökningen på något sätt. Vi kan därmed inte hävda att vår undersökning är helt neutral, beroende på hur vi agerat vid intervjuerna.

Undersökningar kan delas in i två olika huvudtyper. Den första har som mål att beskriva dagens situation, medan den andra avser att mäta vilka verkningar och vilken effekt en viss åtgärd har (Jacobsen, 2002). Vår undersökning berör den första av dessa två typer då vi har undersökt vilka processer som används för att säkerställa användbarhet i ett system.

### 3.6 Källkritik

Källkritik handlar om att kontrollera fakta. Trovärdigheten i källan behöver bedömas för att vi ska kunna använda den som underlag. Teoretiskt sett är källkritik ganska enkel, men då man ska använda det i praktiken kan det uppstå svårigheter (Thurén, 2005). Thurén (2005) har skapat fyra tumregler att utgå ifrån vid arbete med olika källor.

- Äkthet
- Tidssamband
- Oberoende
- Tendensfrihet

Det är viktigt att källan är det som den utger sig för att vara och källan får inte vara för

gammal, så att den förlorar relevans. Det får alltså inte gå för lång tid mellan den undersökta händelsen till dess att källan har skapats. Vidare skriver Thurén (2005) att källan ska kunna stå för sig själv och inte vara ett referat av andra källor. Källan måste även vara objektivt framtagen och inte styrd av personliga intressen. Den måste ge en verklig bild av situationen eller miljön den beskriver (Thurén, 2005). Vi har haft dessa teorier som grund då vi valt våra källor. Vi anser att de källor vi valt är trovärdiga och relevanta då det handlar om erkända forskare som ligger till grund för litteraturen. Dock kan det alltid finnas källor som varit bättre för specifika ändamål men som vi inte har kommit i kontakt med.

Den data som samlas in genom intervjuer är viktiga då information som annars eventuellt inte finns i skriftliga källor kommer fram. Thurén (2005) anser ändå att skriftliga källor ska ha en hög prioritet, då undersökaren ofta baserar sina frågeställningar på dessa tidigare källor. Det krävs grundläggande kunskaper för att vidareutveckla ett antagande eller en teori.

### 3.7 Reliabilitet och validitet

För att få validitet i en undersökning krävs att andra uppfattar undersökningen som relevant (Jacobsen, 2002). Med andra ord ska arbetet vara generaliserbart, det vill säga att ett resultat inom ett specifikt område också ska vara giltigt i andra sammanhang. Jacobsen (2002) förklarar detta med extern giltighet och överförbarhet. Det betyder att resultaten från en undersökning kan överföras från ett sammanhang till ett annat. Det är svårt för oss att leva upp till då vi har en begränsad omfattning på vårt arbete. Vid en individuell intervju mäter man individuella och personliga synpunkter på ett visst fenomen. Det är viktigt för validiteten att man räknar alla synpunkter lika starkt (Jacobsen, 2002).

Samhällsvetenskapen talar om att använda intersubjektivitet istället för sanning. Det innebär att desto fler som är eniga om en viss sanning ju mer korrekt är den beskrivningen. (Jacobsen, 2002).

Vi vill även öka validiteten i vår uppsats genom att lyfta fram nackdelar med ISO 9241 som är relevanta för vår undersökning. ISO använder sig av mätbara värden, medan det egentligen finns fler aspekter att ta i beaktning då man undersöker användbarhet. Vidare avser vi att utföra våra intervjuer inom olika företag, för att öka bredden på den information och data vi ska samla in. Detta är också en metod vi valt för att öka validiteten i vår uppsats. Individuella intervjuer lyfter fram personliga erfarenheter från olika situationer och från olika tidpunkter. Jacobsen (2002) anser att detta är viktiga aspekter att utgå ifrån då man avser att öka validiteten i en undersökning.

För att pröva den interna giltigheten lyfter Jacobsen fram två åtgärder som kan användas:

- Kontrollera undersökning och resultat mot andra
- Kritiskt granska de egna resultaten

Då vi inte funnit någon tidigare forskning som behandlar vår frågeställning kan vi inte jämföra vår undersökning och dess resultat med andra undersökningar. Vi måste istället se på våra egna resultat ur en kritisk synvinkel.

Reliabilitet betyder att det ska gå att lita på att undersökningen är rätt. Jacobsen (2002) menar att det ska vara möjligt att utföra samma undersökning utifrån liknande förutsättningar men ändå få samma resultat. Beroende på vilken undersökningsmetod som valts påverkas reliabiliteten olika. I vårt fall då vi har utfört individuella intervjuer finns risken för intervjuareffekt. Vi måste vara medvetna om att vår närvaro i intervjun kan ha skapat speciella resultat. Vidare tar Jacobsen (2002) upp att det även kan uppstå kontextuella effekter, vilket betyder att platsen där vi utför intervjun kan påverka resultatet. För att eliminera dessa effekter valde vi att utföra våra intervjuer på ett så likartat sätt som möjligt. Vi har utfört två intervjuer på intervjuobjektens arbetsplats, det är en trygg och välbekant miljö för intervjuobjekten. Vi har då suttit så avskilt och ostört som möjligt. En intervju utfördes inom universitetets lokaler i ett grupprum efter förslag från intervjuobjektet. Han har tidigare varit student på universitet så vi anser att detta inte bör ha påverkat resultatet.

## 4 Empiri

*I det här kapitlet presenteras våra intervjuobjekt och den information som våra intervjuer resulterat i.*

---

### 4.1 Presentation av informanter

Klas Skogmar är Vice VD på Arkatay Consulting. De arbetar med att hyra ut projektledare till olika IT-projekt. Deras affärsidé bygger på att erbjuda tjänster inom projektledning, utbildning och verksamhetsutveckling, det innebär att de både åtar sig uppdrag i större och mindre företag. Klas berättar att hans arbetsuppgifter kan variera beroende på vilka uppdrag som är prioriterade, men han agerar bollplank till VDn på Arkatay Consulting vid strategiska frågor. Då vi träffar Klas berättar han bland annat om det senaste uppdraget han varit involverad i, vilket innebär att vara projektledare vid IKEA´s IT avdelning. Där jobbade han bland annat med att ta fram processer för Scrum. Arkatay Consulting drivs i nära samarbete med ett annat företag som inriktar sig på att hyra ut systemutvecklare. Klas anser därför att de har goda kunskaper inom såväl systemutvecklingsmetoder som projektledning i IT sammanhang.

Jonas Wallander jobbar som systemutvecklare på Tetra Pak Packaging Solutions vilka är en del av Tetra Pak koncernen. Hans arbetsuppgifter går ut på att ta fram informationssystem och applikationssystem som ligger nära produktionen. Jonas sitter i en grupp som har hand om Inline Quality Control. Tetra Pak har till uppgift att tillverka system för förpackning och distribution av flytande livsmedel. Systemen som Jonas tar fram ska användas av operatörerna som kör maskinerna ute i fabrikerna, det ska vara styrsystem som effektiviserar tillverkningsprocessen.

Michael Lindberg arbetar på Ninetech som är ett konsultföretag. Hans ansvarsområden är att agera Scrum-master, konsult och systemutvecklare. Michaels huvudsakliga arbetsuppgifter är inom programmering och styrning av olika projekt. Ninetech arbetar med interna projekt men även externa ut mot kund. De har som mål att leverera helhetslösningar inom IT där fokus ligger på affärsnytta. Vid tillfället då vi intervjuar Michael arbetar han tillsammans med övriga medarbetare på Ninetech på ett internt projekt, där de utvecklar den egna produkten, ett system för processhantering. Ninetech i Malmö består av en mindre avdelning på bara tre personer där alla medarbetarna har likvärdiga kunskaper och delar på arbetsuppgifterna.

## 4.2 Resultat

Klas Skogmar på Arkatay Consulting berättar om hur de arbetade med Scrum när han jobbade som konsult för IKEA. De använde Scrum i delar av arbetet, främst för Change Requests. De arbetade med IKEAs hemsida och de jobbade med att utveckla den befintliga hemsidan. Det handlar alltså inte om ett helt nytt system. Det har betydelse för hur man jobbar med användbarhet. Klas beskriver situationer där gränssnittet inte alls påverkas eller bara påverkas marginellt av de förändringar som görs. Men han ser också att det kan bli problem, exempelvis med att alla sidor inte ser likadana ut under tiden som utvecklingen pågår.

*”Därför har jag ofta inom Scrum fått höra, att när man väl är färdig, och säger att nu känns det bra. Då lägger man ofta till en iteration där man säger att nu förbättrar vi sakerna för att finslipa detaljer och bygga färdigt funktionen.” (Klas Skogmar, Bilaga 3, sid. 47)*

Projektet beställdes från andra organisationer inom IKEA. Beställningen gick sedan genom kommunikationsorganisationen till utvecklingsorganisationen. Kraven på systemet specificerades alltså utanför utvecklingsorganisationen. Eventuell förberedande analys av användningskontexten och formulering av kravspecifikationer utförs av kommunikationsorganisationen. Det är redan färdigt när Scrumprojektet påbörjas. Scrumtemat får en användbarhetsbeskrivning i textform och en visuell beskrivning av hur sidan ska se ut.

Eftersom behov och krav identifieras utanför scrum-teamen finns ingen direkt användbarhetsansvarig inom utvecklingen. Det är Business Analysts som arbetar med användbarheten. Inom utvecklingen försöker de använda befintliga verktyg för att göra bland annat beroendeträd och navigering. De mätetal de använder för att kontrollera utvecklingen är främst av teknisk karaktär såsom svarstider för servrar och browserkompatibilitet. De har viss testning av användbarheten, men den är sällan utförd av de verkliga användarna, utan det är beställarorganisationen som får representera användarna.

Utvärdering är också ett område som behöver förbättras på IKEA anser Klas. Han poängterar vikten av att använda statistik i utvärderingen och i samband med det nämns exempel på nyckeltal. Det kan vara hur lång tid som en person tillbringar på sidan, hur fort de kommer till en produktsida eller var de lämnar sidan. Han påpekar dock att det är viktigt att komplettera utvärderingen med kvalitativa studier där man ”tittar över axeln” på en person som använder hemsidan. Annars är det svårt att veta om en person lämnar sidan fort av irritation för att den inte hittar det som söks eller för att personen snabbt kommer till den sökta informationen och därefter lämnar sidan. Ur testningssynpunkt anser han att det finns fördelar med agila metoder när man jobbar med just webbtjänster.

*”Egentligen är webb en av de bättre bitarna att jobba med agila metoder. Man kan trycka ut något ganska snabbt till slutanvändarna. Och om de tycker att det är dåligt och klagar kan man lätt dra tillbaka det.” (Klas Skogmar, Bilaga 3, sid. 49)*

Klas berättar också att inom webb anses det lämpligt att använda så kallade mock-ups som prototyper för att testa användbarhet. Man kan navigera i dessa men det finns ingen kod bakom och inga riktiga värden. Det går att testa vissa saker på så sätt men inte allt. Exempelvis är sökfunktioner en viktig del av användarupplevelsen, de kan inte testas med mock-ups. De användbarhetsstudier de gör fokuserar på att de gör rätt saker, men inte på exempelvis kundnöjdhet. På IKEA utförde de ibland användartester, men det var inte alltid det fanns tid att rätta till eventuella brister som framkom. ISO 9241 känner Klas inte till. Antagligen är det inget de använder.

### ***Jonas Wallander – Tetra Pak***

Jonas Wallander jobbar med att utveckla ett informationssystem internt på Tetra Pak. Det handlar främst om automation för maskiner. Maskinerna behöver i allt större utsträckning kommunicera med andra system och därför behöver nätverk mellan dem utvecklas. Utvecklingen gäller ett nytt system som ska utvecklas från grunden och kommunicera med flera andra system och fungera med flera olika arbetssätt. I vanliga fall används en egenutvecklad, vattenfallsinspirerad, utvecklingsmetod. Men i det projekt som Jonas arbetar med kunde de inte få en klar överblick av problemet från början. Därför har de valt att arbeta agilt med Scrum. Det gick inte att byta för hela projektet utan det är bara mjukvaruutvecklingen som sker med Scrummetod. Det gör att det krävs en del speciallösningar för kommunikationen mellan de olika delarna i projektet.

*”Vi har en roll som är utanför Scrumteamet som är traditionell projektledare. Flera stycken faktiskt, för det är så stort att det är flera delprojekt. Vi har en traditionell projektledare, och så är jag produktägare i Scrum-mening då, och så en Scrum-master som sitter i teamet.” (Jonas Wallander, Bilaga 4, sid. 55)*

Utvecklarna arbetar helt enligt Scrum men Jonas som är produktägare får tillsammans med projektledaren översätta arbetet till en plan som visar de övriga intressenterna att de vet vad de gör.

