



GÖTEBORGS UNIVERSITET

Design av komplex “Software as a Service” som self-service

En fallstudie i användarbehov och designprinciper

Designing complex “Software as a Service” as self-service

A case study in user needs and design principles

DANIEL ALTEBORG
CHRISTIAN RENLUND

Kandidatuppsats i Informatik

Rapport nr. 2012:042
ISSN: 1651-4769

Abstrakt

Utvecklingen inom informationsteknologi (IT)-området har gjort det möjligt för företag att distribuera tjänster och mjukvara via webben, enligt affärsmodellen Software as a Service (SaaS). Det har dessutom visat sig att det finns ytterligare vinst att göra för såväl konsumenter som leverantörer av dessa verktyg om de kan distribueras utan omfattande implementeringsaktiviteter och utbildning. Därför vill man uppnå en hög grad av self-service i applikationerna.

Affärsprocessföretaget Barium AB har uppmärksammat denna potential hos SaaS-produkterna och hyser en önskan att implementera self-servicetänket i deras produkt ”Barium Live!”. Då den forskning som finns rörande design av self-serviceprodukter i huvudsak handlar om enkla system av kiosk- och automattyp har de tagit initiativ till detta arbete för att reda ut vilka designimplikationer self-serviceaspekten har för mer komplexa applikationer. Detta utgör syftet för vår undersökning och leder fram till vår frågeställning; *Vilka designprinciper är av särskild relevans vid design av komplexa SaaS-self-serviceapplikationer?*

För att besvara frågeställningen gjordes observationer och intervjuer med användare från Bariums kundkrets som för första gången använde ”Barium Live!”. Utifrån framtagna scenarion fick de använda produktens olika delar. De upptäckter som gjordes i detta ställdes mot existerande designlitteratur och resulterade i ett ramverk med designprinciper som utgör studiens huvudsakliga resultat.

De huvudsakliga slutsatser som framkommer är att ”konsekvens” och ”tydlighet” är nyckelord för designen och äger stor tyngd för att self-servicetänket ska kunna implementeras samt att hjälpfunktionalitet spelar en central roll. Dessa måste finnas i flera nivåer men framförallt dynamiskt och avgränsat för de specifika moment som för tillfället utförs i applikationen.

Rapporten är skriven på svenska.

Nyckelord: Software as a Service, SaaS, self-service, design, designprinciper, Barium, Barium Live!

Abstract

Advances in the information technology (IT) area have enabled companies to deliver software and services through the web in a business model known as Software as a Service (SaaS). In addition to this, further profits have shown to be made for both production and consuming companies by delivering and implementing this software without extensive training activities. Therefore a high self-service level is desirable for the applications.

The business process company Barium AB has acknowledged this potential in SaaS and holds a desire to implement the self-service aspects in their product "Barium Live!". While the existing research in self-service design is mainly focused on kiosk-based technology with straight forward softwares Barium has taken the initiative to this study to explore the design impacts of self-service on more complex softwares such as their own. This constitutes the purpose of this study and results in the following research question; *Which design principles are of particular relevance in design of complex SaaS-self-service software?*

To answer the research question we conducted observations and interviews with users from Barium's customer base who hadn't yet been in contact with the product. They were given tailored scenarios which made them try various aspects of the product with a self-service approach but in the current, non self-service centered, design. The findings from these sessions were contrasted with state-of-the-art design publications and this resulted in a framework of design principles with particular relevance for design of complex SaaS-self-service software, which is the main result of the study.

The key findings from the study are that "consistency" and "lucidity" are key phrases to guide the design and that they are of great importance to be able to implement the self-service aspects in the software. In addition to this help-functions play a key role to come to terms with the complexity of the product and these functions need to be offered at different levels, but most importantly with a dynamical presence in each and every task the user has at hand.

The report is written in Swedish.

Keywords: Software as a Service, SaaS, self-service, design, design principles, Barium, Barium Live!

TACK

Vi vill tacka Dana Markovic och Jesper Palm på Barium AB för initiativ till och fokusering av detta arbete. Tack också för hjälp med kontakter till lämpliga informanter.

Dessa informanter vill vi också rikta tack till för att ni tog av er tid och kom med ovärderlig input till denna uppsats resultat och slutsatser.

Slutligen vill vi tacka vår handledare, Agneta Ranerup, för effektiv feedback och uppmuntring samt examinatorn Lisen Selander för tips på sista-minuten-korrigeringar.

Innehåll

1. Introduktion.....	1
1.1 Bakgrund.....	1
1.2 Problem.....	1
1.3 Syfte och frågeställning.....	2
1.4 Definition och avgränsning.....	2
1.5 Undersökningens upplägg.....	2
2. Teori.....	4
2.1 Software as a service som affärsmodell.....	4
2.2 Self-service.....	5
2.3 Designteori.....	6
2.3.1 Bakomliggande teorier.....	6
2.3.2 Design av komplexa self-serviceprodukter idag.....	8
2.4 Teorins roll i studien.....	14
3. Fallstudieobjekt; ”Barium Live!”.....	15
3.1 Vad är ”Barium Live!”?.....	15
3.2 Funktionaliteten i ”Barium Live!”.....	15
3.2.1 Startside.....	16
3.2.2 Navigera mellan delarna.....	17
3.2.3 Modellera.....	18
3.2.4 Bygg.....	21
3.2.5 Köra processer.....	22
3.2.6 Mina aktivitetslistor.....	23
3.2.7 Uppföljning.....	24
4. Metod.....	25
4.1 Observationer.....	25
4.2 Intervjuer.....	26
4.3 Urval.....	26
4.3.1 Presentation av urvalsgruppen.....	27
5. Resultat.....	29
5.1 Layout.....	29
5.2 Navigering.....	33
5.3 Hjälp.....	35
6. Resultatanalys.....	39

6.1 Layout.....	39
6.2 Navigering	41
6.3 Hjälp	42
7. Slutsatser.....	46
7.1 Studiens relevans och överförbarhet	46
7.2 Förslag till fortsatt forskning	47
Källor.....	48

- Bilaga 1 – Användbarhetstestmanus
- Bilaga 2 – Användbarhetstestuppgifter
- Bilaga 3 – Inspelningsmedgivande
- Bilaga 4 – Intervjufrågor

1. Introduktion

Nedan ges en bakgrund till det undersökta problemområdet samt en presentation av den problematik som vi baserar vår frågeställning på. Sist i kapitlet presenterar vi relevanta definitioner och avgränsningar för studien samt en översikt över dess upplägg.

1.1 Bakgrund

Utvecklingen inom informationsteknologi (IT)-området har gjort det möjligt för företag att distribuera tjänster och mjukvara via webben, enligt en affärsmodell känd som Software as a Service (SaaS). Utvecklingen av dessa tjänster har på senare tid ökat i takt med att de tekniska förutsättningarna och intresset för dem stigit bland företagen. Detta innebär att konsumenterna i allt större utsträckning undviker en lokal installation av produkter med hänsyn till den driftsbörda det kan innebära och istället enligt en licensmodell köper produkten som en tjänst (Papazoglou & Ribbers, 2006; Dubey & Wagle, 2007; Verma, 2011).

Både företag som säljer dessa tjänster och dess kunder kan uppleva fördelar av webbaserade tjänster. Förtjänsterna för distribuerande företag innebär bl.a. att nå ut till fler potentiella kunder, slippa anpassningar av produkten och få en större kontroll över de tjänster man levererar. Konsumenterna slipper i sin tur, som nämnts ovan, att drifta hårdvara själva och kan anpassa sin systemflora efter tillfälliga behov vilket innebär att de inte behöver låsa sig vid ett alternativ (Papazoglou & Ribbers, 2006; Dubey & Wagle, 2007; Verma, 2011). Dessutom finns ytterligare vinst att göra för konsumenterna av dessa verktyg om de kan distribueras utan omfattande implementeringsaktiviteter och utbildning, då det gör att värdet av verktygen snabbare kan tas till vara på. Man vill uppnå en grad av "self-service", låta kunderna klara sig själva, i produkterna (Dubey & Wagle, 2007). Själva essensen av ett self-serviceverktyg är att det ska vara utformat så att dess användare snabbt kan förstå och nyttja det på egen hand och på så vis minimera behovet av utbildning (Avery, 2009).

En annan viktig aspekt av detta är att även om produktens funktionalitet är fullgod kommer dess positiva effekter aldrig utnyttjas om det inte finns en god användaracceptans för produkten (Vella, 2011). Genom att minska ansträngningen för användare att lära sig och kontinuerligt använda en produkt lägger man grunden till en god användaracceptans, vilket avgör om produkten används och i vilken utsträckning dess potential utnyttjas, eller om den rent utav riskerar att bytas ut (ibid.).

1.2 Problem

Idag finns mycket forskning kring self-service och SaaS, men denna behandlar främst varför det är eftersträvt att anskaffa dessa produkter, dess konkurrenskraft och hur man för in dem i organisationer (se t.ex. Dabholkar, 1994; Ladik 1999; Anselmsson, 2001; Wang & Namen, 2004). Mindre forskning behandlar själva designen av produkterna. I den mån det finns någon sådan forskning fokuserar den på produkter av enklare karaktär med grund funktionalitet som t.ex.

beställningsformulär på hemsidor eller system som bankomater eller biljettuthämtningskiosker (se t.ex. Avery, 2009; Wong et al, 2010).

Att designa webb-baserade SaaS produkter ställer höga krav på utvecklarna, inte minst när man levererar en komplex produkt med mycket funktionalitet (Krug, 2006; Tidwell, 2011). Med self-serviceaspekten i tillägg blir förstås designövervägandena fler och utvecklingsansvaret än tyngre.

1.3 Syfte och frågeställning

Vi vill i vår uppsats undersöka hur man kan och bör designa *komplexa* SaaS-applikationer för att de i så stor utsträckning som möjligt skall kunna fungera enligt en self-serviceprincip, på ett sätt som tillfredsställer användarna så att en god grund för användaracceptans skapas.

Utifrån denna bakgrund tar vi en normativ, förståelseinriktad ansats och kan formulera vår frågeställning:

Vilka designprinciper är av särskild relevans vid design av komplexa SaaS-self-serviceapplikationer?

1.4 Definition och avgränsning

För att illustrera kombinationen av SaaS och self-service som vi fokuserar på i denna studie använder vi i följande framställning uttrycket ”SaaS-self-service” med varierande suffix.

Definition och innebörd av SaaS respektive self-service diskuteras genomgående i teorikapitlet (kap. 2).

Med ”komplex” menar vi i denna studie avancerade produkter med bred och djup funktionalitet, i kontrast till traditionella self-serviceprodukter som har smalt användningsområde och grund funktionalitet (se avsnitt 2.2).