Användbarhet i det här sammanhanget beskriver Jonas som något som kommer att användas och fungera. Dessutom får det inte störa operatörernas nuvarande arbetssätt. De ska göra samma sak fast med vissa tillägg när det nya systemet implementeras. Utmaningen ligger i att inte bli ytterligare ett system som operatörerna ska jobba med, utan smälta in med de existerande systemen. Men användbarhet mot operatörerna är ändå lågt prioriterat. De kommer att få utbildning på det nya systemet men det är inte deras mottagande och användning av systemet som avgör om det kan anses lyckat eller inte.

Projektet har ingen insatt beställare som de kan få feedback ifrån. Därför är kravspecifikationerna också luddiga och muntligt utfyllda. De olika fabriker jobbar inte på samma sätt med maskinerna vilket också ställer till problem. Därför har de en "superoperatör" som har mappat hur de olika fabriker arbetar. Superoperatören får titta på prototyper och processer och säga om något strider mot det nuvarande arbetssättet. Men innan systemet kommer i drift kan de bara göra simulationer för att testa funktionaliteten. Implementeringen kommer dock att ske stegvis med början i en pilotfabrik. Projektet har heller ingen användbarhetsexpert inblandad, det närmsta de kommer är att Jonas själv är systemvetare.

ISO 9241 känner inte heller Jonas till och de använder sig inte av standarden. I deras arbete är domänkunskap och sunt förnuft viktigast och de är väl insatta i hur det som de jobbar med fungerar. Sunt förnuft är ett uttryck som används av våra informanter, det innebär att de fattar beslut utifrån sina åsikter och sin förståelse för problemet. Trots det lever de på många punkter upp till kriterierna för ISO 9241. Användningskontexten undersöktes i början av projektet, visserligen mest i form av de övriga systemen som de ska jobba med. Men det är de viktigaste komponenterna i det här fallet. De har även tittat på pilotfabriken och arbetet där. Förutom superoperatören ska de också sätta ihop en referensgrupp av olika intressenter som ska få titta på systemet. De kommer in sent i projektet eftersom det måste finnas något för dem att titta på. De har dock inga mätetal för användbarhet, bara mot affärssidan. Det är de ekonomiska målen som är motivationen för att införa systemet.

### ***Michael Lindberg – Ninetech***

Michael arbetar i första hand med utveckling av Ninetechs egen produkt PrimeCase. Den har funnits i något år och de utvecklar den löpande. När systemet initialt utvecklades användes inte Scrummetod, det är något som har kommit in på senare tid. Nu försöker de i största möjliga utsträckning jobba i sprintar. Initiativ till utveckling av systemet kommer dels från dem själva men det kan också komma från deras kunder. De kan ha flera projekt som går parallellt men de försöker samla allt i sprintar. Men vissa saker måste ändå hållas utanför.

*"Om det är mycket integrationsarbete att jobba mot andra system och andra leverantörer så går det inte ibland. Det blir för mycket väntan fram och tillbaka." (Michael Lindberg, Bilaga 5, sid. 63)*

När det gäller användbarhet så anser Michael att det är att funktionerna är lätta att förstå. De ska vara intuitiva så att användaren behöver använda manualen så lite som möjligt. När de gäller användbarhet har de inga speciella processer för att säkerställa det. De har heller ingen person som har ansvar för den delen. De diskuterar kring användbarhet och funktioner tillsammans, men det är inget som dokumenteras och specificeras särskilt väl. Domänkunskap i kombination med sunt förnuft är det som används. Många gånger är dessutom de delar som utvecklas väldigt små. Exempelvis en kompletterande funktionalitet

till den redan existerande funktionaliteten som snabbt utvecklas och läggs till där den behövs. Sådana projekt tar ibland bara några dagar och då anser de inte att de finns något behov av tester.

Inte heller på Ninetech är man särskilt insatt i, eller använder sig av, ISO 9241. De gör heller ingen undersökning av användningskontexten. Det är inget nytt system som utvecklas utan de utvecklar ett befintligt. De har själva utvecklat systemet från grunden och har därför god domänkunskap om det. De upplever också att de är insatta i hur deras kunder arbetar med systemet eftersom de håller i utbildningar ute hos kunderna. Deras kravspecifikationer är ofta korta och informella och de skriver inte mer än vad som behövs för att medlemmarna i teamet ska ha en gemensam förståelse för målet med utvecklingen. Om de behöver skriva kravspecifikationer som ska visas för kunder så krävs lite mer omfattande dokumentation.

De har inga mätbara kriterier och gör inte heller användbarhetstester. Då de ständigt jobbar med att förbättra produkten så kan de ändra problem i efterhand. De testar själva funktionaliteten så gott de kan, men sedan läggs det direkt ut till kunden. De märker att något inte fungerar som det var tänkt i samband med att de utbildar i systemet eller om de får in mycket frågor till sin support. De anser att det i dagsläget inte är lönsamt att lägga tid och pengar på användbarhetstester och analyser eftersom de fortfarande har ganska små volymer, ca 20-30 kunder.

*”I början är det mest effektivt att bara svara på frågorna direkt i telefon. Men ju större man blir och ju fler kunder man får ju mer värt blir det att få in den typen av tester.” (Michael Lindberg, Bilaga 5, sid. 65)*

Då de ständigt jobbar med systemet så kan de löpande göra förbättringar där de tycker att det behövs. De har vissa saker som de vet att de vill göra och när någon är inne och jobbar någonstans i systemet så ändrar de även de sakerna. På så sätt blir utvecklingen väldigt iterativ.

Michael tror inte att det går att använda Scrum helt i alla sammanhang. Till exempel när det gäller gränssnitt. Innan ett stort projekt kan komma igång måste det finnas en plan. Då krävs det en del bakgrundsarbete. Men för den typen av arbete de utför så tycker han att det fungerar bra.



### Sammanställning av intervjuer

Tabell 4.1 och 4.2 visar två tabeller med sammanställningen av våra intervjuer uppdelade på samma sätt som vår intervjuguide.

Tabell 4.1 Samanställning av intervjuer- Tema Arbetsprocessen

Fråga	Intervju 1	Intervju 2	Intervju 3
Hur ser en utvecklingsprocess ut hos er?	Utomstående beställare. Scrum används främst för change request.	Scrum används inom mjukvaruutvecklingen men måste anpassas mot vattenfallsmetoden som används i andra delar av organisationen.	De försöker samla ihop interna och externa projekt i sprintar.
Vad betyder god användbarhet för dig?	User experience, intuitivt, estetiskt tilltalande. Shortcuts.	Något som kan användas och fungera. Måste passa in i operatörernas normala verksamhet.	Funktioner som är lätta att förstå.
Vilka processer använder ni för att uppnå god användbarhet?	De som jobbar med användbarheten använder en enkät som heter I Care Of Communication, ICOM.	Det saknas delvis. De har ingen insatt beställare. En operatör får titta på gränssnittsprototyper och processer.	Inga speciella processer. Är dock ute mycket hos kund och får då feedback.
Vem är det som bestämmer vad som är god användbarhet i era projekt?	Ingen speciell egentligen. Mest business analysts som jobbar med det.	De ska sätta ihop en referensgrupp med olika intressenter.	Ingen speciell. Det diskuteras när de utvärderar vad som ska göras.
Har ni någon person som är ansvarig för användbarhet i teamet? (Utbildad?)	Användbarhetskrav och dylikt bestäms utanför IT-avdelningen.	Nej. Inte mer än att han själv är systemvetare.	Nej, alla delar på ansvaret.
Vilka verktyg använder ni för att uppnå god användbarhet?	Existerande verktyg, Rational ClearQuest. Används bl.a för navigering och beroendeträd.	Använder inga mätverktyg.	Inga, bara sunt förnuft och domänkunskap.
Hur följer ni upp systemen ni utvecklar? (Användbarhet)	Inte så bra. Vill förbättra det. Viktigt med statistik, men också kvalitativa undersökningar.	Implementerar först i en pilotfabrik och utvärderar därefter innan de går vidare med implementeringen.	De får feedback från användare. Både när de håller utbildningar och genom servicedesk. Inga egna uppföljningar.

Tabell 4.2 Samanställning av intervjuer- Tema ISO 9241

Fråga	Intervju 1	Intervju 2	Intervju 3
Känner du till ISO 9241?	Nej.	Nej.	Har säkert hört talas om den.
Använder ni er av ISO 9241?	Antagligen inte.	Nej. Mer sunt förnuft och domänkunskap.	Nej.
Hur ser era kravspecifikationer ut?	De kommer utifrån. Användbarhetsbeskrivning i form av text och visuell beskrivning som visar utseende.	Luddiga och muntligt utfyllda. Mapper i efterhand.	Från några rader till offerter. Beroende på funktionaliteten och vilket syfte det har. Mer om det är utomstående inblandade.
Undersöker ni användningskontexten? (Användare, miljö, uppgifter)	Det sker i så fall utanför IT avdelningen.	Ja, tillsammans med leverantörerna till de externa systemen. Mest de andra systemen som det här systemet ska jobba mot. De har också besökt pilotfabriken.	Nej, det är oftast mindre delar som utvecklas. Sällan ett helt nytt system. Många gånger rör det sig om några dagars arbete och ett komplement till befintlig funktionalitet.
Identifierar ni mätbara användbarhetskriterier?	Inte direkt för användbarhet. Men för exempelvis svarstider och browser-kompabilitet.	Nej, operatörerna blir bara "drabbade" av systemet. Målet är att öka effektiviteten i maskinerna. Operatörerna får utbildas.	Nej.
Hur går ni tillväga för att testa användbarheten under utvecklingen? Utifrån vilka kriterier?	Business Acceptance Testing. Sällan rena användbarhetstester utan organisationen får representera slutanvändarna.	En operatör får titta på gränssnittsprototyper och processer. En superoperatör som har mappat hur de olika frabikerna gör i en viss situation. Sedan ska en referensgrupp få titta på det.	Inga användbarhetstester eller liknande. Testar själva så gott de kan. Sedan löpande kompletteringar av saker som inte riktigt fungerat som de tänkt. Dialog med kunderna ger insikt i vad som behövs göras.

## 5 Analys

*Här presenteras analysen av vårt insamlade material. Vi ska här försöka besvara vår frågeställning.*

---

### 5.1 Användbarhetsproblem

De verksamheter vi har tittat på skiljer sig mycket i fråga om vilken typ av användbarhetsproblem de handskas med. Den första verksamheten arbetar med en hemsida där många olika typer av kunder, oavsett datorvana och övriga egenskaper, ska kunna hitta information på ett tillfredsställande sätt. Hemsidan var dock redan i drift och arbetet handlade om utveckling av den befintliga sidan. Detta borde vara en verksamhet som har stort fokus på användbarhet eftersom det är ett system som ska vara åtkomligt och lättförståeligt för alla IKEAs potentiella kunder.

Vår andra intervju rörde ett system med helt andra krav på användbarhet. Slut användaren är i det här fallet en operatör som ansvarar för att sköta maskinerna. Det stora problemet för utvecklarna är att se till att systemet är kompatibelt med de andra system som det ska kommunicera med. De enda kraven på användbarhet som finns är att det nya systemet ska fungera i operatörernas normala verksamhet. Operatörerna kommer att få utbildning på systemet men de får helt enkelt ta systemet som det blir. Effektiviteten som ska ökas med hjälp av systemet är inom maskinerna och mot andra system, inte ut mot operatörerna.

I den tredje verksamheten jobbar de främst med att utveckla ett eget affärssystem som de säljer till kunder. Systemet utvecklas löpande och initiativ till förändringar kan komma antingen från dem själva eller från kunder. God användbarhet borde vara viktigt även här eftersom de utvecklar ett stort system som har många funktioner och som ska kunna användas av många olika personer i olika verksamheter.

### 5.2 Användbarhetsprocesser

Två av tre verksamma systemutvecklare som vi varit i kontakt med anser att användbarhet är en viktig del i utvecklingsprocessen, den tanken baseras på i vilket område man är verksam inom. Men att samtidigt lägga resurser på utveckling av god användbarhet verkar

vara svårt att motivera. ISO 9241 säger att genom definitionen av vad användbarhet är kan man uppnå en gemensam förståelse för användbarhet samt vilka delar som är viktiga för det specifika projektet. I detta sammanhang kan vi till viss del se en likhet mellan ISO och Scrum då man definierar vilka mål och önskemål som ska arbetas fram i ett Scrumprojekt. Klas på Arkatay Consulting berättade att de använder sig av enkäter för att säkerställa användbarhet i systemen som utvecklas. Det är egentligen ett sätt att i slutändan se om man har lyckats med att utveckla ett användarvänligt system, under vägens gång vet de inte om de är på rätt spår. Kraven som från början ställs på systemen som Klas var med i på IKEA kommer från en beställarorganisation, via en kommunikationsorganisation. Utöver enkäterna och en del statistik är uppföljning av systemen och användbarheten inte något som utförs i tillräcklig omfattning anser de själva. Enligt ISO 9241 behöver man både ta specifikation och utvärdering i beaktning. Det vi kan se är att vissa moment i utvecklingsmetoden som man använder på Akratay Consulting kan kopplas ihop med ISO 9241. Vad som är användbarhet är ofta väldigt individuellt och man har här valt att försöka identifiera browserkompatibilitet och svarstider. Om man ser dessa mätvärden som en del i användbarheten finner vi att ISO standarden kan matcha denna process. Det kan härledas till åtkomlighet i användbarhetsteorierna och effektivitet i ISO 9241. Enligt standarden bör man ta fram kritiska mätpunkter för användbarhet, men det kan vara svårt att se var, när och hur man ska mäta rent praktiskt i ett Scrumprojekt.