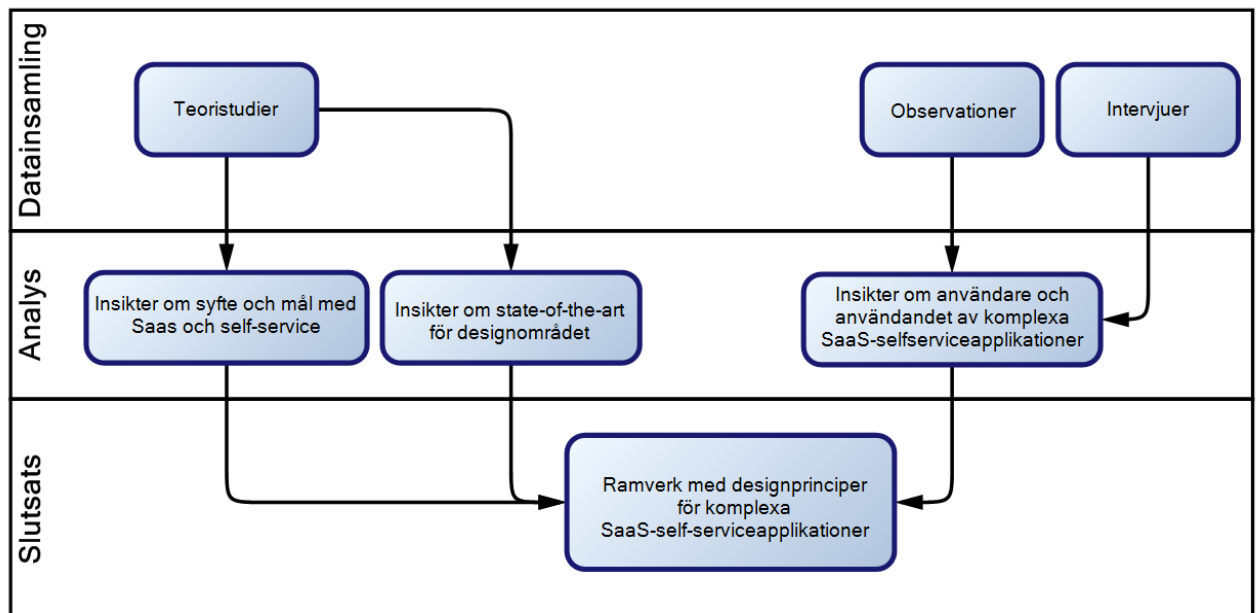
När vi i denna uppsats diskuterar design fokuserar vi hela tiden på applikationer som, i enlighet med SaaS-modellen, är åtkomliga via webben, med en dator, tangentbord samt mus som huvudsaklig hårdvara.

Produkten ”Barium Live!”, som är vårt fallstudieobjekt och som presenteras utförligt i kapitel 3, är ett typexempel på vad vi menar med komplex SaaS-self-serviceprodukt som används på detta sätt.

1.5 Undersökningens upplägg

Figur 1.1 visar översiktligt hur vi ser den här studiens upplägg. Teoristudier kommer att redovisas i kapitel 2 och förväntas ge insikter om vad som är syftet med att distribuera system enligt en SaaS-self-servicemodell. Vad dessa begrepp verkligen betyder, hur man ser på design av den aktuella typen av produkter idag samt vilka existerande designrekommendationer som är applicerbara på dem är andra insikter som eftersträvas i kapitlet.

I studien kommer observationer och intervjuer att göras med användare av en specifik produkt av den komplexa karaktär vi ovan beskrivit. Detta tillsammans hoppas vi ska ge oss insikt i hur användare upplever den här typen av produkter när de utan träning, på egen hand, ska utföra ett antal uppgifter i en produkt de inte är vana vid. Därigenom hoppas vi kunna dra generella slutsatser för hur produkter av den här typen bör utformas för att underlätta för användarna, vid den användningssituation som kännetecknar SaaS-self-servicemodellen. Den studerade produkten och dess funktionalitet presenteras i kapitel 3 och hur dessa moment förberetts och genomförts beskrivs i kapitel 4. Resultaten av datainsamlingen presenteras i kapitel 5. Utifrån dessa insamlade insikter för vi sedan ett analyserande resonemang i kapitel 6 som resulterar i ett ramverk innehållande de specifika designprinciper som vi identifierat som särskilt viktiga för att kunna göra en lyckad design av en komplex SaaS-self-serviceprodukt. I kapitel 7 presenteras sedan slutligen studiens slutsatser med huvudpoänger från det framkomna resultatet samt studiens begränsningar och uppslag till vidare forskning.



Figur 1.1 Skiss över studiens upplägg.



2. Teori

För att förstå problemområdet som diskussionen i denna uppsats kommer att röra sig i finns det en rad begrepp som måste redas ut och definieras. Affärsmodellen med SaaS-produkter, vår anammade tolkning av self-servicebegreppet samt vad som skrivs om design idag är de mest centrala aspekterna och det är teorin kring dessa vi går igenom i detta kapitel. Fokus i vår framställning kommer att ligga på, förutom begreppsdefinitionen, vad i dessa olika modeller (SaaS och self-service) som kan ha inverkan på designen av produkter för området. I designavsnittet (avsnitt 2.3) lyfter vi fram de delar av designforskningen som vi ser har tyngst betydelse för just SaaS-/self-serviceprodukter.

Litteraturen är vald utifrån omfattande sökningar i olika typer av artikeldatabaser. Sökorden formulerades utifrån våra huvudsakliga teoriområden SaaS, self-service och gränssnittsdesign. I tillägg till detta har vi utgått från erkända verk rörande design som presenterats av universitetet samt produktleverantören som initierat undersökningen (se kapitel 3). I nästa steg har vi gått igenom källförteckningarna i de funna verken för att fördjupa oss inom nödvändiga områden.

Teorikapitlet tillåts att ta relativt mycket utrymme i denna uppsats då, som vi visat i figur 1.1, teoristudierna är av stor betydelse för vår undersökning och kommer att utgöra ett viktigt underlag för våra kommande slutsatser.

2.1 Software as a service som affärsmodell

Affärsmodellen software as a service går ut på att företag och organisationer kan använda mjukvara som distribueras av programvaruleverantörer genom nätverksteknologi, istället för en lokal installation av programvaran (Dubey & Wagle, 2007). Verma (2011) poängterar också den centrala aspekten hos SaaS-produkter att applikationen såväl som all data lagras centraliserat hos leverantören. Papazoglou och Ribbers (2006) definierar dessa tjänster som ett *"package [of] software and infrastructure elements together with business and professional services to create a complete solution that they [the Application Service Providers] present to the end customer as a service on a subscription basis."* (s.584). Tanken med SaaS-modellen är alltså att konsumenter av mjukvarutjänster ska kunna hyra just den mjukvara som de för tillfället behöver och som de anser fungerar väl. All datalagring och behandling sker hos distributören av tjänsterna och kunderna kommer åt dessa genom sin webbläsare.

Vid lokala installationer har mjukvaruleverantörer oftast använt en affärsmodell där de debiterar en större klumpsumma vid införandet av en mjukvaruprodukt och sedan mindre summor varje gång kunden vill uppgradera produkten. SaaS-modellen innebär istället kortare perioder mellan uppdateringarna och mindre summor löpande licenskostnader (Dubey & Wagle, 2007). Vanligt är att kunden betalar löpande för en viss period t.ex. månadsvis eller per transaktion, projekt eller fasta kostnader per användare (Verma, 2011).

SaaS-modellen ger konsumenten av programvara en större flexibilitet då tjänster snabbt kan anskaffas eller tas bort och andra leverantörer kan väljas. Att användaren snabbt kan välja att avbryta en prenumeration på en tjänst innebär en stor förändring från traditionella licensmodeller då kunden först efter att betalt en stor summa och börjat använda programvaran kan komma att

upptäcka dess brister, men då vara låst till lösningen (Hoogvliet, 2008). Det är först på senare tid som SaaS-produkter blivit ett allvarligt hot mot den mer traditionella affärsmodellen. Genom den tekniska utvecklingen kan leverantörer av SaaS-produkter nu leverera lika komplexa och avancerade programvaror som en lokal installation skulle möjliggöra. För distribuerande företag innebär SaaS-modellen en stor fördel genom att de inte behöver utveckla sina produkter till flera olika plattformar, då detta inte spelar någon roll när tjänsterna används genom webbläsare. Detta innebär att det potentiella antalet framtida kunder blir större (Dubey & Wagle, 2007).

Enligt Dubey och Wagle (2007) kommer andelen SaaS-produkter att öka i förhållande till traditionella lokala installationer. De menar att SaaS-modellen ger kunderna ett antal fördelar t.ex. en större flexibilitet, bättre service från leverantören då dessa måste vara mer lyhörda för sina kunder för att inte tappa licenser och dessutom blir totalkostnaden mindre. Dessa fördelar identifierar också Vella (2011) men han visar också på att SaaS-modellen fortfarande är ny och leverantörer testas sig fram för att fånga kunder. Själva grunden för dessa tjänsters attraktionskraft är att de är enkla att få tillgång till och enkla att byta ut. Design blir centralt för dessa produkter då kunderna ofta får tillgång till produkterna för att under en begränsad tid kunna testa och utvärdera dessa och snabbt skapa sig en bild om dess tillgänglighet och funktionalitet. Dubey och Wagle (2007) poängterar därför vikten av att produkterna måste designas för att dess användare snabbt ska kunna börja arbeta med produkten och se nyttan med den och på så vis undvika att de vänder sig till en annan produkt och en annan leverantör. Hoogvliet (2008) adderar till detta att produkten måste ha ett väl designat gränssnitt och ett logiskt arbetssätt.

2.2 Self-service

Begreppet self-service, i det avseende vi använder det i den här uppsatsen, handlar om att erbjuda teknikbaserade tjänster till kunder utan att ansikte-mot-ansikte leda dem genom användandet utan istället låta dem utföra hela eller delar av uppgifterna på egen hand. Denna typ av self-service går under benämningen teknikbaserad self-service (TBSS) (Dabholkar, 1994; Ladik, 1999; Anselmsson, 2001; Wang & Namen, 2004; van Beuningen et al, 2009). Anselmsson (2001) har utifrån Dabholkars (1994) klassifikation av TBSS tagit fram en modell som illustrerar och exemplifierar förhållandet mellan olika typer av TBSS och sättet samt platsen som användaren interagerar med produkten (se figur 2.1). Denna modell tjänar sitt syfte här då den visar på hur den typ av TBSS som vi fokuserar förhåller sig till andra TBSS samt ger en inblick i vilken typ av system som self-serviceforskningen traditionellt kretsar kring. Överlag rör det sig, som figuren visar, om enklare produkter med grund funktionalitet som är lätt att överblicka (uttagsautomater, bensinpumpar, etc). Undantaget ser vi i cell 2, i vilken användarna kommer i direkt kontakt med systemet från sin hem-/jobb miljö, ofta via Internet. I denna cell indikerar exemplen utrymme för mer komplexitet och det är också denna typ av TBSS som vi fokuserar på i denna studie.

Forskning och industrirapporter som van Beuningen et al (2009) tagit del av visar att kunder till self-serviceprodukter med påtagligt högre frekvens stöter på problem än kunder till fullservice (personligt, ansikte-mot-ansikte, ledda) motsvarigheter. Detta är extra problematiskt när produkterna erbjuds i en onlinemiljö med låga kostnader för leverantörsbyte (van Beuningen et al, 2009) (jmf. avsnitt 2.1). För att råda bot på denna problematik måste man se till att

	At service site	At customer's place
Direct contact	<p>CELL 1</p> <p>Customer goes to service site and uses technology to perform service. <i>E.g. ATMs, automated ticket machines, self-scanning at retail and library checkouts, automated recipe guide in retail store, self-gas pumps, blood pressure machines, tourist info.</i></p>	<p>CELL 2</p> <p>Customer uses technology from home/work to perform service. <i>E.g. Internet shopping, interactive TV-shopping, reservations and information seeking over the Internet, account information, financial transactions, distance learning.</i></p>
Indirect contact	<p>CELL 3</p> <p>Customer goes too service site and uses automated telephone system to perform service. <i>E.g. automated wake-up calls at hotel room, telephone banking at bank, account information at libraries and retail stores.</i></p>	<p>CELL 4</p> <p>Customer calls automated telephone service from home work to perform service. <i>E.g. telephone-banking, automated ticket ordering over telephone (airports, ferries, cinemas), automated time schedules (e.g. busses, trains).</i></p>

Figur 2.1 Klassifikation av TBSS från Anselmsson (2001, s. 13), anpassad från Dabholkar (1994, s. 247).

användarna har utförliga informationsresurser att luta sig mot i handhavandet av produkten (ibid.). Kan man dessutom få kunderna att helt eller delvis producera denna information på egen hand så kan det möjliggöra ökad specialanpassning av innehållet och i längden en mer tillfredsställande upplevelse för kunderna (Pralhad & Ramaswamy, 2004). Att skapa sådan delaktighet i informationsskapandet är dock svårt, särskilt i fallet med komplexa produkter/tjänster där användarna ofta har svårt för att känna sig självsäkra nog eller motiverade att bidra. Man måste därför se till att användarna känner att rollen som informationsproducent är intressant, utmanande och/eller belönande, så att de lägger den energi som krävs för att bygga upp ett självförtroende i förhållande till produkten (van Beuningen et al, 2009).

Ett annat sätt att avhjälpa den höga felfrekvensen hos self-serviceprodukter är genom god gränssnittsdesign (Avery, 2009; Bikers & Slawsky, 2010; Wong et al, 2010). Extra viktigt för self-serviceprodukter är att man är konsekvent i sin design, försöker hålla processen som driver användandet så enkel och logisk som möjligt (Avery, 2009) samt att man i ett tidigt skede i designprocessen involverar slutanvändare för utvärdering och feedback för att öka chansen för ett gott mottagande (Krug, 2006; Avery, 2009; Sharp et al, 2011).

2.3 Designteori

2.3.1 Bakomliggande teorier

De designprinciper och -rekommendationer som kommer från Människa-datorinteraktions (MDI)-området idag har sin grund i flertalet teorier, framförallt kognitiva, sociala och organisatoriska, som härstammar från andra discipliner (Sharp et al, 2011). Sharp et al. (2011) beskriver teorier som en kunskapskälla som influerar MDI-området och som också styr forskningen inom detta. De beskriver vidare teorier som förklaringar av delar av ett fenomen,

vilka är värdefulla inom området för att förutse användares interaktion med olika grafiska gränssnitt och förmåga att utföra olika uppgifter inom dessa.

Kognitiva teorier har en central roll inom MDI-området och bidrar genom att fokusera på vad människan är bra på och dålig på, vilket bidrar till förståelsen om hur man ska designa för att utnyttja de starka sidorna och kompensera för de svaga när man utför design av teknik (Sharp et al., 2011). Kognitionsteorierna bidrar till MDI-området bl.a. genom att ge kunskap om hur det mänskliga minnet fungerar och hur människor planerar och genomför uppgifter (Shneiderman & Plaisant, 2010; Sharp et al., 2011). Sådana insikter ger kunskap om hur man bör utforma applikationer och dess funktioner för att öka chansen till att uppgifterna blir lösta.

Krug (2006) sammanfattar denna forskning kärnfyllt i tre punkter:

1) Vi läser inte, vi skummar.

Användare spenderar väldigt lite tid på att faktiskt läsa information, framförallt på en datorskärm där det är svårare att läsa än på papper. Istället skummar de igenom innehållet och letar efter ord eller fraser som stämmer överens med det vi söker. Anledningen till detta beteende är, enligt Krug (2006), i huvudsak att de har bråttom och helt enkelt inte anser sig ha tid att läsa, de är medvetna om att mycket lite av all tillgänglig information faktiskt rör det de är intresserade av samt det faktum att människor i hela sitt liv tränat på och blivit vana att skumma igenom tidningar, böcker och andra informationskällor.

2) Vi gör inte optimala val, vi nöjer oss.

Det verkar finnas en föreställning bland designers, menar Krug (2006), att användare skummar igenom sidan, överväger alla tillgängliga alternativ och sedan väljer det bäst lämpade. I verkligheten visar det sig dock att användare oftast väljer det första rimliga alternativet de hittar och provar det (Simon, 1957; Klein, 1998; Krug, 2006; Tidwell, 2011). Anledningar till detta är bland annat att det är svårt och tar lång tid att göra optimala val och det är mer kostnadseffektivt att göra ett ”tillräckligt bra” val (Klein, 1998). I applikationssammanhang är kostnaden för att göra ett felaktigt val inte heller särskilt stor då man, förutsatt att den finns, kan använda en ”tillbaka-knapp” (Krug, 2006). Är applikationen som används dåligt designad så finns dessutom risken att jämförelsen mellan olika alternativ inte ökar användarens chanser att välja rätt då inget alternativ verkar leda dit den tänkt sig (ibid.)

3) Vi räknar inte ut hur saker fungerar, vi provar oss fram.

Användare nyttjar i stor utsträckning applikationer, ibland dagligen, utan att veta vad de egentligen är till för eller hur de verkligen fungerar. Ibland skapar de till och med felaktiga mentala bilder av vad det är som händer när de använder applikationen – och varför det fungerar. Användaren är nöjd så länge det de tror sig göra faktiskt fungerar (Krug, 2006).

2.3.2 Design av komplexa self-serviceprodukter idag

Återkommande teman i litteraturen om SaaS och self-service är som vi sett ovan hur man kan hjälpa användarna med handhavandet av produkterna samt hur viktigt det är att applikationerna utformas så att användarna snabbt och problemfritt kan förstå och navigera sig i dem. I detta avsnitt lyfter vi därför fram tre teman som svarar mot dessa insikter; *Layout*, *Navigering* och *Hjälp*, under vilka vi presenterar hur dagens designpraktik ser ut. *Layout* och *Navigering* är valda då vi anser dem rymma de huvudsakliga poänger som ges i designlitteraturen kring self-serviceprodukter (se t.ex. Avery, 2009; Bikers & Slawsky, 2010; Wong et al, 2010) och därmed de viktigaste aspekter som måste fångas för att kunna implementera ett self-servicetänk i SaaS-design. *Hjälp*-temat tillkommer då komplexiteten i den kategori av produkter vi studerar går utanför gränserna för de traditionella self-serviceprodukterna samt att miljön de används i (se avsnitt 1.4) tillåter mer utrymme för hjälpfunktionalitet. Dessutom lyfter flera av de publikationer som behandlar design på mer allmän nivå (se t.ex. Schneiderman & Plaisant, 2010; Tidwell, 2011) fram vikten och behovet av välformulerad hjälp för att användare ska kunna hantera ett givet gränssnitt.

Layout:

Grafisk layout handlar om att presentera innehåll i tryckta eller digitala verk, på ett sätt som är lättillgängligt och som fångar beskådarens uppmärksamhet (Tidwell, 2011). Tidwell (2011) skriver: "*Page layout is the art of manipulating the user's attention on a page to convey meaning, sequence, and points of interaction*" (s.131).

Då människor skummar igenom innehåll istället för att läsa måste applikationer underlätta den typen av granskande (Krug, 2006). Grundläggande för att uppnå detta är en konsekvent grafisk layout, där den grafiska uppbyggnaden och den använda terminologin går att känna igen mellan de olika vyerna. Detta underlättar användarens inlärande och förståelse för layouten och dess beståndsdelar så att hon snabbt kan förstå hur nya vyer av gränssnittet är uppbyggda (Schneiderman & Plaisant, 2010). Även konventioner, alltså vedertagna designgrepp som återfinns i tidigare produkter inom samma domän, spelar en avgörande roll för detta då de skapar förväntningar hos användare om vart olika gränssnittsobjekt finns placerade och hur de kan manipuleras. Detta innebär potentiella effektivitetsökningar då användare kan utnyttja tidigare erfarenheter för att utföra samma moment i nya produkter (Krug, 2006; Tidwell, 2011). Konventioner ska betraktas som best-practice och alltid användas, om man inte är säker på att det man designar är värt en inlärningskurva för användarna (Krug, 2006).

Tidwell (2011) presenterar vidare ett antal aspekter av gränssnittslayout som enligt henne är grundläggande och som man måste ta hänsyn till för att uppnå en god layout. Dessa återkommer även i flera andra verk (se t.ex. Koblanck, 2003; Krug 2006; Vignelli, 2008) och presenteras sammanfattat nedan;

Visuell hierarki handlar om relationen mellan de olika objekten som visas på en sida. Genom att på olika sätt visa vilka objekt som är relaterade till varandra och vilka objekt som är delar av andra objekt gör man en sida mer lättillgänglig och minskar ansträngningen som behövs för att begripa den (Krug, 2006). Objektens betydelse i relation till varandra ska synas i de grafiska representationerna de har, alltså ska viktiga objekt se viktigare ut än de mindre viktiga, vilket t.ex.

kan uppnås genom storlek eller färgval (Krug, 2006; Tidwell, 2011). Genom att placera mindre objekt i större kan man också visa vilka delar av gränssnittet som är en del av andra delar, vilket underlättar för användaren att avgöra i vilken del av sidan som denna ska leta på (Krug, 2006; Tidwell, 2011).

Visuellt flöde handlar om hur användare tar in en sida visuellt, vilka områden som användaren lägger uppmärksamhet vid och i vilken ordning. Tidwell (2011) definierar visuellt flöde som: "*the tracks that readers' eyes tend to follow as they scan the page*" (s. 136). Man kan alltså säga att det handlar om vad användare tittar på och lägger märke till i ett gränssnitt. Att använda en layout med väl utformat visuellt flöde skapar möjligheter för att kontrollera vart användaren först lägger sin uppmärksamhet, vilket gör att man kan påverka användarens interaktion med gränssnittet. Man kan skapa visuellt flöde genom att placera fokuspunkter i gränssnittet som lockar till sig användarens uppmärksamhet (Tidwell, 2011). Fokuspunkter kan vara vilka grafiska objekt som helst, men som placerats eller designats för att urskilja sig från det resterande gränssnittet. Genom att placera dessa fokuspunkter där användarens interaktion ska börja leder man användaren rätt och minskar bördan för denne. Från denna fokuspunkt bör man sedan placera den resterande interaktion som ska ske i gränssnittet i en visuell bana vilket försäkrar att användaren inte behöver gå fram och tillbaka i det för att slutföra en uppgift. T.ex. bör knappar som avslutar transaktioner av ifylld data ("klar"-knappar) vara stora och välmärkta samt placerade i slutet av det visuella flödet (Tidwell, 2011).