När de har tagit fram ett system på IKEA utför de egentligen inte några konkreta tester på användbarhet utifrån slutanvändarna, vilket går emot ISO 9241. De har istället valt att låta organisationen där systemet ska implementeras representera slutanvändarna. Det kan ge en rättvis bild, men det beror i hög grad på hur väl insatt organisationen är i användningsprocessen. För att följa ISO 9241 standarden bör tester utföras med slutanvändare. Dessutom ska mätetal identifieras för både effektivitet, ändamålsenlighet och tillfredsställelse.

Beroende på vilken verksamhet som ett system ska utvecklas i ställs olika krav på användbarhet. I vårt andra exempel, där Jonas på Tetra Pak utvecklar processsystem för industri, står inte användbarhet speciellt högt i fokus. Istället fokuserar de på effektivitet i tillverkningsprocessen och bortser från användbarhetstänkande och hur maskinoperatören upplever systemet. De har valt att använda Scrum för utveckling av mjukvarusystemen. Som Jonas själv säger har de valt denna metod för att de arbetar mot otydliga krav och det är framförallt vid den osäkerheten som Scrum passar. En fördel med Scrum är flexibiliteten i utvecklingsarbetet genom att man arbetar med mindre sprintar och tar fram små delar som till slut sätts ihop till en enhet. Samtidigt menar Seffah et al. (2005) att denna flexibilitet är en svårighet att efterleva då användbarhet ska prioriteras. Det går att kringgå denna svårighet genom att kombinera användbarhetstänk med Scrum. På Arkatay Consulting och Nintech kan vi ana denna kombination genom att de vill ha en användarvänlig produkt samtidigt som de inte vill ge avkall på processen i systemutvecklingsmetoden. Dock är det svårt att veta hur de lyckas utan att ha pratat med användarna. På det sätt som Jonas på Tetra Pak använder Scrum går det inte att se någon likvärdig koppling till användbarhet. Men användbarhet är också av mindre vikt i det fallet.

ISO 9241 talar om olika måttvärden, och om vi ser på hur Tetra Pak arbetar skulle man kunna mäta effektiviteten genom vilken kostnad eller vilken tid det tar för maskinoperatören att utföra en viss uppgift. Men operatörens arbete är inte det som avgör effektiviteten i maskinen. Det är ekonomiavdelningen som ansvarar för kontroll och uppföljning av produktionskostnader och dylikt. Med tanke på hur liten vikt operatören har för slutresultatet lägger de ändå ner en del arbete på att säkerställa användbarhet. Bland annat genom att låta en superoperatör utvärdera prototyper och processer.

På Ninetech arbetar man helt utefter de ramar som finns i Scrum när de använder metoden. Men som Michael berättar ägnar de inte mycket tid åt användbarhet i utvecklingsprojekten. Istället har de en fråga från början en bild över hur systemet ska vara. Det rör dessutom ofta små delar av systemet.

### 5.3 Användbarhetslösningar

Tabell 5.1 Jämförelse mellan ISO 9241s kriterier och resultatet av våra intervjuer.

ISO 9241 kriterier	IKEA	Tetra Pak	Ninetech
Identifiera och specificera användningskontext	Nej, det kom utifrån.	Ja, delvis. Men inte så mycket ur användbarhetssynpunkt	Nej, de förlitade sig på domänkunskap.
Mätbara kriterier och användbarhetsspecifikation.	Användbarhetsspecifikation fick de utifrån. Egna mätbara kriterier för effektivitet men inte för ändamålsenlighet eller tillfredställelse.	Enbart mätbara kriterier för ekonomiska mål.	Nej
Utvärdering av användbarheten.	Ibland, men inte så bra.	Ja, bland annat med prototyper.	Nej.
Uppföljning och förbättringar.	Ja, ständiga förbättringar. Utvärdering mest med hjälp av statistik.	Har inte kommit så långt ännu men de planerar för uppföljning.	Inte egna uppföljningar men ständiga förbättringar i de fall det visar sig behövas.

Enligt vår undersökning stämmer processerna i Scrum inte med kriterierna som finns i ISO 9241. Det kan också bli svårt att applicera ISO 9241-standarden på Scrum utan att behöva lägga till processer i metoden. För det första krävs betydligt mer förberedande arbete än vad som finns med i Scrum. Att lägga till en lång analysfas tar dock delvis bort syftet med Scrum, att snabbt få ut ett fungerande system. Därför anser vi att Scrum är mer lämplig då analysen kommer utifrån eller när delarna är så små att ingen djup analys är nödvändig. På Tetra Pak hade de mest analys med i processen. De var också de enda som utvecklade ett stort system från början med Scrummetod.

I två av våra intervjuer framkom att kravspecifikationerna för systemet helt eller delvis kommer från någon källa utanför utvecklingsteamet. På IKEA skapades kravspecifikationerna utanför IT-avdelningen så att all evetuell analys av användarkontext och dylikt skedde utanför utvecklingen. På Ninetech kom vissa krav utifrån och tillsammans med den domänkunskap utvecklarna har om systemet kunde de bedöma vad som behövde göras och var de nya funktionerna kunde passas in. Domänkunskap var också det som användes främst i projektet på Tetra Pak. Där var det mycket teknisk kunskap som var viktig och kommunikation med andra system var det som hade betydelse för effektiviteten, snarare än integration med operatörerna.

Identifieringen av användningskontext samt kravspecifikation skedde i inget fall så som det rekommenderas i ISO 9241. Istället sköttes det utanför Scrum av en beställarorganisation eller togs fram med hjälp av domänkunskap och ”sunt förnuft”. Det verkar trots allt fungera, även om vi inte har slutanvändarnas syn på saken. I de fall där beställningarna kommer utifrån är de redan färdiga när Scrum-projektet påbörjas. Några av de artiklar vi har tittat på i kapitel 2.4 föreslår just att analysera först som en möjlig utväg för att lösa användbarheten i Scrumutveckling. Även våra intervjuobjekt har påtalat behovet av en grundläggande analys *innan* det går att börja med Scrum. I vissa fall används domänkunskap för att identifiera vad som behöver göras. I de fallen finns ofta inte ens en ordentlig kravspecifikation att arbeta efter.

Brister syns även när det kommer till att identifiera mätbara kriterier och att utföra tester under utvecklingen. Vissa kriterier och tester utförs, men ganska sparsamt och enbart utifrån några få av de kriterier som beskrivs i ISO 9241. Det är främst kriterier för effektivitet och ändamålsenlighet som nämns. Tillfredsställelse verkar vara lågt prioriterat. Kanske för att det är svårt att mäta och motivera ekonomiskt. På Ninetech lägger de ingen tid på att testa under eller efter utvecklingen. Dock handlar utvecklingen hos dem ofta om väldigt små tillägg till deras existerande system. För att leva upp till ISO standardens kriterier krävs mer arbete med användbarheten under utvecklingens gång. Kanske skulle en metod enligt U-Scrum kunna förbättra den biten. Genom att ha en produktägare med användbarhetsansvar och även en användbarhetsexpert i teamet borde det vara lättare att prioritera användbarhet och leva upp till kriterierna utan att tappa allt för mycket fart i utvecklingen.

När de gäller uppföljning och förbättring av systemen så lever samtliga bättre upp till kriterierna i ISO 9241. I två av fallen, på IKEA och Ninetech verkar de jobba så att de utvecklar någonting efter mer eller mindre bestämda krav, levererar det och rättar till eventuella misstag i efterhand i nästa iteration. På IKEA genomför de en del tester och förlitar sig på statistik för att se om det de gjort fungerar. På Ninetech förlitar de sig på att kunderna själva påtalar brister antingen vid utbildning av systemen eller genom supporten. Löpande utveckling med ständiga förbättringar är grundtanken med Scrum och i de sista stegen, utvärdering och förbättring av produkten, stämmer det ganska väl överens med ISO standarden. På Tetra Pak har de inte kommit så långt i processen, men tanken är att först implementera systemet i en pilotfabrik. Det är väl ett rimligt antagande att de kommer att göra ändringar innan de fortsätter implementeringen om de upptäcker några större brister.

## 6 Slutsatser

*I detta kapitel redovisas vilka resultat och svar vi kommit fram till baserat på vår frågeställning. Även vissa ytterligare reflektioner behandlas.*

---

### 6.1 Besvarande av undersökningsfrågor

Nedan lyfter vi åter fram vår frågeställning och försöker ge svar på de delfrågor vi presenterade i kapitel 1.

#### *Hur väl uppfyller Scrum kriterierna för användbarhet som presenteras i ISO 9241?*

Slutsatsen är att Scrum inte uppfyller kriterierna i ISO 9241 särskilt väl. Samtliga intervjuer vi har gjort visar på brister i såväl analys av användningskontext, identifikation av mätbara värden och testning inom Scrumteamet. I vissa fall sker några av aktiviteterna utanför utvecklingsteamet och i vissa fall finns det med, men inte i så stor utsträckning som är önskvärt enligt standarden. Vissa delar är dock helt utelämnade. Den del i vår studie som stämde bäst överens med standarden var uppföljning och förbättring. Men uppföljningen var ofta inte så väl strukturerad. Då problem dök upp så åtgärdades de i en senare iteration men ansträngningarna att hitta problem var små. Vi upplevde att användbarhet över huvud taget inte var en särskilt prioriterad fråga. Delvis på grund av ekonomi, det är svårt att visa på direkta vinster av användbarhet och det går åt resurser att säkerställa god användbarhet,

#### *Hur kan man gå tillväga för att införa användbarhet enligt ISO 9241 processen?*

Teoretiskt anser vi att det finns möjlighet att implementera ISO 9241 standarden i Scrum. Det beror på att Scrum är så pass anpassningsbar till förändringar genom att man arbetar i sprintar. Men rent praktiskt verkar det vara svårt att införa ett tänkande som strävar efter god användbarhet och som uppfyller de kriterier som ställs i ISO 9241. Det vi kan se idag är att det inte finns någon process i Scrum som direkt säkerställer god användbarhet.

I vår undersökning upptäckte vi att ingen av de tillfrågade lägger någon större vikt vid att uppnå god användbarhet, och ännu mindre att använda sig av ISO 9241. För att säkerställa god användbarhet i systemet tror vi att den bästa lösningen är att koppla in ett mindre team

inom Scrum teamet som arbetar i förväg, bereder väg och undanröjer hinder i utvecklingsarbetet. Det skapas då tillfällen under projektets gång där användbarhet i förmodellen kan utvärderas och vi säkerställer ännu en aktivitet som ISO 9241 kräver. Dessutom kan det vara nödvändigt med en analysfas innan Scrumteamet ska börja arbeta med systemet, detta för att säkerställa att en korrekt analys har gjorts av användningskontexten. Det är också en aktivitet som förespråkas av ISO 9241.

## 6.2 Ytterligare slutsatser

De utvecklare vi intervjuat anser att det är viktigt att sträva efter användbarhet, men få lägger ner tid och resurser på att säkerställa god användbarhet. Istället har domänkunskap och sunt förnuft varit det utvecklarna förlitat sig på i användbarhetsfrågor. Delvis har också ansvaret för användbarheten legat utanför utvecklingsavdelningen och Scrumteamet. Kanske är det så att Scrum är en metod som bäst används i projekt där användbarhet antingen inte är så viktig eller när utvecklarna får färdiga kravspecifikationer utifrån. Vid underhåll och utveckling av ett befintligt system verkar Scrum också vara en väl fungerande utvecklingsmetod. Då har utvecklarna god domänkunskap och de har ett färdigt system, en helhet, som de kan passa in den nya funktionaliteten i. Kritiken mot Scrum ligger ju just i att det är svårt att i början av ett projekt veta hur slutresultatet kommer att se ut och därmed planera för bland annat gränssnitt. Om då helheten redan finns så krävs det kanske inte alltid avancerade analyser och tester för att utöka systemet. Om systemen hade blivit bättre av sådana aktiviteter är omöjligt för oss att svara på. Men uppenbarligen anser utvecklarna själva att domänkunskap och sunt förnuft räcker för att skapa acceptabla resultat. Vad användarna anser om saken berör inte den här uppsatsen.

Användbarhet var ingen prioriterad fråga som vi ser det. Vi kunde inte heller identifiera några andra aktiviteter för att säkerställa god användbarhet än de som finns med i ISO standarden. Dock utformade vi våra intervjuer med fokus på ISO så vi kan ha missat andra typer av aktiviteter.