Gruppering av objekt i gränssnitt är viktigt för att leda användaren till rätt objekt och minimera bördan med att behöva leta efter dem. Människor försöker omedvetet skapa ordning och helhet av det vi ser genom den tidigare kunskap och erfarenhet vi har (Koblanck, 2003; Tidwell, 2011). Vi försöker hela tiden forma en större helhet av de mindre delarna vi ser framför oss (Koblanck, 2003; Tidwell, 2011). Gestaltpsykologin är en gren inom psykologin som länge har forskat inom detta område och presenterat tre gestaltungs-lagar: Lagen om närhet ("proximity"), lagen om likhet ("similarity") och lagen om slutenhet ("closure"). Lagen om närhet säger att, genom att placera relaterade objekt nära varandra visar man att de hör ihop och det är således enklare för användaren att hitta de objekt denna letar efter (Koblanck, 2003; Tidwell, 2011). Om man placerar kontrollerna för långt bort från den del av gränssnittet man vill manipulera och utan att göra dem extra synliga, kommer användaren behöva leta efter dem (Tidwell, 2011). Istället bör man alltså, för att minimera letandet i gränssnitt, placera tillhörande objekt nära varandra. Lagen om likhet menar att pga. att den mänskliga hjärnan försöker skapa ett sammanhang av mindre former, kommer de former som liknar varandra sorteras ut och antas höra ihop (Koblanck, 2003; Tidwell, 2011). Detta kan man utnyttja genom att relatera spridda objekt med varandra genom dess form och utseende för att tala om för användaren att de hör ihop (Koblanck, 2003). Lagen om slutenhet säger att om något avgränsas från dess omgivning på något sätt upplevs det som att det hör ihop, oavsett om det avgränsade inte har likadan form (Koblanck, 2003; Tidwell, 2011).

Uppdelning av sidan i logiska avdelningar som är tydligt avdelade från varandra är viktigt för att användare snabbt ska kunna avgöra vilka delar av sidan som är intressanta och vilka delar som de kan bortse från (Krug, 2006). Enligt Krug (2006) bestämmer sig en användare väldigt snabbt för vilka delar av en sida denna tror är relevanta och bortser helt från de andra delarna, vilket gör att information som placerats i dessa delar förbigås helt. Dessa uppdelningar kan med fördel göras

med s.k. "white spaces" (vita ytor) som på ett subtilt sätt skapar avgränsningar av gränssnittets olika delar (Vignelli, 2008; Tidwell, 2011). Tidwell (2011) menar att vita ytor kan användas både för att skapa visuella hierarkier och visuellt flöde, vilka av resonemanget ovan visat sig vara centrala aspekter i grafisk layout.

Navigering:

Krug (2006) jämför att navigera sig i en applikation med att befinna sig i en ny affär och menar att du när du kommer in i affären har två val; själv leta upp det du vill hitta eller fråga någon. I en applikation motsvarar detta att klicka runt i applikationen för att hitta rätt eller skriva in sökord i ett sökfält (om det finns något). Börjar du leta på egen hand så kommer du titta efter uppenbara skyltar som leder dig mot det du söker och i bästa fall, om skyltarna är väl placerade och korrekt formulerade så hittar du ganska snart rätt. Om du inte hittar rätt kommer du för eller senare använda dig av personalen/sökmotorn för att guidas i rätt riktning, förutsatt att du inte redan tröttnat och lämnat affären/stängt ner applikationen. Ger personalen/sökmotorn inte svar som leder dig rätt kan du fråga någon annan/omformulera frågan, men vid det laget börjar tålmodet vara ganska väl påfrestat och risken att du lämnar missnöjd överhängande stor.

Tidwell (2011) lyfter fram ytterligare en viktig aspekt hos applikationsnavigeringen när hon liknar det vid att pendla; när man måste dela upp ett gränssnitt i flera delar eller flera sidor måste avståndet användaren ska ta sig vara så kort som möjligt, för att det ska upplevas så bekvämt som möjligt. Dessutom innebär det alltid en ansträngning för användaren varje gång denna öppnar en ny sida som ska granskas och analyseras, vilket adderar till den totala ansträngning som krävs för att slutföra en uppgift eller nå den eftersökta informationen (Sharp et al, 2011; Tidwell, 2011).

Koblanck (2003) poängterar också vikten av att navigering ska minska avståndet som användare måste ta sig när hon säger att applikationer ska vara funktionellt enkla för att ge god tillgänglighet.

Oavsett liknelse så är litteraturen samstämmig i att god applikationsnavigering uppnås genom att ge användaren en tydlig indikator på 1) Var denne befinner sig, 2) Vad den aktuella applikationen innehåller och 3) Hur användaren kan ta sig till det den söker (Koblanck, 2003; Krug, 2006; Schneiderman & Plaisant, 2010; Tidwell, 2011).

Litteraturen presenterar många olika sätt att lösa dessa designproblem. Gemensamt för lösningarna är behovet av översikt över applikationen och att visuellt presentera var användaren befinner sig i förhållande till denna översikt. Detta kan, i mer komplexa applikationer, göras genom att en liten del av varje sida dedicerats till en grupp knappar eller länkar som leder till alla nyckelfunktioner. Detta gör att man, oavsett var i applikationen man befinner sig, hela tiden kan se dess övergripande struktur. För att veta var man är i förhållande till denna översikt kan man enkelt markera det aktuella steget med t.ex. en avvikande nyans eller skuggning mot de övriga valen (Krug, 2006; Tidwell, 2011) (se figur 2.2). Förtydliganden kan göras t.ex. genom att använda sig av "breadcrumbs" (Krug, 2006, s. 50; Tidwell, 2011, s. 121-123) som i toppen av sidan skriver ut alla nivåer i strukturen man borrar sig ner i för att ta sig dit man är (se figur 2.2) och/eller färgkodning där varje nyckelfunktion tilldelas en gränssnittsfärg och all funktionalitet som finns under den nyckelfunktionen delar sedan detta färgschema (Krug, 2006; Tidwell, 2011). Krug (2006) poängterar också vikten av att varje sida har ett eget namn så att en användare enkelt kan navigera sig liknande hur man gör med gatuskyltar. Titta på figur 2.3 och föreställ dig hur du



Figur 2.2 Exempel på nyansdifferentiering i menyval med "Breadcrumbs" under (från www.goteborg.se)

navigerar med hjälp av gatuskyltar; "Jag är i korsningen mellan 'Arbete' och 'Lediga jobb i Göteborgs Stad'". Det ger en bra mentalbild av var man befinner sig och hur man kan röra sig i gränssnittet (jmf. Krug, 2006, s. 72).



Figur 2.3 Exempel på användandet av unika sidnamn för att underlätta navigeringen (från www.goteborg.se)

Om man vid något tillfälle känner att man inte är där man förväntade sig att vara eller hamnat i ett läge som man vill ta sig ur måste applikationen tydligt visa hur man på ett snabbt och säkert sätt kan ta sig ur den besvärliga situationen utan att det medför obehag som förlust av data, omständiga popup-rutor eller ifyllda formulär som töms (Krug, 2006; Tidwell, 2011). Tidwell (2011) menar att detta gör att användare vågar utforska applikationen och snabbare lär sig dess gränssnitt och upplever det mer positivt än användare som inte tillåts utforska under dessa förutsättningar. För att förverkliga denna aspekt av användarupplevelsen måste "ångra"-knappar finnas vid datamanipulation, bakåt-knappar måste alltid stega tillbaka till huvudsidan (för att undvika navigationsförvirring) och tydliga knappar som uppenbart tar användaren bort från den aktuella platsen och till ett välbekant område måste finnas på varje sida som har begränsade navigeringsmöjligheter (Tidwell, 2011).

Hjälp:

När användare ska lära sig använda ett nytt gränssnitt, eller nya funktioner i ett gränssnitt krävs det en ansträngning av hängivenhet och koncentration. Ovana användare har ofta svårt att förstå grundläggande skärmobjekt medan användare med större datorvana får problem först när de ska lära sig mer avancerade funktioner eller nya användningsområden. Oftast lär sig användare nya funktioner eller gränssnitt genom att lära sig av någon annan, eller genom att testa sig fram. Hjälpdokumentation används sällan under sådan inlärning, så länge som systemet är designat så att användaren vet vad denna ska göra härnäst, men när användaren någon gång fastnar med en uppgift blir hjälpdokumentation och hjälpfunktioner i olika former avgörande. Anledningen till svårigheter att lära sig ett system och de fel som förekommer vid användning beror främst på bristfällig design med ogenomtänkta gränssnitt. Men att designa ett system som kan användas av

alla är svårt. Därför är lättillgänglig hjälpdokumentation fortsatt väsentlig (Schneiderman & Plaisant, 2010). Krug (2006) poängterar dock; *“Your objective should always be to eliminate instructions entirely by making everything self-explanatory, or as close to it as possible. When instructions are absolutely necessary, cut them back to the bare minimum”* (s47).

Hjälpfunktioner kan ta olika form och nedan presenterar vi några av de vanligaste formerna som används idag, i huvudsak baserat på Schneiderman och Plaisants (2010) genomgång:

Manualer där all interaktion med systemet förklaras och där all systemets funktionalitet går igenom. Består oftast av en omfattande dokumentation där varje del av ett gränssnitt förklaras i detalj (Schneiderman & Plaisant, 2010). Det finns en klar trend i att manualer i högre grad placeras online, istället för att vara pappersbaserad (Schneiderman & Plaisant, 2010). Detta ger fördelarna att användaren snabbt kan komma åt dokumentationen, det är effektivare att navigera i dokumentationen genom t.ex. sökfunktioner och användaren kan dra nytta av interaktiva funktioner så som bokmärkning. Uppdatering av hjälpdokumentation effektiviseras också betydligt med onlinebaserad dokumentation (ibid.).

Snabbstarts guide är en kortfattad form av användarmanual där användaren får ett antal korta exempel på hur interaktionen med ett gränssnitt kan gå till för att användaren ska kunna komma igång att arbeta i gränssnittet. Dessa består ofta av många exempel med korta beskrivningar och mycket grafiska presentationer (Schneiderman & Plaisant, 2010).

Wikis är webbaserade diskussionssidor som både kan vara öppna för alla att lägga till och redigera innehållet eller begränsade till att bara vissa personer med rättigheter kan lägga till och redigera information (Schneiderman & Plaisant, 2010). Wikis används som informationskälla där vissa ämnen eller teman presenteras och som diskussionsmedium där medlemmarna hos Wikin bidrar till kunskapsbildningen genom löpande diskussioner om olika ämnen. Innehållet kan vanligtvis hittas genom sökfunktioner eller genom menyer (Bocij et al., 2008). Wikis är på så vis ett sätt att ge plats för den användargenererade information som van Beuningen et al (2009) håller som viktig för användarupplevelsen av TBSS (se kap. 2.2).

Online communities är webbaserade samlingsplatser vars syfte är att skapa social interaktion. Dessa kan t.ex. skapas och användas i syfte att diskutera och samarbeta kring kunskapsbildandet av ett system (Sharp et al. 2011). Bocij et al. (2008) menar att det är den bidragande mentaliteten, att individer uppmuntras till att bidra med sin kunskap för att besvara andra individers frågor om olika ämnen, som kännetecknar denna typ av interaktion, kallad "the wisdom of crowds" (Bocij et al., 2008, s.163). Individer som använder communities söker i regel inte en specifik person, utan vänder sig till snarare öppet till användare som besitter den efterfrågade kompetensen (Schneiderman & Plaisant, 2010). På så sätt ger

communities på samma sätt som Wikis utrymme för användargenererad information i linje med van Beuningen et als (2009) rekommendationer.

Online-handledning, eller "tutorials", är en interaktiv testmiljö där användare kan utföra tester med ett gränssnitt genom att utföra simulerade uppgifter. Att få användaren att lära sig själv genom testövningar är en av de största styrkorna med tutorials och ett svar på användarnas strävan att vara självlärande istället för att läsa omfattande instruktionsmanualer (Schneiderman & Plaisant, 2010).

Animerade demonstrationer används för att göra utbildningsmaterial som stegvis visar hur funktioner används i ett systems gränssnitt. Inkorporering med vanliga uppspelningskontroller gör att användare av dessa demonstrationer kan pausa, spola tillbaka eller ändra tempo för att kunna anpassa uppspelningen. Animerade demonstrationer är oftast inspelningar av ett bildspel, skärminspelning eller videoupptagning av någon som använder systemet (Schneiderman & Plaisant, 2010).

Annan gränssnittsbaserad hjälp som finns inbäddad i ett systems gränssnittsobjekt och funktionalitet ger användaren aktuell hjälp under specifika tillfällen av dennes interaktion med systemet (Schneiderman & Plaisant, 2010). Olika typer av gränssnittsbaserad hjälp relevant för SaaS-self-serviceapplikationer kommer presenteras djupare nedan.

Som nämnts ovan menar Schneiderman och Plaisant (2010) att användare främst vill lära sig ett gränssnitt på egen hand och snabbt kunna se resultat av deras lärande. De använder tidigare kunskap under detta självlärande som t.ex. kunskap om den aktuella domänen, erfarenheter av tidigare gränssnitt, men också genom att testa sig fram. Därför är det viktigt att man i applikationen använder begrepp som i så stor utsträckning som möjligt känns igen av användarna (Sharp et al, 2011). Genom att designa gränssnitt som innehåller inbäddad hjälp kan man öka chansen för att användares självlärande blir framgångsrikt (Schneiderman & Plaisant, 2010). Som också nämnts ovan finns det många typer av gränssnitts-inbäddade hjälpfunktioner som kan användas, dessa presenteras i detalj nedan. Vi har valt att behålla de engelska benämningarna då de är vedertagna begrepp även inom svensk designpraxis.

Tooltips - Korta beskrivningar av gränssnittsobjekts funktionalitet som visas när användaren placerar muspekaren över dessa. Används för att beskriva objekt eller funktioner i gränssnittet som inte är självförklarande (Tidwell, 2011; Schneiderman & Plaisant, 2010).

Input hint - Bredvid inputfält placeras beskrivning av, eller exempel på, vad användaren förväntas mata in i fälten. Detta underlättar för användaren att förstå vad denna ska fylla i och i vilket format det förväntas vara (Tidwell, 2011).

Input prompt - Inputfält som presenteras för användaren är ifyllda från början med en fråga eller uppmaning på vad användaren ska göra eller fylla i. Detta gör

gränssnittet mer självförklarande och kan ibland vara lättare för användaren att upptäcka än "input hints" (Tidwell, 2011).

Start up tip - En startruta visar en förklaring av en funktion varje gång ett system startas. Dessa kan användas för att visa slumpmässiga funktioner, men mer användbart är att visa funktioner som just den aktuella användaren inte använt alls för att hjälpa denna igång med dessa (Schneiderman & Plaisant, 2010).

Content-specific help - Specifik hjälp och funktionsbeskrivningar för den vy som användaren för tillfället befinner sig i. Placeras t.ex. i en avgränsad del av gränssnittet som användaren efter önskemål kan flytta eller ta bort (Schneiderman & Plaisant, 2010; Tidwell, 2011).

Oftast används en kombination av hjälpfunktionerna som nämns ovan för att tillgodose olika användares behov. Tidwell (2011) poängterar att olika användare uppskattar olika typer av hjälp, vilket dessutom förändras med tiden. En balans mellan att erbjuda användaren hjälp och att irritera användaren genom att överanvända hjälpfunktionalitet måste alltså träffas.

2.4 Teorins roll i studien

Den litteratur som vi nu gått igenom kommer, som vi skrev inledningsvis i detta kapitel och som visas i figur 1.2, utgöra underlag för studiens resulterande designramverk för komplexa SaaS-self-serviceapplikationer som presenteras i kapitel 6.

Texterna om SaaS och Self-service har bidragit med förståelse för den specifika produkttypens (komplex SaaS-self-service) förutsättningar och roll som verktyg. Detta tillsammans med avsnittet om designteori används för att bygga våra intervju- och observationsunderlag för att säkerställa att vi har rätt fokus i dessa. Designavsnittet kommer också att användas i vår resultatanalys för att ställas mot upptäckterna från intervjuerna och observationerna. På så vis kan vi hitta likheter och eventuella olikheter och föra ett resonemang som låter oss formulera nya och/eller anpassa existerande designprinciper så att de blir direkt applicerbara på komplexa SaaS-self-serviceapplikationer.



3. Fallstudieobjekt; "Barium Live!"

För att relatera de insikter som ges i litteraturen diskuterad ovan till ett verkligt fall samt komplettera dessa insikter med faktiska användares upplevelser kommer vi att göra en fallstudie av en produkt kallad "Barium Live!"¹. I detta kapitel kommer vi översiktligt att beskriva vad "Barium Live!" är och i viss utsträckning gå igenom dess funktionalitet.

3.1 Vad är "Barium Live!"?

"Barium Live!" är ett webbaserat SaaS-verktyg för att modellera organisationsprocesser som man sedan kan omvandla till ett fungerande, interaktivt system för att hantera de olika stegen i processen. Produkten är skapad av det Göteborgsbaserade företaget Barium AB. I dagsläget är inte "Barium Live!" en renodlad self-serviceapplikation utan nya användare av applikationen utbildas under upprepade workshops ledda av representanter, s.k. processkonsulter, från produktionsbolaget. Barium hyser dock en önskan om - och har påbörjat ett arbete för - att verktyget i större utsträckning skall fungera och säljas enligt en self-servicemodell där användare med olika roller ska kunna skapa ett konto, logga in och med endast de hjälpmedel som finns tillgängliga i gränssnittet kunna förstå och börja använda produkten.

Fördelen med att undersöka en produkt som inte är utformad för att fungera enligt en self-servicemodell är att vi har utrymme att se vilka delar av produkten som ändå fungerar väl utan utbildning och vilka delar som verkligen är problematiska. Det ger oss möjlighet att se vilken typ av hjälp och/eller vägledning som användarna är i behov av i olika situationer och fritt spelrum att, utifrån tidigare forskning och studiens upptäckter, skapa teori i form av designprinciper för hur sådana situationer borde designas för att svara mot den identifierade problematiken.

Bariums intresse av att förbättra self-serviceupplevelsen i sin produkt är också vad som initierat detta arbete och deras vinning ligger i att få ta del av våra slutsatser kring design av komplexa SaaS-self-serviceprodukter samt de upptäckter vi gör rörande deras produkt i vår undersökning.

3.2 Funktionaliteten i "Barium Live!"

"Barium Live!" är en komplex produkt med mycket funktionalitet. Översiktligt kan man dock dela upp funktionaliteten i tre huvudsakliga steg som leder fram till en färdig processapplikation. Dessa är; Modellera – Bygg – Kör, vilka vi kommer att gå igenom nedan. I tillägg till dessa finns funktionalitet för att på olika sätt strukturera uppgifter man förväntas utföra i olika processer samt följa upp och utvärdera processerna och arbetet i dem. Också detta kommer vi att illustrera i nedanstående funktionalitetsgenomgång.

Målgruppen för produkten är tjänstemän som jobbar med verksamhetsfrågor, inte i första hand anställda på IT-avdelningar. Barium ser i huvudsak tre typer av användare; 1) *De som modellerar*

¹ "Barium Live!" kan hittas på www.bariumlive.com och går att testa gratis även av privatpersoner. Prova gärna för att få ytterligare förståelse för de funktioner vi beskriver i detta arbete.

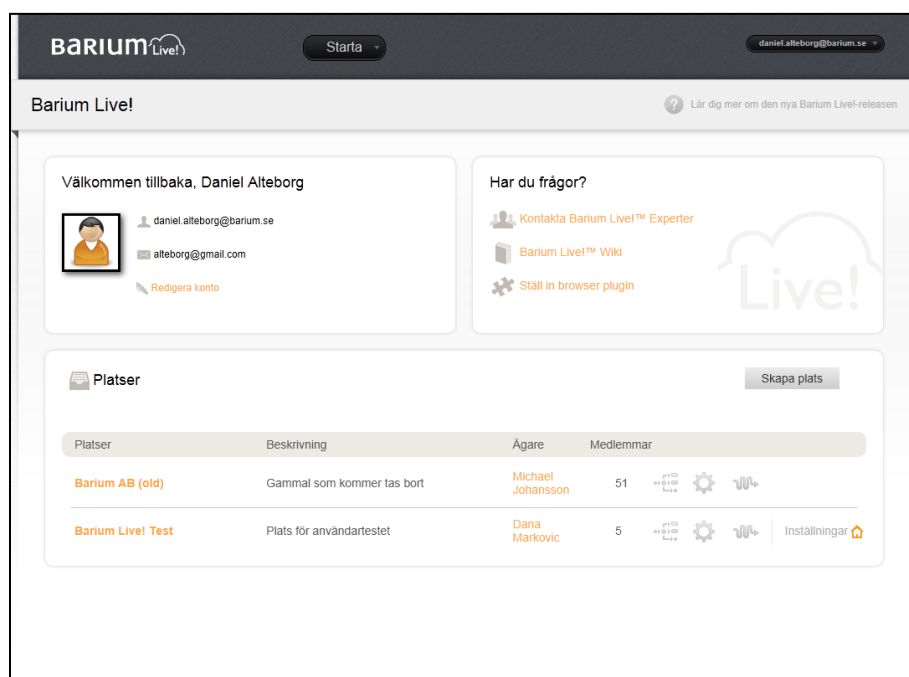
processer och skapar applikationer. Detta är i huvudsak personer som är ansvariga för den del av verksamheten där processerna återfinns, men tanken med produkten är att så många som möjligt ska kunna vara med och lämna förslag på eller direkt utforma processerna så att man hela tiden jobbar med processutveckling. 2) *De som utför arbetsuppgifter (processteg) i de skapade applikationerna.* Det här är alla personer som är involverade i en specifik process. När någon uppgift i processen ska utföras av en viss person eller roll notifieras denne/dessa av systemet och de måste utföra utgifter i och/eller utanför systemet. 3) *De som använder produkten för uppföljning och utvärdering av processerna.* Det här kan vara verksamhetschefer som är intresserade av att upptäcka potentiella flaskhalsar i processerna eller se andra indikationer på hur väl ärenden hanteras i organisationen.