## 6.3 Kunskapsbidrag

Kunskapsbidraget av vår uppsats är blygsamt. Vi har kunnat konstatera att Scrum varken i teorin eller praktiken stämmer överens med kriterierna för att uppnå god användbarhet som beskrivs i ISO 9241. Däremot kan vi inte svara på om det i sin tur medför att systemen som utvecklas med Scrum har sämre användbarhet än om ISO 9241 hade uppfyllts. Enligt vår studie sköts användbarhetsfrågor inom Scrumteamet med domänkunskap och sunt förnuft, i kombination med en iterativ process där systemet ständigt utvecklas och förbättras. Det tyder på att metoden passar bäst i projekt av mindre skala och när det handlar om att vidareutveckla existerande system. Vår studie är dock för liten för att vi ska kunna dra några generella slutsatser.



## Bilagor

---

### Bilaga 1 - Exempel

Exempel på attribut i användningskontexten.(ISO 9241-11)

Users	Tasks	Equipment
<p><b>User types</b>                      Primary                      Secondary and indirect users</p> <p>Product skill/knowledge                      System skill/knowledge                      Task experience                      Organizational experience                      Level of training                      Input device skills                      Qualifications                      Language skills                      General Knowledge</p> <p><b>Personal attributes</b>                      Age                      Gender                      Physical capabilities                      Physical limitations and disabilities                      Intellectual ability                      Attitude                      Motivation</p>	<p>Task breakdown                      Task name                      Task frequency of use                      Task duration                      Task flexibility                      Physical and mental demands                      Task dependencies                      Task output                      Risk resulting from error                      Safety critical demands</p>	<p><b>Basic description</b>                      Product identification                      Product description                      Main application areas</p> <p><b>Specification</b>                      Hardware                      Software                      Materials                      Services                      Other items</p>

Organizational environment	Technical environment	Physical environment
<p><b>Structure</b>                      Hours of work                      Group working                      Job Function                      Work practices                      Assistance                      Interruptions                      Management structure                      Communication structure</p> <p><b>Attitudes and culture</b>                      Policy on use of computers                      Organizational aims                      Industrial relations</p> <p><b>Job Design</b>                      Job flexibility                      Performance monitoring                      Performance feedback                      Pacing                      Autonomy                      Discretion                      Intellectual ability</p>	<p><b>Configuration</b>                      Hardware                      Software                      Reference materials</p>	<p><b>Workplace conditions</b>                      Atmospheric conditions                      Auditory environment                      Thermal environment                      Visual environment                      Environmental instability</p> <p><b>Workplace design</b>                      Space and furniture                      User posture                      Location</p> <p><b>Workplace safety</b>                      Health hazards                      Protective clothing and equipment</p>

## Bilaga 2 - Intervjuguide

### *Inledande frågor.*

1. Vilket företag är du verksam på?
2. Vad är dina ansvarsområden?
3. Vilka direkta arbetsuppgifter har du?
4. Vilken typ av uppdrag arbetar ni med?
5. Hur många är ni i utvecklingsteamet?

### *Frågor med inriktning på arbetssätt utifrån ett användbarhetsperspektiv.*

6. Hur ser en utvecklingsprocess ut hos er?
7. Vad betyder god användbarhet för dig?
8. Vilka processer använder ni för att uppnå god användbarhet?
9. Vem är det som bestämmer vad som är god användbarhet i era projekt?
10. Har ni någon person som är ansvarig för användbarhet i teamet? (utbildad?)
11. Vilka verktyg använder ni för att uppnå god användbarhet?
12. Hur följer ni upp systemen ni utvecklar? (användbarhet)

### *Frågor med inriktning mot ISO 9241.*

13. Känner du till ISO 9241?
14. Använder ni er av ISO 9241?
15. Hur ser era kravspecifikationer ut?
16. Hur undersöker ni användningskontexten? (användare, Miljö, uppgifter?)
17. På vilket sätt identifierar ni mätbara användbarhetskriterier?

18. Hur går ni tillväga för att testa användbarheten under utvecklingen? Utifrån vilka kriterier

*Avslutande frågor:*

19. Något övrigt du vill tillägga?
20. Kan vi kontakta dig senare om vi har fler frågor?

## Bilaga 3 – Intervju 1

*Intervju med Klas Skogmar från Arkatay Consulting.*

---

I texten visas (\*) vid otydlig inspelning.

DA: Vilket företag är du verksam på

KS: Det är Arkatay consulting, och jag är vice VD. Det säger inte så mycket, vi är sex personer så det innebär bara att jag är bollplank åt vd:n, när det gäller strategiska beslut

AR: Vilka direkta arbetsuppgifter har du?

KS: Ja, det beror lite på uppdragen. Som nu senast var jag ute på IKEA service..., där jag skulle hjälpa till och ta fram team, som skulle jobba lite mera agilt och innan dess har jag jobbat då jag satt på verksamhetsidan alltså på business, tidigare har jag suttit på IT sidan en massa år på samma avdelning och utforma webbsida... utvecklare... problem... Manager.. och lite projektledning

DA: Vilken typ av uppdrag?

KS: Det är framförallt projektledningsuppdrag jag har, och det kan variera lite på vilken typ av projektledning det är. Men ofta är det traditionell projektledning men i och med att vi fokuserar på IT-projekt finns det mycket fokus på agila metoder som Scrum.

AR: Hur många är ni, det varierar förstås?

KS: Ja, men vi är 6 personer i bolaget.

DA: Hur många är ni i teamen som ni arbetar med?

KS: I och med att vi är konsulter arbetar vi inte i team själva. Jag tolkar frågan som att när vi väl arbetar. Då kan det variera mycket, allt från 5 till 40 personer inom teamen, lite beroende på. Nu senast var det en förstudie för att ta fram en ny organisation då var vi bara 3 personer tillsammans. Då skissade vi på en framtida organisationsutveckling

AR: Hur ser en utvecklingsprocess ut hos er, då får det vara utifrån ett exempel då ni använder Scrum?

KS: Det är ganska intressant för man vill ha svar på dokumentation, vad som är krav på, man ska tillgodose alla intressenter och stakeholders intressen. och Se till att man får ut (\*) makt men man jobbar också med många parallella projekt som använder samma kodbas till att integrera olika projekt. Så det är uppdelat ur en aspekt som påverkar hur agilt man kan jobba. Det finns begränsningar i det agila sättet att jobba, baserat på hur separerbar kodbasen är. Så, ofta på IKEA kanske vi arbetar med både PPF, RUP och sen delvis med Scrum.

DA: Så det är en ganska ny metod?

KS: Ja precis, det är ganska nytt på IKEA, och då har man valt att börja jobba med change request. Det känns egentligen ganska naturligt att jobba med Scrum för change request. Alltså man har paketerade småpaket, man har redan någon slags backlog, det enda man behöver göra är att strukturera upp hur man prioriterar, och hur mycket man ...tex, och vem som beslutar för det. Har någon .....produktägare. Och sen jobbar man lite mer agilt inom projektet. Man leverera färdigt ett visst antal featcher eller changes innan man börjar nästa batchlog. Alltså det lämpar sig ganska väl. Jag läste någonstans att när man jobbar med Scrum projekt eller när man jobbar med agila metoder så är det bara första iterationen som är ett projekt, sen kan man egentligen jobba mer i team i organisationen. det är en lite intressant tanke att ha i bakhuvudet. så change request lämpar sig ganska väl att jobba med. Man har ett eget projekt.

DA: Den här dokumentation, är det något ni tar fram internt med kunderna?

Det är mer intern dokumentation i och med att där har jag jobbat för ett projekt med webbsidor där all dokumentation sker till slutanvändaren, så att det är bara intern dokumentation. För att göra förändringar, underhålla. För att överblicka krav.

AR: Vad betyder god användbarhet för dig?

KS: Ja, god användbarhet är. Det är mycket prat om user experience också, att det är en upplevelse. I upplevelsen så ligger det dels i hur man ska, någon form av estetik, det ska vara estetiskt tilltalande. Det ska vara intuitivt, man ska förstå att man trycker på en knapp, vad som händer redan innan man gör det. Det ska vara tydligt, man ska kunna se att det är en länk, att man kommer någon annanstans. Man ska ha så få förvåningsmoment som möjligt. Man ska inte bli förvånade av själva funktionen. Sen skulle man kunna tänka sig att visualiseringen, att man förstår att man klickar på en knapp så kommer man till en ny sida. Sättet man går över till den nya sidan, det kan vara förvånande. Jag har tittat lite på Apple som har lagt ut user guidelines, som pratar lite om sådana saker att man hela tiden ska göra det enkelt för användarna att gå vidare till en ny sida, eller man ska förstå intuitivt vad det innebär. Men att själva övergången eller hur det ser ut kan vara lite svårt att göra.

Man ska så att säga chocka genom att sättet inte vara. Man ska inte göra något annat än vad man tror att man gör. Man kan göra upplevelsen bra ändå. Det är fortfarande samma funktion, men med snyggare upplevelse.

Sånt gör tex Apples Iphone, det finns små detaljer, dem avslöjar tex att det mesta är ganska intuitivt. Men då är det att, när du trycker på något kan det ibland vara bra att hoppa till lite eller så för att det ska se snyggt ut. Sen finns det en annan upplevelse att man ska kunna göra en funktion snabbare om man vill, tex shortcuts. Det ska kunna finnas någon sådan här, alltså typ något man gör på rutin. Ska det finnas sätt att lära sig göra dem på ett effektivare sätt.

AR: Klick som man lägger in?

KS: Nja, Det kan vara på allt möjligt. Men också om man tex läser ur apples användbarhets guide att om man använder 3 fingrar eller 2 fingrar i något eller om man drar fingret åt sidan så betyder det någonting, så blir det som en shortcut. Det finns ju flera metoder på datorn att använda tangentbordet mycket är också att använda shortcut effektivt för att göra något snabbare. Och då är det viktigt att genom användbarhet måste man också ha färdig hantering till dem som inte fattar det här med shortcut, så måste man se till att dem kan lösa det på det vanliga sättet.

Jag vet av egen erfaren, ibland blir man jätteförvånad då man råkar man komma åt någon shortcut med olika fingersättningar.

Det är bra att dem finns där, men man måste också kunna navigera på något intuitivt och enkelt sätt. Jag tror att det är lite så, lite estetik, lite att det är intuitivt, lite att man kan jobba snabbt om man vill det.

DA: Har ni någon speciell person som bestämmer god användbarhet i era projekt?

KS: Nja, dem har varit lite dåliga. Men man gör också användbarhetsstudier, men dem kommer ganska sent. Jag vet att man håller på att rekrytera någon som är användbarhetsexpert, men annars är det dem som är business analysts som har lite att göra med det arbetet. Dem som översätter dem kraven till någon slags semitekniska krav.

AR: Men det här gör ni ganska sent?

KS: Ja, att ta in en användbarhetsexpert.

AR: Vad har ni för processer för att uppnå god användbarhet, har ni någon speciella användarstudier?

KS: Nja, dem som jobbar med användbarhet, där har det varit dokumentations delen av en enkät som heter I Care Of Communication ICOM. Dem säger allt ska fungera, eller inte hur allt ska fungera, dem sätter de visuella effekterna. Och sen kanske det har kommit något krav för att vi behöver nytt utseende eller, det är lite luddigt.

AR: Det står lite utanför utvecklingsprocessen, då får man ganska...?

KS: Ja, se det som en reklambyrå, alltså om man vill utveckla något så tar man hjälp av en reklambyrå.

AR: Den analysen av vad användaren behöver sker inte där inne utan...

KS: Nä utan, det sker...man kan se det som olika organisationer, beställarorganisationen och IT organisationen och en kommunikationsorganisation Och då är det beställarorganisationen som beställer från de andra två. Och då beställer man ofta från dem samtidigt, eller ibland beställer man från IT först och ibland från kommunikation först. Men det kanske skulle ha funkade på något annat sätt. För typ kanske 5 år sen var det lätt för att man upplevde som att en verksamhet som beställde från IT som sedan beställde från organisationen. medan det idag börjar bli lite nästan .... som beställer från kommunikationsorganisationen som sen beställer från IT. Och det känns mer naturligt, därför att man försöker formulera ett budskap eller kommunikation och sen tittar man på dem tekniska begränsningarna.

(Klas avbröt kort för ett Telefonsamtal)

AR: när vi har tittat på Scrum, är ett av problemen just ur användbarhetssynpunkt att det är svårt att man med små etapper i taget få en bild av hur helheten ser ut. Men det problemet har inte ni, ni har redan en färdig bild?

KS: Det är en skillnad på om man utvecklar en produkt från början eller om man har en produkt man vill förbättra. Och det är alltså för det mesta så att man har en produkt som försöker jobba vidare på. Och då har man ett projekt som förbättrar något som redan finns, men man kanske ändrar lite i funktionaliteten. Man kanske integrerar mot något annat backend system då påverkar man inte alls användbarheten. Eller också i agila projekt där man går från 5-steps check out till en 4-steps check out. Dvs att man vill förbättra användbarheten vid köpprocessen på nätet. Och då var en del i det mot användaren det visuella 4-steps check men del var också att påverka att man integrerar mot andra betalningslösningar och säkra uppkopplingen. Användbarheten blir liksom bara ett projekt för att förbättra användbarheten i IT. Därför att man jobbar med en mer specifik del av Itn och sen förbättrar man det. Problemet kan då ibland vara att vissa saker släpar efter, att man inte får ett helhetsgrepp.