Normalt finns alla tre typerna av användare i en och samma organisation och olika personer kan ha konton med tillgång till funktionalitet utefter vad den har för användningssyfte med produkten. En och samma person kan också ha alla tre rollerna och alltså modellera/bygga, arbeta i och följa upp/utvärdera processerna.

Nedan följer genomgången av produktens olika delar.

3.2.1 Startside

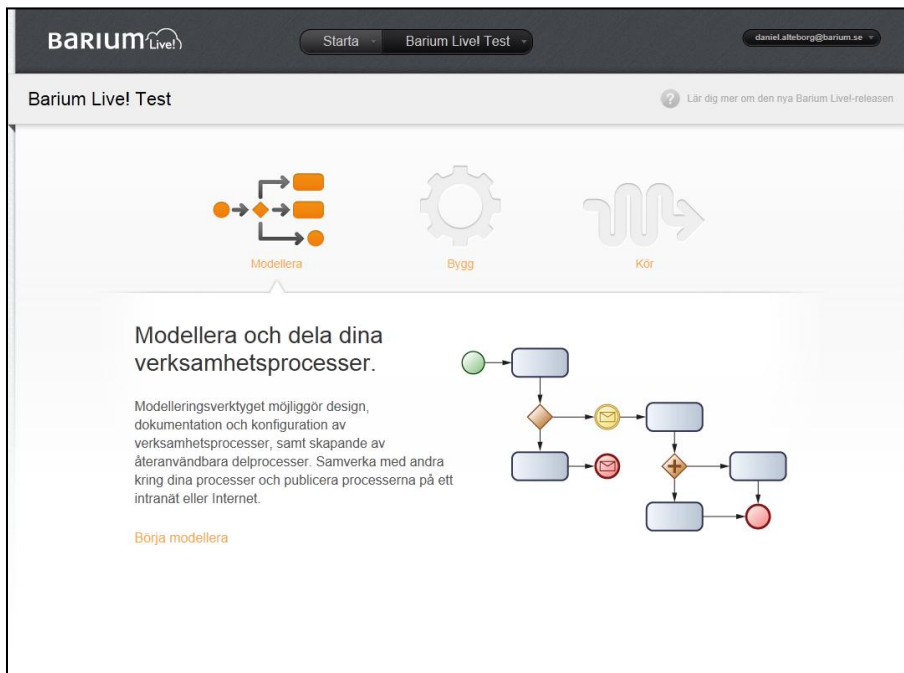
Det första man möts av när man loggar in till ”Barium Live!” är produktens startside som visas nedan i figur 3.1. På denna sida ser man information om sitt konto och de olika ”platserna” man är medlem i. Platser används för att strukturera allt innehåll i ”Barium Live!” och kan liknas med olika grupper där användare samlar relaterade processer, applikationer och uppgifter. Man kan som användare vara del av obegränsat antal platser och själv skapa nya platser. Man väljer vilken plats man vill jobba med genom att välja den i listan på platser man är medlem av. På startsidan finns också ett avsnitt med hjälpfunktioner, genvägar till produktens Wiki och till ett kontaktformulär. Startsidan ger en överblick över användarens konto, de platser som denna är medlem i och de inbjudningar till andra platser som användaren fått.



Figur 3.1 Startsidan

3.2.2 Navigera mellan delarna

När man i startsidan valt vilken plats man vill öppna kommer man till den valda platsens startguide, som visas nedan i figur 3.2.



Figur 3.2 Startguide för platsen

Startguiden består av en sida med tre ikoner som representerar ”Barium Live!”:s övergripande delar: *Modellera*, *Bygg* och *Kör*. När man rör muspekaren över de olika ikonerna får man en beskrivning av vad de olika delarna gör och man kan under dessa beskrivningar klicka sig fram till respektive del för att använda dessa.

Alternativt kan man för att navigera mellan produktens olika delar också använda den aktuella platsens drop-downmeny som är placerad i överkanten i alla fönster. Denna drop-down meny visas i figur 3.3.

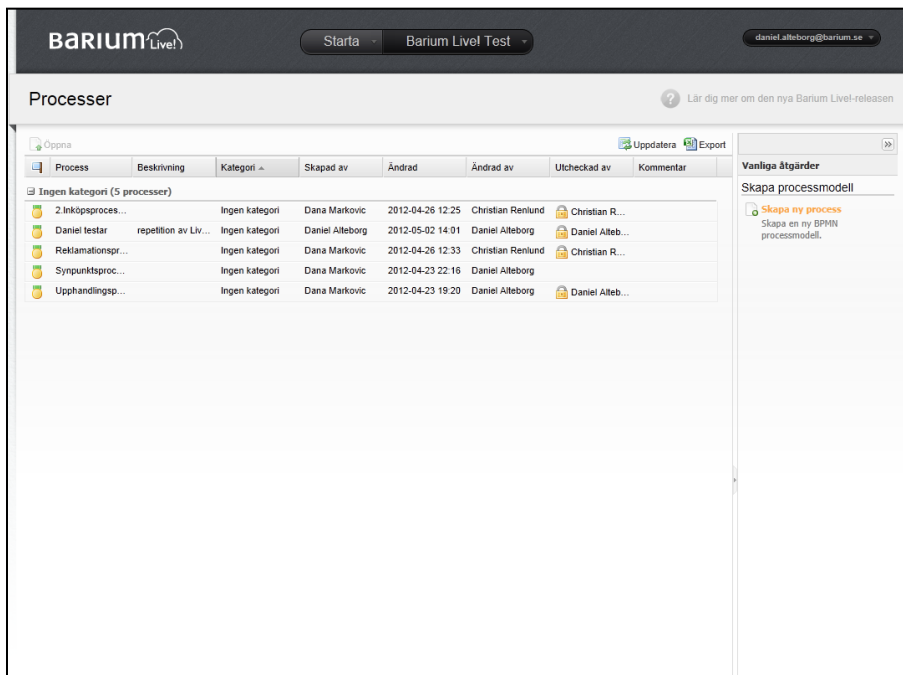


Figur 3.3 Drop-downmeny

I denna meny får man också en kortfattad beskrivning av de olika delarnas funktionalitet. Drop-downmenyn finns tillgänglig oavsett vart man befinner sig i ”Barium Live!”, när en plats har valts och öppnats från startsidan.

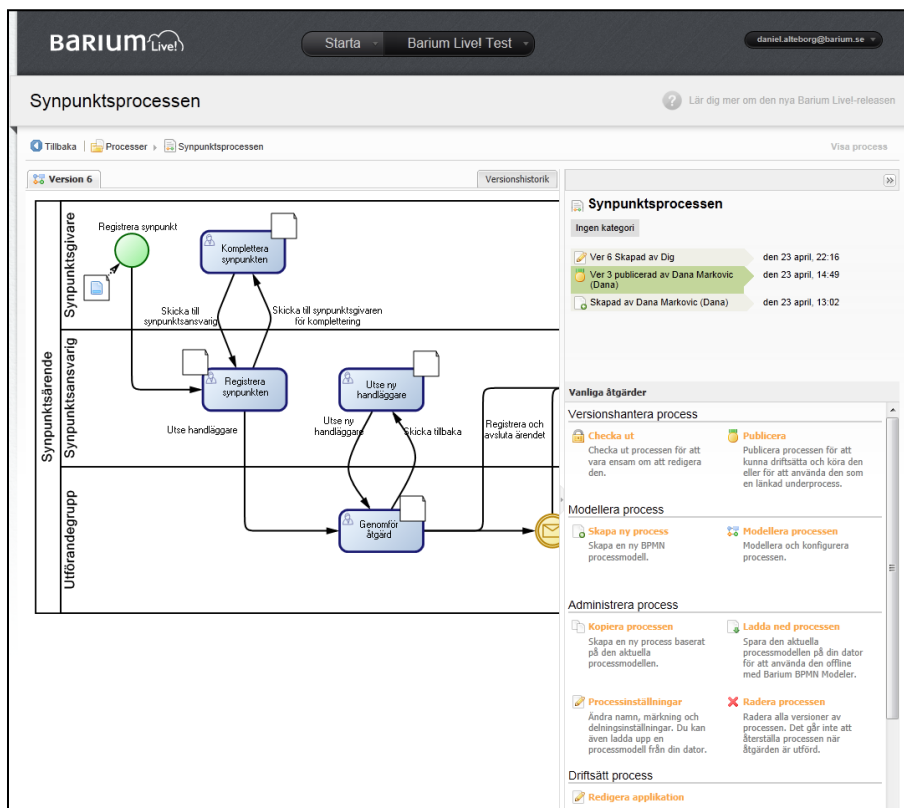
3.2.3 Modellera

Den första vyn som visas i modelleringssteget är en vy där man i listform ser de befintliga processerna som finns i den aktuella platsen. Längst till höger finns en meny där man kan välja att skapa en ny process. Om man väljer att öppna en befintlig process öppnas en vy där man ser en förhandsvisning av processen, som visas i figur 3.4 nedan och olika menyalternativ där man kan arbeta med denna.



Figur 3.4 Lista över befintliga processer

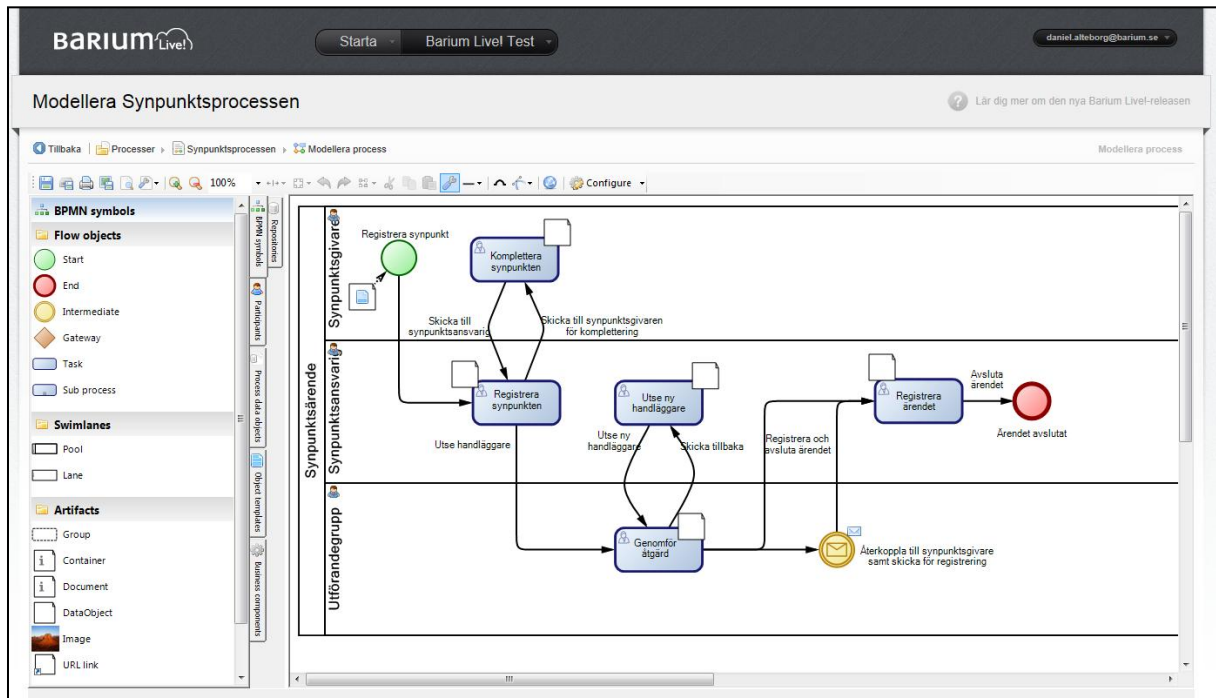
Om man väljer att öppna en befintlig process, öppnas en översiktsvy över den valda processen, som visas nedan i figur 3.5.



Figur 3.5 Processöversikt

I denna vy visas en förhandsvisning av processen och till höger finns en palett där man får information om versionshistoriken. Under denna information finns menyalternativ med vanliga åtgärder. Dessa består av funktioner för att versionshantera processen, modellera processen och för att administrera processen. Versionshanteringen används för att publicera nya versioner av processen och för att återställa en process till en föregående version ifall att något blivit fel med en nyare version. Administrera en process gör man främst när man ska göra den körbar och koppla skapade formulär till olika steg av processen (se avsnitt “3.2.5 Köra processer” nedan). Man kan även i denna meny välja att driftsätta processen vilket innebär att man skapar en körbar process, eller en “processapplikation” som det heter i ”Barium Live!”. Först då kan man börja utforma körbarheten i processen i senare steg av ”Barium Live!”.

Om man istället för att öppna en befintlig process väljer att skapa en helt ny process öppnas, efter att namn och beskrivning fyllts i, modelleringsfönstret där en ny process kan skapas (se figur 3.6 nedan).

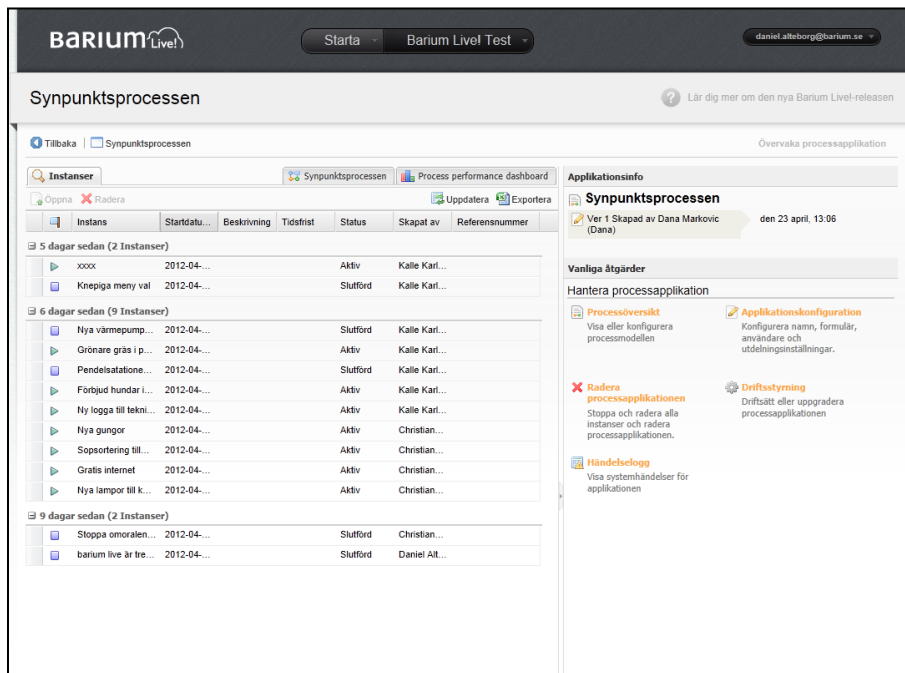


Figur 3.6 Modelleringsfönstret

I mitten av modelleringsfönstret i "Barium Live!" finns en rityta där processen ritas. Till vänster i fönstret finns en palett med olika verktyg som behövs för att skapa en process, grupperade på ett antal flikar. Som standard visas en flik med BPMN (Business Process Modeling Notation)-symboler som är tillgängliga för modelleringen att använda vid skapandet av processer. Dessa ritar man ut genom att dra och släppa dem i ritytan. Fliken med dessa symboler är den mest centrala för modelleringen men det finns också under de övriga flikarna andra verktyg som används i modelleringen av processer. T.ex. finns det under "participants"-fliken verktyg för att ange vilka roller som finns i processen och under "object templates"-fliken visas vilka formulärobject som är skapade och som finns tillgängliga att använda i processen. Ovanför ritytan finns en menyrad med verktyg som är av mer generell natur än de BPMN specifika som finns i paletten till vänster. Här hittar man verktyg som t.ex. "spara", "skriv ut", "zooma" och "ångra". Objekt som placerats ut på ritytan kan senare placeras om genom att flytta runt dem och man kan också skapa relationer mellan relaterade objekt genom att dra pilar mellan dem.

3.2.4 Bygg

Byggdelen av ”Barium Live!” används för att göra om modellerade processer till körbara processer, alltså processapplikationer. När man först öppnar ”bygg”-delen av ”Barium Live!” möts man av en lista på de tillgängliga processapplikationer som redan finns, likt den första vyn som visas när man går in i modelleringsdelen. I listan väljer man vilken processapplikation man vill arbeta med och öppnar den. Då öppnas en översiktlig processapplikationsvy där man i en centralt placerad lista ser alla pågående och avslutade instanser av denna process (se figur 3.7 nedan).

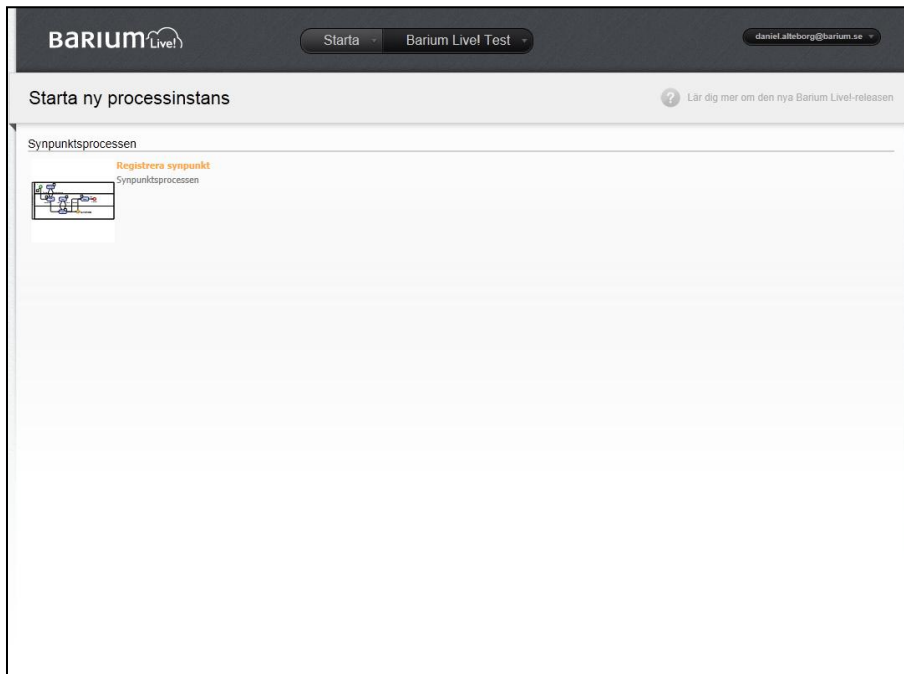


Figur 3.7 Översikt över en processapplikation

Till höger av listan finns en palett med versionsinformation samt olika menyalternativ som kan användas för att modifiera processapplikationen. En central funktion som finns i menyn är menyalternativ för att konfigurera applikationen, vilket är viktigt för att göra en process körbar och ger användaren möjlighet att skapa egna formulär för de olika rollerna som är inblandade i processen. Det finns också menyalternativ för versionshantering av applikationen, att se händelselogg, att ta bort applikationen och dess instanser, samt genväg till att modellera processen. Över listan på befintliga instanser finns flikar för att visa en dashboard för övervakning av applikationens prestanda (se figur 3.10). Här kan man se hur lång tid de olika stegen tar att slutföra vilket kan bli underlag för att effektivisera processen. ”Bygg”-delen av ”Barium Live!” är idag allt för komplex för att fungera enligt en self-service modell, därför har vi i vår undersökning inte tagit med denna.

3.2.5 Köra processer

I kör-delen av ”Barium Live!” kan man skapa nya instanser av befintliga processer, dvs. man startar en process. Denna funktionalitet i ”Barium Live!” når man genom att välja alternativet ”skapa ny processinstans” under kör-delen. I denna vy (se figur 3.8 nedan) visas de processer som den aktuella användaren har behörighet att starta.

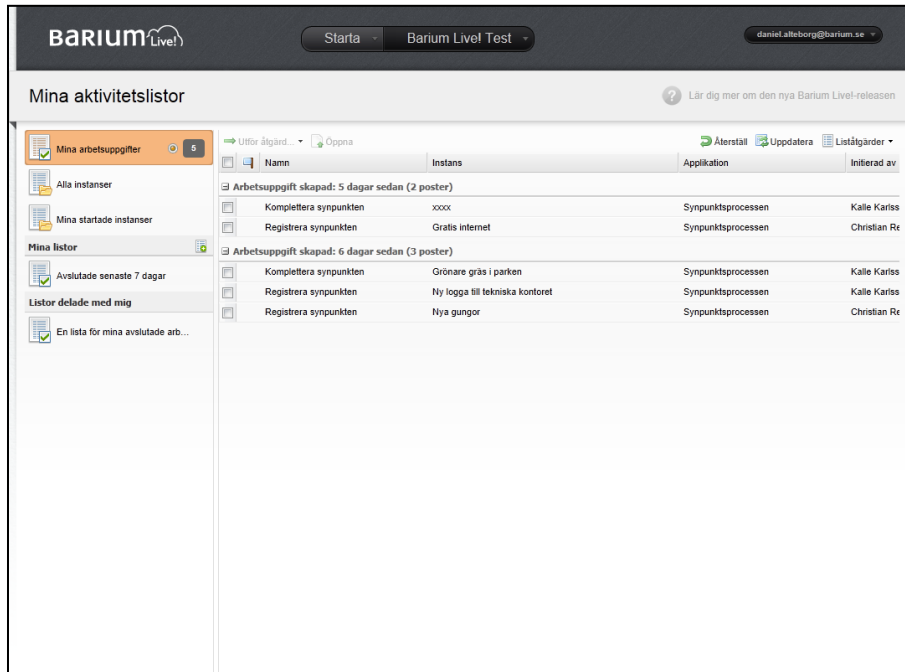


Figur 3.8 Skapa ny processinstans

Genom att klicka på någon av dessa startpunkter öppnas det formulär som är associerat med detta startevent (om man valt att konfigurera sin processapplikation med ett formulär i startpunkten). När användaren trycker på ”starta”-knappen som finns precis över formuläret startas processinstansen.