DA: Inom vissa delar av användbarheten eller?

KS: Ja, att det inte visuellt ser likadant ut på en sida som en annan.. men det är en del som säger att dem projekten är kanske 1:års projekt som jobbar med det. Och där har man också samma problem som inom Scrum att man jobbar med små delar sen får man kanske efteråt se över om helheten passar. Därför jag har ofta inom Scrum fått hör att när man väl är färdig liksom, och säger att nu känns det bra. Då lägger man ofta till en iteration där man säger att nu förbättrar vi sakerna för att finslipa på detaljer och bygger färdig funktion.



DA: Ja, det måste ju anpassas i slutet av projektet.

KS: Nä, det måste inte det. Det finns vissa som menar att det finns vissa negativa aspekter, för då förstör du Scrumflödet varje gång man gör en extra iteration. Med en iteration kan man göra extra labb tester tex, vilket är viktigt i många stora företag, då förstår du också Scrumcyklerna. Därför att meningen med varje Scrumcykel är att du ska kunna leverera en färdig form. Kan du inte göra det, kan du lika gärna ha en PPM eller RUP. Där man först gör sina små delar och sen avslutar man med testning. Men det är ju det man försöker komma ifrån.

DA: Vad jag försått är det väl mest i mindre bolag eller mindre projekt där det fungerar bättre med agila metoder?

KS: Ja, precis. Men att det funkar, dem menar det, Scrumskaparna att man kan låta ett team, att man ska hålla ett Scrumteam litet. men det kräver att dem jobba i olika separata, att man har lite beroenden inom koderna, att om jag jobbar med en sak inte påverkar ett annat team. ju mer de påverkar varandra ju svårare är det att jobba med agila metoder så som det är tänkt att man ska göra. Så det handlar jättemycket om beroenden. Det är samma som när man läser objektorientering. Så pratar man också om att det ska vara så få beroenden till varje klass som möjligt. Det är nästan samma princip för Scrum, att man har två beroenden. Tydliga gränssnitt mellan dem olika delarna.

AR: Och sen binder man ihop dem?

KS: Ja, precis.

AR: Det krävs lite kommunikation mellan grupperna då, för att...

KS: Ja, och då har man något man kallar Scrum of Scrum's, som man då knyter ihop olika Scrumteam. Och jag har inte jobbat med de bitarna, men kan lite teori om det.

Har ni läst något om Scrum... (\*).

AR: Är det några speciella verktyg ni använder för att uppnå god användbarhet?

KS: Ja, jag är ganska intresserad av verktyg, jag kan skicka en länk. Jag har skickat ett långt inlägg om man går in på [skogmar.com/klas](http://skogmar.com/klas). Så kan ni titta på andra verktyg. Men det man jobbar med på tex IKEA är att man försöker använda dem existerande verktyg för att använda dem till Scrum. Då har man ofta ett verktyg som tex Rational ClearQuest som är ett verktyg för att hantera change Quest's och annat problem som integrerar mot rationalsclear case som är en version som finns i systemet. Det är ett större verktyg, men mycket grafiska beroende träd och navigering.

DA: Följer ni upp de här systemen som ni utvecklar, i projekten ni är ute med?

KS: De interna verktygen? Nja, man försöker anpassa processen, arbetsflödet efter verktygen. Om man gör det kan man sätta prioritering, så man kan titta på, man bygger upp någon sorts backlogg. Problemet är att man, för att kunna ta ställning till, alltså Scrum säger att man ska ta sin backlogg och så ska man arbeta igenom den inför sitt projekt. Och välja ut dem som är viktiga. Men problemet är att man måste göra en analys för att kunna ta ställning till om dem är viktiga eller inte. Och ibland säger dem att man får en dialog kring varje ärende, så då kan man diskutera; ”ja ok, det tar så lång tid. Men om man skär bort den biten, hur lång tid tar det då?” så får man en dialog. Så kan man välja det som, man får en lite på det som är viktigast i förhållande till tiden. Det är en aspekt, det är lite return of investment kan man säga. Vad man tjänar på det man utvecklar.

Problemet är om man har en för stor, komplex backlog där man inte hinner analysera, hur man skulle göra det på ett möte inom varje sprint så skulle det ju ta en vecka att komma igenom. Därför gör man också en preliminär analys, så här många timmar tror vi det tar innan man kommer till ett slags beslutsorgan. Istället för att bara ha en produktägare, där gör man det också lite enkelt för sig genom att bara ha en produktägare. I verkligheten har du ju många olika stakeholders, intressenter som vill påverka vad som är viktigt. Många olika organisationer och personer som vill vara med och fatta beslut, och det gäller att väga dem mot varandra. Och det kan vara en person som verkligen hjälper till att prioritera mellan olika stakeholders.

Men det är viktigt att man får ta del av alla olika stakeholders också. Det är svårt och komplext att hantera en sådan situation i verkliga livet. Det kan vara till exempel, man har en organisation, en webbsida till exempel, då har man ett system som levererar data som ska visas på websidan. Så vill man ha lite mer produktinformation som man vet finns i ett beroendesystem men man får det inte levererat till sig. Då skickar man en change request dit. Då är man själv en stakeholder till dem. Om man då jobbar i Scrum så har man en produktägare. Och hur får man honom att prioritera sin change request över de andra change requests de har internt? Då måste han ha en massa information. Man kan inte bara sitta i ett möte och bestämma prioritet lite ad hoc. Man behöver göra lite grundläggande analys. Man kan ta till sig mycket av Scrumprincipen, men man måste tänka igenom det, kan vi göra det här i Scrum. Nä, vi måste nog göra det här först, sen kan vi Scruma.

AR: Mycket förarbete alltså?

KS: Mmm, det blir det.

AR: De färdiga systemen då? Hur följer ni upp dem? Nu gör ni kanske inte hela system men när ni är färdiga med en del, om man säger så? Med fokus på användbarhet då...

KS: Jaa, där är man ganska dålig, det vill man bli bättre på. Och tanken är... Alltså egentligen är webb en av de bättre bitarna att jobba med agila metoder. Man kan trycka ut något ganska snabbt till slutanvändarna, och om de tycker att det är dåligt och klagar så kan man lätt dra tillbaka det. Man har inte riktigt den möjligheten om man skickar ut en cd-skiva med program på. Webb är fantastiskt bra, man kan bara trycka ut saker, mäta om det

blev bra och fråga användarna ”tyckte du att det här blev bra?” och tyckte de inte det så kan man ta bort det. Det är en av styrkorna med de här metoderna, just till webbtjänster.

Att följa upp användbarhet tycker jag handlar väldigt mycket om statistik. Och då kan man göra så att man har en grupp som får se en sida och en grupp som får se en annan sida och så kan man till exempel mäta hur många som lämnar sidan i respektive grupp, hur många som fortsätter och hur länge de är på sidan och så försöker man bilda sig en uppfattning om vad som är bäst. Eller så frågar man. Men det kan vara lite påträngande, själv trycker jag alltid bort sådana undersökningar. Det gäller att kunna hitta hur man kan visa det här statistiskt. Jag vill ju att de så snabbt som möjligt ska kunna ta sig från förstasidan till produktsidan. Med hjälp av statistik kan jag kontrollera det, jag kan mäta hur lång tid det tar för användaren att ta sig från första sidan till produktsidan. Så då kan man ha det som nyckeltal och sen se hur man kan förbättra det.

(Avbrott för byte av rum)

KS: Då när man jobbar med statistik passar det ganska bra med agila metoder egentligen. Man vill ju ha en så kort iterationscykel som möjligt, då kan man ha en två veckors Scrumiteration, och så lägga ut saker och mäta om det blev bättre eller inte så kan man ta bort det om det blev dåligt, kanske i nästa iteration. Då vet man att då förbättrar man hela tiden användbarheten. Är man en liten organisation kanske det är lättare, då kanske man kan ha en veckors iterationer för att förbättra användbarheten. Man kan ha ett Scrumteam som jobbar i en veckors iterationer som jobbar med mindre förbättringar och ett Scrumteam som jobbar i fyra veckors iterationer som jobbar med större saker. Man brukar säga att man ska ha en rytm, en takt, ofta vill man att man har sagt att det ska vara fyra veckors iterationer, då ska man hålla fyra veckor. Fast jag kan tycka att det kan vara olika takt för olika typer av aktiviteter. Det är ingenting jag sett någon annanstans, men det känns naturligt. Om man har projekt som är sex månader för infrastrukturändringar för ett backend system som ska vara lite tröga. Vi kan inte ändra gränssnitten mellan backend to backend systemen jättesnabbt.

AR: Nu kommer vi in lite på ISO 9241. Det var inget som du kände till?

KS: Nej det gör jag inte.

AR: Då är det antagligen inget ni använder er av då heller?

KS: Det är möjligt men det känner jag inte till.

AR: Hmm, det här stämmer inte riktigt mot er verksamhet... Men vi kan ta... Hur ser era kravspecifikationer ut?

KS: Det kommer oftast i två delar. Det är alltså någon slags användbarhetsbeskrivningar som är i textform. Sen är det också en visuell beskrivning som visar att såhär ska det se ut. Ofta är det som underhålls bara den textdelen, den visuella informationen om hur

webbsidan ska se ut levereras bara en gång. Det finns en viss problematik i det. Så det finns någon slags kravdokument för ett projekt, fast de slutar vara kravdokument när man gått igenom iterationen. Vilket är lite lustigt. Och där har vi ett dilemma när det gäller just användbarhet. Det finns vissa som menar då, att jobbar man med webbprojekt, då ska man bygga en slags mockup tidigt i ett projekt och använda det som... och som fungerar, som man kan navigera i men man har inga riktiga värden och ingen kod i slutändan. Det finns verktyg för att bygga sådana verktyg också. Så menar man att utifrån den färdiga mockupen så ska man bygga sin sida och skapa sin kod. Problemet är bara att i användbarhet så är det ju inte bara det visuella, utan t.ex. hur sökning fungerar. Om man skriver ett sökord, hur snabbt man kan hitta det man söker. Då funkar ju inte mockups. Hur sökmotorn fungerar är en viktig del av användarupplevelsen och att man har en väl fungerande sökmotor kan man inte testa med en mockup. Då får man ha andra sätt.

AR: Men då har man ändå börjat lite från användbarhetssidan, med mockups....

KS: Man börjar ofta med mockups bara i photoshop, bara grafiskt då så det är lite annorlunda. Då är det antal pixlar och så som man mäter när man sen ska bygga.

AR: Hmm, frågan om hur ni undersöker användningskontexten passar inte riktigt in här eftersom ni får det utifrån, men identifierar ni mätbara användbarhetskriterier?

KS: Det finns ju vissa saker som vi mäter, som t.ex. tid för att sidan ska visas. Det är ju också en del av användbarhet men det glider in i de icke-funktionella kraven lite grann. Där kommer det in lite användbarhet också, att det ska funka med ett visst antal webbrowsers t.ex. Däremot är man nog ganska dålig på att formulera användbarhetskrav i termer av kundnöjdhet och att följa upp dem. Man har ju sina användbarhetsstudier men de bygger inte på någon slags kravstudie utan de är mer för att följa upp att man gör rätt saker. Om du förstår skillnaden? Man skulle kunna sätta mätetal. tex. vi vill att 80% ska förstå vad den här knappen innebär, att man har det som ett krav in och sen så får man följa upp det. Men istället gör man i användbarhetsstudier vad är problemet med det vi gjorde? Det är olika sätt att se på det.

AR: Hur testar ni användbarheten under utvecklingen?

KS: Jaaa, man har ju sån Business Acceptance Testing. Man börjar med någon slags business requirement, nån kravställning, och sen bryter man ner den lite grann. Man kanske har nån business analyst som formulerar interna krav eller tekniska krav. Sen kommer en arkitekt in i slutändan och skapar arkitekтуella krav och tillslut kommer det till utvecklaren. Han får fatta små beslut själv, tex om hur man ska optimera prestandan eller att jag behöver lösa de här klasserna på det här sättet. Och sen när man testar så börjar han med att testa sig själv, sen testar man de arkitekтуella besluten och till sist på business nivån. Problemet är att i användbarhetstesterna är det ju verksamheten, organisationen som får representera slutanvändarna. Man gör liksom inga användbarhetsstudier, eller jo, i vissa projekt gör man användbarhetsstudier..

AR: Under projekten?

KS: Ja, under projekten. Ofta hinner man inte göra någonting åt dem bristerna dock.

AR: Ändå lägger man tid på att göra testerna?

KS: Ja, man kanske hinner initiera förändringar som gör att det inte är problem så länge. Eller man kanske kan göra små justeringar.

AR: Ja, det var väl ungefär det vi ville fråga om. Är det något vi inte pratat om som känns relevant?

(I fortsättningen har vi plockat ut det som är relevant för uppsatsen, vi diskuterade en del annat som är intressant, men som inte är relevant för vår frågeställning)

KS: Det finns mycket som är intressant med Scrum, det finns mycket som jag själv har funderat igenom. ... Om man tex ska ha ett fastpris-kontrakt, att man säger att vi vill ha den här funktionaliteten till ett fast pris. Det är ofta väldigt svårt om man har en externkund som säger att vi vill ha det här till ett fast pris, då är det väldigt svårt att jobba med agila metoder, för principen är att förändringar i princip ska vara gratis, man ska anamma förändringar. Om kunden säger att det här vill vi ha, och så kör man igång och så har man bestämt sig för ett fast pris. Sen säger kunden efter en tid att nå vi vill ha något annat istället. Jobbar man då hela tiden mot samma timtaxa då måste man säga att nu får ni inget mer. Då funkar det med fast pris. Men om kunden tror att vi har en lista på krav, men han vill också ha med något annat, då kommer det inte att funka. Då får man säga att om du lägger till någonting, då tar vi också bort någonting.

Det som är bra med Scrum är att man tvingar folk att prioritera, folk är extremt dåliga på att prioritera. Genom att hela tiden ställa frågan: vad tycker du är viktigast? Det här eller det här? Och att göra det kontinuerligt, så leder det upp till ganska bra resultat. Om man inte tvingas välja det som är viktigt, då väljer man allt. Och det är det som är det traditionella problemet. Det är förtjänsten av de här agila metoderna, att man blir mycket bättre på att prioritera.

...

När man testar så räcker det inte bara med enkäter, de är för kvantitativa. Man behöver också kvalitativa. Man behöver någon som står och tittar över axeln på någon som använder det. Och den typer av studier gör vi på IKEA också, att man göra sådana kvalitativa studier. Att vi sätter en person och säger "gå in till "Billy" -produkten och lägg den i varukorgen" och så tittar man på hur lätt de har att hitta de sakerna och hur de navigerar med musen. Så kvalitativa studier är jätteviktigt.

Det är svårt att avgöra om någon ägnar en minut på en sida, om det är för att de försöker hitta någonting eller om de läser texter eller går och hämtar kaffe.

Någonting som jag tycker är viktigt och som jag föreslog på IKEA är att man borde ha nyckeltal som är att de här sidorna vill vi att användarna ska lämna sidan ifrån. De kommer alltid att lämna siten på någon sida, då är det bra om det sker på sidor som man själv har valt. En produktsida kanske eller att man kollar om den finns i lager. Då vet man att nu har de kollat om det finns i lagret och har lämnat siten då vet man att de åker till varuhuset och köper den sen. Så det är en bra sida att sluta på. Man får fundera ut några sådana som känns viktiga och sen får man kontrollera att man styr mot det.

Man ska så långt som möjligt mäta med statistik, men jag tror man måste komplettera med kvalitativa. Tex man går in och lägger varan i varukorgen och efteråt får man upp ”Hur tyckte du det kändes? Varför sökte du inte efter den? Jaha, kunde man söka?” Då får man en dialog med folk.

## Bilaga 4 – Intervju 2

*Intervju med Jonas Wallander från Tetra Pak.*

---

I texten visas (\*) vid otydlig inspelning.

DA: Ja, först har vi lite bakgrundsfrågor. Vilket företag är du verksam på?

JW: Tetra pak. Packaging solutions heter de. Om man ska vara noggrann.

AR: Vad är dina ansvarsområden?

JW: Alltså jag är anställd som någon sorts informationssystem. Alltså utveckling av informationssystem. Så jag utvecklar.

AR: Vad har du då för arbetsuppgifter?

JW: Det är ganska nära produktionen, så att det finns en massa applikationssystem och. Så det är lite grann att knyta samman, alltså det är ett större behov av att göra något som inte går att göra i ett styrsystem och lite såna grejer. Så kan jag antingen göra det, om det är en mindre grej på egen hand eller som i det här som är lite större får det bli ett projekt med ett team. Och, ja, sen ...frågor och. ja man kan säga en massa små bitar som ligger ett steg bakom och som styr systemen, så traditionellt sett där jag jobbar, det är ju en avdelning som tar fram automation för maskiner, och maskinerna har blivit mer och mer sammankopplade med andra grejer så att säga, att man behöver ha nätverk mellan oss. Så det är relativt nytt kan man säga. Att man behöver den kompetensen i den avdelningen så att säga. För sen dem som jobbar med nätverk och IT inom Tetra pak, sitter i ett helt annat bolag som nästan inte vet vad produktionen är. Det är mer kontors och affärssystem och lite såna grejer.

AR: Vilken typ av uppdrag, då är det interna uppdrag?

JW: Det är bara interna uppdrag.

AR: Hur många är ni, det har du redan svarat på?

JW: Min avdelning, ja det är lite svårt att säga. Min avdelning där är vi sju och sen bolaget är ju typ 100-200 tjänster.

DA: men ni är alltså sju stycken i utvecklingsgrupperna?

JW: Alltså jag sitter i en grupp som egentligen är lite konstigt konfigurerad. Den handlar om inline quality controll. Så det är in inspektionssystem, så då har jag mer eller mindre hamnat i den gruppen. Så att det finns visst ingen avdelning som sysslar med det jag gör. Ja, det gör det kanske snart, men inte just nu. Utan det är liksom man ser det som att det är ok, det hänger ihop med någon sorts insamlings... datainsamling och såna grejer.

AR: Hur ser en utvecklingsprocess ut, när ni håller på med projekt

JW: Alltså den interna processen, det är en egenutvecklad. Men den är alltså vattenfallsbaserad helt klart. Med tydliga faser. Och egentligen är den delen där jag jobbar där finns det ingen ren teknikutveckling. Utan man ser det mera som, nu implementerar man någonting som man vet någonting om. Men, just inom det jag håller på med, där kanske man inte insett att det inte är att bara göra det, så att säga. Så där ligger vi lite ikläm så att säga. Att den interna processen förutsätter att man kan gå igenom bara rakt upp och ner och bara göra det. Och sedan att vi inte alls är så väldefinierat som dem andra projekten som man har gjort likartade grejer tio gånger innan.

DA: Men ni använder Scrum som metod?

JW: Ja, jag kände liksom i det här projektet, det större projektet som jag är med i då, att där måste vi ha någon sorts agilt, för det är ingen som kan peka jättehårt på kraven i det. Utan vi måste hitta på eller komma fram till de själva. Men vi kunde liksom inte helt och hållet byta för hela projektet då. Så då är själva mjukvaruutvecklingen Scrum och resten är den traditionella processen.

DA: ok.

AR: ok, då kör ni lite parallellt?

JW: Ja, för då finns det både. Vi har liksom en roll som är utanför Scrumteamet som är traditionell projektledare. Flera stycken faktiskt, för det är så stort att det är flera delprojekt. Vi har en projektledare som är traditionell och sen är det, och så är jag produktägare i Scrum mening då, också en Scum master som sitter i teamet.

DA: ja, det är Scrum vi är mest intresserade av

JW: ja, så det blir liksom en liten bubbla så här, det man visar, det måste finnas artefakter i den traditionella processen så att säga. Så får jag försöka mappa ihop då så att säga, så att. Att det på något sätt ser lik...

AR: Men då ...(\*)...

JW: Ja, nej utvecklarna ser bara Scrum. Medans jag och projektledaren, vi får ju liksom



mappa ihop då så att vi kan översätta burnout och sånt till någon sorts plan, att det ska se ut som att vi har planerat allting jättebra och så här. Men att, men det är mera siffrorna som visar på att vi har koll så att säga. Att man kan lita på. Men vi måste ändå på något sätt ha någon sorts...

AR: Passa ihop med färdiga delar..

JW: Ja, vi måste leverera en plan och visa att vi följer den. Men det blir då lite mera ett sorts spel så att säga, att det liksom inte avviker sig från den planen det reagerar på. Utan det är mätetalet i Scrum mening så att säga Velocity inte är som den borde vara.

AR: Ja, vi har ju koppling till användbarhet i våran uppsats. Vad betyder god användbarhet för dig?

JW: Mmm, ja, nu vet jag väl ungefär definitionen på användbarhet, så det är väl svårt att säga om jag kan frikoppla mig från den definitionen. Men, alltså jag sitter med domänkunskap så att säga, så för mig är det att göra någonting som faktiskt kommer att användas och fungerar så att säga. Att vi inte ställer till det, kan man nästan säga. För att vi har ett övergripande mål som satt varför vi gör det så att säga. Och sen är det något som ska pluggas in i operatörernas normala verklighet så att säga. Dem gör något idag. Och dem ska göra ungefär samma sak fast lite till med vårt system. Och då inte det bli för mycket konstigt.

AR: nä, ni får inte ändra arbetssätt och...?

JW: Nä, inte så att det blir dubbelt så mycket att göra och så där.

DA: det är lite av processen du beskriver nu också? Nästa fråga är vilken process som ni använder för att uppnå god användbarhet?

JW: Ja det vill säga att, det är det som lite grann saknas. Vi har ingen insatt beställare som kanske så här, kan avgöra om vi gör rätt grejer till exempel. Vi har en operatör inblandad så att säga, för att ge input på förslag. Titta på gränssnittsprototyper och även processer om vi säger att "vi hade tänkt göra så här", "nä, det kan ni inte göra för att vi måste göra så här istället då". Så det, nu är inte det en, han är mer en superanvändare kan man säga, alltså en superoperatör om man säger så. I hans arbetsuppgifter normalt, så sett har han så att säga mappat vad dem olika fabrikerna gör, hur dem gör i en viss situation. För fabrikerna gör inte likadant till exempel. Så där är ju lite lurigt för oss. För vi kommer inte att veta om det här funkar riktigt förrän vi varit i ett antal fabriker egentligen. Man har en så stor frihet att köra maskinerna så att säga, på lite olika sätt. Det finns ingen manual till exempel, på vissa av maskinerna finns det ingen manual utan det är någon sorts, ja, vad heter det. Man får gå bredvid och lära sig ungefär, så får man skaffa sig en bild av hur det hela fungerar. Och då att vi ska plugga in den verkligheten är lite...

AR: Ja, det måste ju vara svårt, när det är olika verklighet för samma projekt liksom..

JW: Ja, precis. Nu kommer vi till en fabrik, vi är i den fasen där vi ska köra i rätt miljö. Och då kommer vi ju få en massa feedback på att det här är bra eller dåligt. Så just nu vet vi inte exakt, vi får simulera det mesta så att säga. Nu är det inte så hemskt mycket gränssnitt egentligen, utan den mesta koden är så att säga dold. Det är bara funktionalitet som bara ska fungera bakom. Men det är ett antal olika system som vi hämtar data ifrån och som operatörerna är vana vid att arbeta i. Så att vi vill inte bli, vi försöker inte bli ytterligare ett system, att man liksom slutar jobbar i något av dem här andra systemen, för och liksom det inte ska bli ett dugg bättre.

AR: Vem är det som bestämmer vad som är god användbarhet i era projekt?

JW: Det är ganska segt att få ihop, men tanken är att vi ska ha en referensgrupp som ska säga något innan årets slut. Denna fas som vi befinner oss i nu, där vi ska verifiera det här kontraktet, alltså kunna köra det kan man säga, genom en linje. Alltså en maskintyp av varje så att ungefär, så ska flöda igenom. Där ska det finnas en referensgrupp med lite olika intressenter.

DA: Är det några användare, eller slutanvändare då eller?

JW: Där får man ..(\*).. från så att säga mera affärssidan eller om man säger så.

AR: men dem här har inte varit med tidigare i, under utvecklingen då, sedan dem kom in på slutet.

JW: Alltså vi måste visa någonting för att dem ska förstå vad det är dem..

AR: jo, det är klar

JW: Så får man nog säga. Det är lite..(\*).. att vi inte har någon beställare som exakt vet vad vi vill ha så att säga. Så att vi måste agera båda sidor kan man säga.

DA: Det är framförallt det funktionella ni är ute efter? Inte Gränssnittet?

Alltså det är ganska lite gränssnitt, däremot får..processen är viktig. Men det är frågan... att han gör någonting vid ett visst tillfälle och får lov att göra vid ett visst tillfälle och såna grejer. Det är viktigt.

AR: Och det kan ju vara svårare att ge prototyper på det än ett gränssnitt?

JW: Ja, och där har vi en uppfattning om vad som är bra och dåligt och sådär. Alltså vi har en rullproduktion så att säga. Där man tillverkar en ny rulle på 10 min en kvart ungefär. Och så ska operatören mata in någonting i systemet som vi har samlat in under den tiden. Då ska han till exempel få lov att börja med det under tiden rullen håller på, och inte efter så att säga. För då är det ju en ny rulle som håller på att tillverkas. Så där håller vi på att

ställa krav på andra leverantörer också, då att det inte är tillräckligt att operatören får tillgång till data efter att rullen är färdig till exempel.. det är en sån grej där vi inser att så måste det vara. Sen är det egentligen domänkunskap och allt man liksom vet att, hur processen funkar och så.

AR: Det kanske ni vet ibland då bättre än dem som jobbar ute på fabriken och så?

JW: Ja, problemet är ju att mappa ihop det. Om vi vet att ”ok, det här är kanske någonting som är bra” men en leverantör klarar inte av att leverera i förhållande till det. Så är det ju svårt, från operatören får man ju då säga naturligtvis att ”så måste det vara”. Så det gäller ju, det är en balansgång där med vad vi vet vad som går att klara av så att säga, och vad vi tror att dem vill ha.

AR: Ja, det är intressant... har ni någon person som är ansvarig för användbarhet?

DA: Om det är någon som är utbildad för det?

JW: Nä, det kan jag inte säga. Mer än att jag är systemvetare, men typ så.

AR: Vilka verktyg använder ni för att uppnå god användbarhet?

JW: Där finns det ju egentligen inga riktiga verktyg, alltså i den meningen att vi kör någonting och testar och får ut något måttetal eller någonting, det finns det inte. Utan det kommer ju bli användarmedverkan på något sätt. Och .. alltså det finns övergripande måttetal på varför det här systemet införs. Om man ska lägga in det här i användbarhet, det kan man ju, det ska man kanske göra... det är nog affärssidan så att säga, bara för det här systemet ska existera.

AR: Det är mer av ekonomiska orsaker?

JW: Ja, precis, vi vill effektivisera en viss del av processen. Och uppnår vi det ur affärsdelens perspektiv så har vi uppnått användbarhet.

AR: Och gränsytan ut mot dem som sköter det är inte så avancerade att dem kräver så...?

JW: Nä, utan det är mer att det hänger ihop, att alla dem här systemen, att vi får fler som hänger ihop med alla andra system så att säga. Så operatörerna ser inte så mycket och...ja, det är ..(\*).. så att säga om man ska ha olika system. Vi har en person i teamet som är dediterad testare redan. Ja, det är lite så baklängesgrej. Vi som har insikt i vad vi borde göra har talat om för utvecklarna vad dem ska göra. Och sen så blir det bakvägen så att säga, att han identifierar vad koden uttrycker för krav och försöker formalisera dem kan man säga. Så det blir ju en viss tekniska specifikationer, men inte så där jätte.. dem är ganska hög.. fast han försöker göra dem mer detaljerade genom att titta. Vad är det koden säger att, vad är det för krav koden uppfyller. Och så översätter och gör dem mer detaljerade. Och så får jag liksom ”okey:a” , ”ja, det är rätt uppfattat” eller ”nej, det är inte riktigt svar på detaljnivå som jag hade tänkt”.

DA: Följer ni upp det ni utvecklar? Men det är kanske lite av det du beskrev precis..

JW: Ja, alltså får vi ju massa... när det gäller själva kodens kvalitet så har vi ju verktyg och processer. Frågan om vi får in underhållsbarhet och lite andra såna grejer, så försöker vi täcka in det med verktyg så att säga. Alltså hur många enhetstester har vi, täcker enhetstesterna allting, när standard är strikt. Skriver man någonting som bryter mot en namngivningsregel så går inte bygget igenom till exempel. Vi har liksom inte "warning" på det vi har "error" på namngivning och sånt. Då får han en enhetlighet. Så vi vet så att säga att någon skrev den här koden under en ganska lång tid efteråt så att säga.

AR: Vi har ju då ISO 9241 som ramverk. Har du hört talas om det?

JW: Nej.

AR: Då antar jag att ni inte använder er av den heller?

JW: Nä, det är mer på något sätt "sunt förnuft"-principen där. Någon sorts domänkunskap, att det här funkar nog, det här funkar nog inte.

AR: Kravspecifikationen har vi varit inne lite grann på. Men hur ser den ut?

JW: Som sagt ganska luddiga och ganska mycket muntligt utfyllda så att säga. Och sen med lite efterklokhet börjar vi mappa dem i efterhand då kan man ju säga.

DA: vem är det som ställer kraven?

JW: Alltså som sagt vi har ju ingen bra kravställare. Så vi är ju både kravställare..så vi sitter på båda sidor hela tiden liksom. Det finns ingen vi kan lämna över den till. Alltså det finns ingen, det är inte utsedd någon förvaltningsorganisation än. Så det är också ett problem liksom. Då hade man ju också behövt säga att "för att det här ska överleva över tio år så måste ni göra detta också". Men det måste vi i så fall också gissa just nu.

AR: hitta på helt enkelt och ändra?

JW: Ja, precis.

DA: det gäller att ha lite fantasi också?

JW: Ja, och då missar man ju säkert någonting också. Och frågan är ju då vad det är.

AR: hur undersöker ni användningskontexten?

JW: Dels är det ju tillsammans med leverantörerna med dem här externa systemen då. Varav ett stort system är internt Tetra Pak också, alltså det är

produktionsplaneringssystemet kan man säga. Och där är en svårighet att dem har en mycket långsammare process. Så vi har fått göra vårt system där vi till och med fått skapa en modul som simulerar deras system. För att dem levererar långt senare. Och där vi nu då, där vi på något sätt är nästan klara, så är dem inne i sin definitionsfas kan man säga, att dem ska bestämma vad dem ska göra. Men då är det ju vi som måste tala om att, det är ungefär det här ni ska göra annars så passar det inte. Eller så får vi göra om alltihopa och det vill vi inte. Och sen har vi dem här leverantörerna av bra system som måste direkt ut i produktionen. Sen så har vi varit ute och tittat i den fabrik som ska pilotfabriken..

AR: ja, det här har vi också varit inne på lite, men Identifierar ni användbarhetskriterier?

JW: Alltså, inte mer än dem så att säga övergripande kommersiella sakerna som vi har. Som jag inte kan avslöja. Men där finns det mätetal. X så av Y annars så har vi inte så mycket. Men det är extremt grovt. För att alla dem andra talen, ja mätetalen man skulle kunna ha på hur operatörerna upplever systemet, för dem är egentligen bara drabbade av det då, kan man säga. Det är inte där som det går ut på att göra bättre egentligen, utan det är effektiviteten i maskinen. Där är det mer upp till oss då bara att vi vet att någonting.. om vi gör på ett visst sätt så blir det ganska kraftfullt. Men inga mätetal.

DA: Det är alltså utökad effektivitet ni söker med systemen?

JW: Ja. kronor och ören har en dragande kraft.

AR: Och operatörerna är ju inte, dem påverkar inte effektiviteten, alltså deras...

JW: Dem, alltså det är för att kunna gå ett steg vidare. Genom utbildning och sånt så försöker man hela tiden att hitta bättre sätt att göra det eller vad som helst. Men detta är liksom en stegförändring.

AR: interaktionen mellan operatören och systemet har inte så jättestor betydelsen för effektiviteten?

JW: Nej, utan vi måste göra ett bra jobb tekniskt. Då kan operatörerna, ja, då får man nyttan ganska enkelt genom att utbilda operatörerna på att vi, göra på något visst sätt så att säga. Men det är inte så mycket att dem ska välja på att göra si eller så. Utan det är en viss formalitet.

AR: När det gäller Scrum, ni har inte problemet som vi har sett med användbarhet, att det är svårt att skapa ett bra användbart system efter som man inte kan se helheten?

JW: Vi vet ju var det ska passa in så att säga. Då kan ju vi ha lite mer så..(\*)..

DA: Det är väl så att genom att ni har mer interna projekt är det lättare att se hur det ska se ut?

JW: Ja, och sen är det fördelar och nackdelar. Hade vi haft en extern leverantör så hade

dem ställt krav på oss att det går inte att utveckla om inte ni bestämmer er för vad ni vill ha. Så i och med att det är internt nu, så kan vi liksom jobba ganska, ja vi kan jobba framåt så att säga utan att ha allting spikat.

AR: Scrum funkar bra hos er?

JW: Ja, Scrum funkar bra.

AR: Inga andra problem som ni tidigare stött på?

JW: Ja, för det är rätt mycket grejer som inte är så definierat, just runt kraven. Jag menar, det finns ju, det finns väl ett antal krav som bör vara satta från början så att säga. Men var den balansen ligger så att säga, det är liksom en sån. Det finns ju fortfarande folk som hävdar att man ska köra vattenfall om man har, vet man vad man ska göra behöver man inte köra Scrum ungefär. Frågan är då om man får ut någonting av Scrum om man ändå har bra utgångskrav då, det är frågan. Ja, det vet inte jag heller, men det förs en intressant debatt om det.

AR: Men det förändras inte så mycket under tidens gång, så att ni behöver liksom anpassa er efter förändringar och utveckling?

JW: Alltså vi kommer ju att få att saker och ting inte är exakt, eller kommer funka som vi har tänkt från början. Och jag menar, jag för ju diskussioner med det här interna bolaget också då ju, just nu, som påverkar. Och där kommer ju bli ett problem att vi måste ju ha två spår levande. För vi har ju utvecklat någonting med den här simulator modulen då, och det ska ju testas i fabrik. Samtidigt som då ska hålla på och göra förändringar i koden för att passa det riktiga systemet. Och den koden kommer inte att kunna användas förrän i liksom slutet av det här året.

DA: Hur länge pågår ett sånt här projekt ungefär?

JW: Nu är det här ganska unikt så att säga. Bara själva, alltså den ordentliga utvecklingen är väl ungefär ett halvår drygt. Och det kommer ju fortgå ytterligare ett halvår kan man säga. Så det är, ja ett drygt år ungefär. Och då är det ändå bara pilotdelen så att säga, sen så kanske.. Sen kommer det bli förändringar och burn out så att säga. Då kommer det säkert att bli...men det blir väl lite mer förvaltnings extra funktioner och justering.

DA: Ja, jag vet inte om det var något mer. Vi har inte fler frågor på dagordningen.

## Bilaga 5 - Intervju 3

*Intervju med Michael Lindberg på Ninetech.*

---

I texten visas (\*) vid otydlig inspelning.

AR: Vilket företag är du verksam på?

ML: Ninetech.

AR: Vad är dina ansvarsområden?

ML: Ja, utvecklare, konsult och Scrum-master. Och produktägare.

AR: Och vad innebär det att du har för arbetsuppgifter?

ML: Ja, allt ifrån programmering till styrning av projekt.

AR: Vad är det för typ av uppdrag ni arbetar med?

ML: Vi har ju dels en egen produkt som vi arbetar med och utvecklar. Ofta tillsammans med kunder. Och i vissa delar även finansierat av våra kunder. Så det är väl det jag jobbar med.

DA: Är det större kunder eller mindre kunder ni har?

ML: Det är blandat. Från små kunder till ganska stora.

DA: Vilken typ av produkter är det då ni utvecklar?

ML: Den produkten som vi jobbar med här som vi utvecklar är PrimeCase som är en produkt för processhantering. Sen har vi också inom Ninetech utveckling specifikt mot kund.

AR: Hur många är ni i ett utvecklingsteam?

ML: Nu är vi tre stycken. Optimalt hade varit att vara fler, men vi har inte ekonomisk teckning för det eller arbete för det.

AR: Hur ser en utvecklingsprocess ut hos er?

ML: Det är lite blandat, en del kommer från kund direkt och en del står vi för och så vidare. Vi är ju ändå ett konsultbolag egentligen. Vi jobbar med lite olika produkter. Men vi försöker ändå samla ihop allting. Så vi samlar sånt som ska utvecklas mot kund och sånt som vi ska stå för i sprintar. Så att vi har...

AR: Så ni har flera projekt parallellt?

ML: Ja, men allting går inte in i sprintar. Vissa saker får vi köra utanför. Saker där vi har för mycket externa beroenden. Om det är mycket integrationsarbete att jobba mot andra system och andra leverantörer ibland då går det inte. För att det blir för mycket väntan fram och tillbaka. Och då får det ligga utanför. Men om det går så vill vi ha det i sprintar så vi jobbar likadant hela tiden.

DA: Ja , det är väl det ultimata egentligen?

ML: Ja, för att liksom få ett förbättringsarbete och kvalitet på det man gör så. Så vill man ju gärna att det ska rulla på likadant så gott det går.

AR: Vad betyder god användbarhet för dig?

ML: Det är ju funktioner som ska vara lätta att förstå. Helst ska man inte behöva läsa manualen när man använder sig av funktionen. Nu är det kanske inte alltid möjligt men det är ändå det som är målet. Det ska vara lätt att förstå.

DA: Självlärande kan man säga då?

ML: Ja.

AR: Har ni några processer för att uppnå god användbarhet i era projekt?

ML: Vi har nog inga speciella processer för det. Men vi, eller i alla fall jag är mycket ute hos kund och så. Och håller även utbildningar i det här systemet så man får ju ganska bra koppling tillbaka vad som fungerar och inte fungerar. Och det tror jag är ganska bra. Att i alla fall några av utvecklarna i ett team har den kopplingen och den erfarenheten. Så att inte alla bara sitter inne på kontoret alltid och inte ser det användas. För då är det inte alltid det blir rätt.

AR: Vem är det som bestämmer vad som är god användbarhet?

ML: Bestämmer? Det är någonting vi försöker diskutera egentligen när vi utvärdera vad som ska göras. Hur vi vill att det ska fungera och även fast man inte har allt nedskrivet eller "specat" i detalj så försöker man ändå ha vissa riktlinjer för var en funktion ska finnas och hur det är tänkt att fungera.



DA: Har ni några diskussioner med slutanvändarna också eller är det produktägarna?

ML: Det beror ju på också vem som finansierar det. Om det är en kund som ska finansiera det så får vi lämna en detaljerad beskrivning på hur det är tänkt att fungera. I vissa fall kanske man bara lyfter luren och pratar med någon kund som man vet skulle ha nytta av funktionen.

AR: Använder ni några verktyg för att uppnå god användbarhet?

ML: Nej det har vi inte.

DA: Ni går på sunt förnuft?

ML: Ja, det gör vi.

AR: Hur stora system är det ni utvecklar?

ML: Mätt i vadå?

AR: Ja det är ju frågan...

ML: Ja men PrimeCase som vi jobbar med här. Det är ju ändå några år som vi har utvecklat det. Det är ju en tre-fyra år i alla fall.

AR: Men det är ändå stora system med mycket funktionalitet och många användare?

ML: Ja det är det ju. Men vi har ju många små projekt också som inte rör den här produkten. Men då är det svårt att få in Scrum. När det inte är samma produkter, samma system man jobbar med.

DA: Ni har inte bara Scrum då utan ni använder andra metoder också?

ML: Ja, det beror på vilket projekt det är, men Scrum är det vi föredrar. Men det kräver ändå en viss storlek på projektet. Om det bara är två veckors arbete så kommer man inte igång riktigt.

AR: Vi har fått uppfattningen att när det gäller väldigt stora projekt så är det svårt att jobba med Scrum...

ML: Eftersom vi jobbar med en egen produkt så släpper vi ju rätt ofta nya releaser. Då blir det ju att man har inte hela bilden framför sig. Men jag kan tänka mig att det kan vara rätt så svårt. Det beror på också vilken typ av kund man har. Vad man har för motparter och hur pass involverade de vill vara. Det kräver ganska mycket feedback och engagemang.

AR: ..(\*)..

ML: Ja det vi gör för kunder också, det gör vi som standardfunktioner i system så mycket som möjligt. Så det ju något som vi sen förvaltar. Så det får vi ju feedback på.

AR: Ni gör inga tester eller så?

ML: Användartester eller så? Nej det har vi inte.

Vi har ju kanske 20 kunder på den här produkten, ja kanske 30. Så vi har ju inte så jättestora volymer, då är det svårt att ha...

DA: Användbarhet är ju det som gör det effektivt i slutänden.

ML: Visst är det så. Men det beror också på vilka volymer man har. I början är det mest effektivt att bara svara på frågorna direkt, svara i telefon. Men ju större man blir och ju fler kunder man får ju mer värt blir det att få in den typen av tester.

AR: Vi har använt oss av ISO9241 i vår uppsats. Har du hört talas om den?

ML: Det har jag säkert.

AR: (beskriver kort ISO standarden) Det är inget ni använder er av?

ML: Nej.

AR: Okej, era kravspecifikationer då, hur ser de ut?

ML: Det beror på vad det är för funktionalitet. Det är allt ifrån bara några rader text egentligen, till offerter.

DA: Och det är kunderna i huvudsak som...

ML: Ja, det beror på vad det ska tjäna för syfte. Vi försöker skriva på en lagom nivå. Det vi behöver för att kunna utveckla. För att vi som jobbar här ska vara överens om vad som menas och vad det betyder. Men ibland när det finns andra parter får man lägga det på en annan nivå. För dem som inte har den bakgrundskunskapen och så vidare som vi har.

AR: Så ni gör inte själva undersökningar av användningskontexten, användarna och miljön?

ML: Nä alltså, de flesta funktioner vi gör, det rör sig mellan en dag och 5 dagars arbete. Det är inte helt nytt system varje gång. Det är oftast mindre delar.

AR: Och då vet man redan var det ska vara och hur det ska fungera?

ML: Ja, ofta är det komplement till befintlig funktionalitet också.

DA: Är det sällan ni tar fram ett helt nytt system?

ML: Ja, i alla fall enligt det här sättet. På kontoret i Karlstad har vi lite andra typer av projekt. Lite större e-handelsprojekt och så där. Men där har inte vi varit direkt inblandade.

AR: Vi har redan varit inne på det lite, men identifierar ni mätbara användbarhetskriterier?

ML: Nej.

AR: Och ni gör inga tester heller?

ML: Nej, inte på användbarhetsnivå på det sättet.

AR: Det känns inte som att ni har behov av det?

DA: Händer det att ni får göra om?

ML: Vi kompletterar ju ganska så löpande saker som kanske inte fungerar som vi har tänkt oss.

AR: Hur får ni reda på det då?

ML: Det kan vara när vi är ute hos kunder. Eller man märker man får frågor till supporten på vissa saker. Ganska ofta försöker vi snappa upp sånt som vi kan justera och ändra i så fall.

AR: Så man kan säga att det får testas i verkligheten så får ni feedback sen.

ML: Ja vi testar så gott vi kan här men vi har inte, med tanke på storlek och volymer och sånt...

AR: Det finns inget behov?

ML: Det finns inte behov och det finns inte möjlighet till det. Vi försöker föra en dialog hela tiden med våra kunder och se vad det är de behöver och vill ha. Det märker vi också. Vi är ofta ute hos kunder och utbildar och hjälper dem så känner vi av det ganska bra.

(Härifrån är det en del diskussioner som inte är helt relevanta för vår uppsats, bara sådant som kan vara intressant kommer att redovisas här)

DA: Hur har ni det med tider? Kan ni sätta tider när ett projekt ska bli klart eller blir det efter hand?

ML: Vi har inte så jättelång framförhållning, det ligger på någon månad sådär. Och som jag sa, vi har inte så stora funktioner eller så. Vi har redan en färdig produkt som går att använda så det underlättar ju ganska mycket. Det är inte något som behöver byggas från grunden. Vi har inte kört Scrum så länge, det tar ett tag innan det lossnar ordentligt.

AR: Vad är det för fördelar och nackdelar som ni har sett?

ML: Fördelen är att i och med att vi är så små så måste vi vara väldigt flexibla. Vi kan inte sitta i tre månader och göra nått som är bestämt utan vi måste kunna styra om ganska fort. Vi kanske är på väg att få ett kontrakt om vi bara kan få in någon funktion till som vi saknar. Då är det ganska bråttom. Tidigare var det så fram och tillbaka hela tiden när det kom nya krav. Nu kan vi i alla fall sitta i två veckor när vi kör ett Scrumtänk.

Vissa saker tror jag inte går att få in i Scrum heller.

DA: Vilka delar tänker du på då?

ML: Vissa saker, som det här med gränssnitt och så måste finnas på plats. Det kan man inte komma ifrån. Innan ett stort Scrumprojekt kan komma igång måste det finnas en plan i alla fall. Det måste finnas en vision eller en tanke. Sen kanske inte alla detaljer måste finnas på plats men det är ju inte bara att sätta sig ner och dra igång. Utan det måste finnas lite mer arbete innan.

AR: Ja, men ofta får ni det utifrån då, när det inte gäller ert eget? Kunden har redan bestämt...

ML: Ja, men sen tittar ju vi på det. Ofta kanske vi kompletterar eller ger andra förslag eftersom vi så mycket som möjligt vill kunna använda samma funktionalitet till andra kunder. Då får vi ju betalt fler gånger för samma sak då. Därför tittar vi på hur vi skulle kunna lösa det på ett annat sätt om vi tycker att det blir bättre. Det kostar ju enormt mycket att utveckla programvara. Det går åt många timmar.

AR: Har ni en så slutfas i slutet av projektet där ni går igenom och kollar systemet...

ML: I och med att vi jobbar med det här hela tiden så har vi vissa saker som vi vet att det här vill vi göra. Och är man inne någonstans då så ändrar man. Om man jobbar så långvarigt med en produkt som vi gör, i många år. Då jobbar man med det hela tiden.

DA: Har ni några fasta roller i projekten?

ML: Nej.

DA: Ni gör allt tillsammans?

ML: Ja, det mesta i alla fall. Det finns ju de som är bättre på vissa saker, och då försöker vi lägga det på dem, men det finns inget som är sagt. Det finns väl ibland vissa önskemål från ledningshåll att vissa personer ska göra vissa saker, men det brukar vi strunta i.

AR: Ja, ni är ju bara tre också. Då kanske det är svårt att ha specifika roller.

ML: Ja om man ska jobba enligt Scrum så är det ganska viktigt att alla kan allt. Sen kan man ha vissa experter, och att man kan olika saker olika bra, men man måste nog kunna ta i lite av varje. Annars kommer det alltid bli så att någon får sitta och vänta på andra.

AR: Hmm, ja det är kanske inte så mycket mer att fråga om användbarhet. Ni jobbar ju inte så strukturerat med det.

ML: Nej, det vi gör att försöka tänka på det. Vi försöker förklara funktionerna och hur de används.

AR: Ni jobbar med utbildning också?

ML: Ja

DA: Mot slutanvändarna?

ML: Ja. Vi har också en manual och ett servicedesk som tar emot samtal.

DA: Något övrigt?

ML: Näe, det är väl möjligt att det här kommer ändras. Om vi blir större och man får lite mer nytta ut av den tid man lägger ner. Det är ändå det som styr i slutändan. Det måste vara ekonomiskt försvarbart. Om man har större volym så kanske man har igen de timmarna man får lägga på användbarhetstester m.m. på att man får mindre support. Men sen tror jag att ibland kan det prioriteras för lågt också, för om det är många småsaker som inte fungerar riktigt som man vill så kan det dra ner helhetsintrycket ganska mycket.

## Litteraturförteckning

- Alpide, I., Karlsson, E. (2002): *Ett IT-verktygs användbarhet inom hemtjänsten – en utvärdering av IMIS*. Blekinge Institute of Technology.
- Allwood, CM. (1998): *Människa – datorinteraktion. Ett psykologiskt perspektiv*. Studentlitteratur, Lund.
- Aston, J., Meszaros, G. (2006): *Adding Usability Testing to an Agile Project*. Proceedings of Agile 2006 Conference (AGILE`06)
- Backman, J. (1998): *Rapport och uppsatser*. Studentlitteratur, Lund.
- Constantine, L. L. (2001): *Process Agility and Software Usability: Toward Lightweight Usage-Centered Design*. Constantine and Lockwood, Ltd, Sydney.
- Constantine, L. L., Lockwood, L. A. D. (2003): *Usage-Centered Software Engineering: An Agile Approach to Integrating Users, User Interfaces, and Usability into Software Engineering Practice*. Proceedings of the 25<sup>th</sup> International Conference on Software Engineering (ICSE`03)
- Edling Andersson, M. (2004): *MDI i mobiltelefoner – en komparativ studie av tre gränssnitt*. Institutionen för informatik, Lunds Universitet.
- Gulliksen, J., Göransson, B. (2002): *Användbarhetscentrerad systemdesign*. Studentlitteratur, Lund.
- Gustavsson, T. (2007): *Agile: Konsten att slutföra projekt*. TUK Förlag, Karlstad.
- ISO 9241 (1998): *Ergonomic Requirements for Office Work with Visual Display Terminals, Part 11, Guidance on specifying and measuring usability*, International Standard.
- Jacobsen D.I. (2002): *Vad, hur och varför? Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen*. Studentlitteratur, Lund
- Kane, D. (2003): *Finding a Place for Discount Usability Engineering in Agile Development: Throwing Down the Gauntlet*. Proceedings of the Agile Development Conference (ADC`03)

Kvale, S. (1997). *Den kvalitativa forskningsintervjun*. Lund : Studentlitteratur.

Lee, J. C., McCrickard, D. S. (2007): *Towards Extreme(ly) Usable Software: Exploring Tensions Between Usability and Agile Software Development*. Center for Human-Computer Interaction, Department of Computer Science Virginia Tech, Blacksburg.

Moe, N. B., Dingsøy, T., Dybå, T. (2008): *Understanding Self-organizing Teams in Agile Software Development*. SINTEF ICT, Norge.

Nilsson, M., Nilsson, C. (2004): *Installationsgränssnitt under Linux och Windows XP utifrån ISO 9241*. Blekinge Institute of Technology.

Patel, R., Tebelius, U. (1987): *Grundbok I forskningsmetodik*. Studentlitteratur, Lund.

Schwaber, K. (2002): *Agile Software Development with SCRUM*. Prentice Hall, London.

Seffah, A., Gulliksen, J., Desmarais, M. (2005): *Human-Centered Software Engineering – Integrating Usability In The Development Process*. Springer, Dordrecht, The Netherlands.

Singh, M. (2008): *U-SCRUM: An Agile Methodology for Promoting Usability*. ChannelAdvisor Corporation, USA.

Thurén, T. (2005): *Källkritik*. Liber, Stockholm.

Travis, D. (2004): *Bluffer's Guide to ISO 9241*. Userfocus Ltd, London.