3.2.6 Mina aktivitetslistor

I kör-steget kan man istället för att starta ny processinstans välja att visa listor på befintliga instanser eller arbetsuppgifter. Då väljer man istället under kör-delen att öppna "mina aktivitetslistor", vilket öppnar "mina aktivitetslistor"-vyn som visas nedan i figur 3.9.



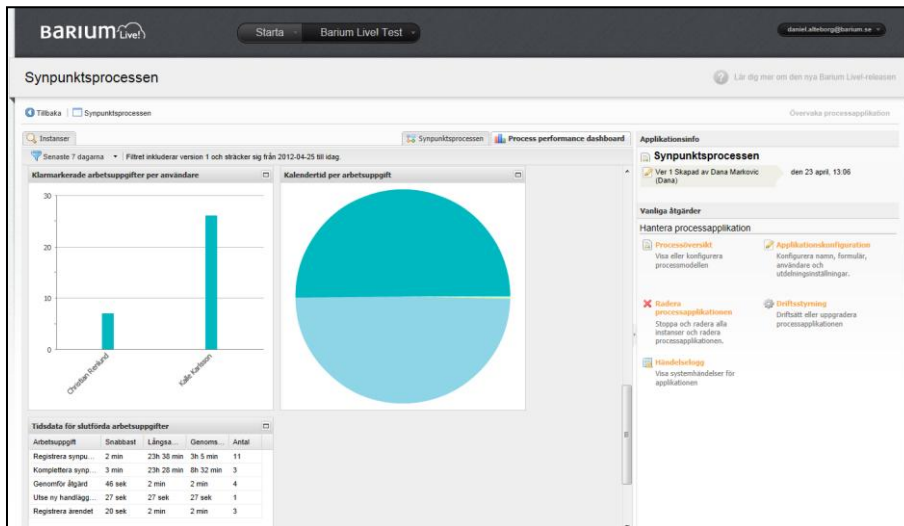
Figur 3.9 Mina aktivitetslistor

Denna vy visar i mitten listor på antingen befintliga instanser, eller arbetsuppgifter beroende på vilken av dessa listor man för tillfället valt att visa.

Man kan även själv skapa egna listor, som visar det man själv väljer. Till vänster om den aktuella listan finns en palett där man väljer vilken lista som ska visas. Under de befintliga listor som finns som standard hamnar de egna listor som användaren skapar. Under de listor användaren själv skapat visas listor som är skapade av andra användare och som delats med den aktuella användaren. För att skapa egna listor eller redigera befintliga listor väljer man "liståtgärder"-knappen som finns i en menyrad till höger över den lista som visas för tillfället. Där kan man också välja vilken lista som ska visas från början när man går in i denna vy.

3.2.7 Uppföljning

Som nämnts tidigare, finns det i tillägg till dessa ovan beskrivna funktioner möjligheten att följa upp och analysera processer och applikationer för att skapa underlag för potentiella förbättringar av dessa. Detta gör man genom den övervaknings dashboard som öppnas genom en flik i översiktsvyn av en processapplikation (se figur 3.10 nedan).



Figur 3.10 Övervaknings-dashboard



4. Metod

Vårt syfte med denna undersökning har varit, som nämnts ovan (se avsnitt 1.2), att induktivt utifrån studier av användandet av en programvara generera generella designprinciper som tillsammans utgör ett designramverk specifikt anpassat för design av komplexa SaaS-self-serviceprodukter. För att studera användandet av "Barium Live!" har vi valt att utföra en kombination av observation och intervju för att få en bredare bild av interaktionen. Patel och Davidson (2011), Sharp et al (2011), m.fl. menar att detta arbetssätt ger en fylligare bild när informationen som genererats från de olika metoderna sedan slås ihop till resultatanalysen.

4.1 Observationer

För att få inblick i hur "Barium Live!" upplevs av dess tilltänkta användare beslöt vi att göra observationer av nya kunder/användare som introduceras till produkten för första gången. Vi gjorde detta enligt en metod kallad "*Think out loud*" (Krug, 2006; Schneiderman & Plaisant, 2010; Sharp et al, 2011). Genom att sitta med en användare som öppnar produkten för första gången och uppmana denne att hela tiden berätta vad som rör sig i huvudet när personen interagerar med produkten hoppades vi få insikt i vilka delar av designen som personen reagerar och reflekterar över. Genom att observera såväl vad personen säger som vad den gör (stannar upp, blir förvirrad, e.d.) utan att ingripa, annat än för att påminna användaren om att berätta vad denne tänker eller hjälpa användaren i rätt riktning om den kör helt fast, hoppades vi att få insikter om hur produkten upplevs av den tänkta målgruppen första gången de exponeras för den, något denna metod visat sig mycket väl lämpad för i tidigare studier (Krug, 2006; Sharp et al, 2011).

Observationerna gjordes i entimmessessioner med en informant åt gången. Dessa sessioner förlades till informanternas arbetsplatser för att i så hög grad som möjligt minska känslan av att vara i en utsatt situation. Med detta hoppades vi att informanternas utförande skulle bli så naturligt som möjligt. Totalt observerades fyra informanter. Alla testdeltagare instruerades på samma sätt om studiens syfte och upplägg och gavs sen i uppdrag att utföra olika uppgifter i programvaran (se bilaga 1 och 2 för en kopia av manuset som användes för instruktioner resp. uppgifter). Användarna fick en kort introduktion till produktens olika delar inför varje observationsmoment för att få en grundläggande bild av den funktionalitet som finns. Testuppgifterna hade vi tagit fram i samråd med utvecklare och processkonsulter på Barium för att säkerställa att de på ett effektivt sätt innefattade de delar av produkten som var intressanta att studera ur ett self-serviceperspektiv (se bilaga 2). De delar som testades var sådana som Barium har för avsikt att kunna erbjuda sina framtida kunder på ett mer self-service enligt sätt och innefattade processmodelleringen, att arbeta med uppgifter i processer samt att skapa och hantera listor för att hantera aktiviteterna (se kapitel 3).

Under testsessionerna närvarade vi båda två. En av oss fungerade som testledare och gav instruktioner till användaren och hjälpte denne framåt när/om denne körde fast. Den andra observerade och tog anteckningar på sådant som upplevdes vara av särskilt intresse för studien, t.ex. vilka moment som verkade oklara, när fel begicks och hur testpersonen löste uppgifterna.

Testsessionerna spelades också, efter godkännande av informanterna (se bilaga 3), in med en programvara² som spelar in ljud och skärmbilder i realtid, vilket möjliggjorde för oss att snabbt och enkelt gå tillbaka och titta på sekvenser av särskild relevans för undersökningen. Genom valet av "think out loud" metoden fick vi då också möjligheten att höra hur testpersonerna resonerade kring dessa moment.

4.2 Intervjuer

För att komplettera bilden av interaktionen med en komplex self-service produkt som vi fick från observationerna utförde vi i direkt anslutning till dessa intervjuer med var och en av de observerade användarna. Intervjuerna varade mellan 20-30 minuter (beroende på hur mycket tid observationen tagit). De lades upp på ett semistrukturerat sätt efter de teman som vi i teorikapitlet definierade som relevanta ur designsynpunkt för SaaS-self-serviceprodukter, dvs.; *layout, navigering* och *hjälp* (se bilaga 4 för en kopia av det frågeformulär som vi utgick ifrån). Informanten fick först dela med sig av sina generella tankar kring produkten varpå vi utifrån var och ett av de givna temana lyfte fram olika moment från observationen som vidrörde det aktuella temat. Utifrån detta fick informanten frihet att berätta och förklara med egna ord hur denne upplevde situationen och vad det var som hade fungerat dåligt och/eller bra, vilket vi följde upp med relevanta följdfrågor. På så vis kom vi åt värdefulla reflektioner om framförallt produkten, men också design av komplexa self-serviceprodukter i allmänhet.

Schneiderman och Plaisant (2010) menar att intervjuer upplagda på detta vis ger en tydligare bild av specifikt viktiga problem som användare upplever än om man använt sig av helt stängda eller helt öppna frågor. Dessutom leder det aktuella intervjuupplägget ofta till konkreta förslag på förbättringar som användarna identifierat (ibid), vilket kan vara intressant input till vår studie.

Sharp et al (2011) lyfter fram vikten av att hålla intervjuerna neutrala och fria från teknisk jargong för att minska kunskapsgapet mellan intervjuaren och den intervjuade. För att utvinna reflektioner och tankegångar på en högre abstraktionsnivå än rent teknisk formulerade vi våra frågor så att den tekniska kunskapen inom området inte skulle påverka det slutgiltiga resultatet.

Intervjuerna spelades in och transkriberades för att underlätta analysen och tydligare kunna se att de slutsatser vi härleder från intervjuerna är motiverade.

4.3 Urval

Eftersom "Barium Live!" är en produkt anpassad för professionellt användande i yrkessammanhang krävdes en viss förståelse för processhantering och vi valde därför att fokusera våra observationer på användare från de företag som Barium idag har bland sina kunder. För att undvika att användarna hade lärt sig att hantera programvaran, vilket är en direkt förutsättning för att utvinna förståelse om dess prestanda som self-serviceprodukt, valdes testpersoner bland anställda som ditintills inte fått utbildning i eller börjat använda produkten. För att täcka in alla aspekter av programvaran valde vi att testa med personer som sannolikt skulle hamna i en

² Camstudio, finns för gratis nedladdning på <http://www.camstudio.org>

position där de modellerade, skapade applikationer och arbetade i de skapade applikationerna. Barium tog i samråd med en av deras kunder fram testpersoner som uppfyllde dessa kriterier.

Med tanke på studiens kvalitativa art, den begränsade rekryteringsbasen, tidsutrymmet samt valet att inhämta information genom triangulering menar vi att informationsinsamling med fyra informanter är fullgott för vår studie. Detta stöds av Schneiderman och Plaisant (2010) som menar att antalet testpersoner i en användbarhetsstudie genom observation kan vara ganska få pga. att man lägger högre fokus vid att utvärdera funktioner och uppgifter istället för att sätta fokus vid olika statistiska summeringar som t.ex. tidsåtgång eller förekomsten av fel. Även Nielsen (2000) och Krug (2006) stödjer detta och menar båda att de första tre användarna man testar kommer att hitta i princip alla stora fel som finns i produkten och att man därför inte vinner mycket på att testa ytterligare användare om man inte delar upp observationerna i olika iterationer och förändrar applikationen mellan varje iteration. Ett sådant upplägg var tyvärr inte möjligt i denna studie då vi inte ägde befogenhet att göra några ändringar i produkten. Upptäckterna från ytterligare observationer hade därför varit mycket begränsade och inte stått i proportion till den tidsåtgång och det arbete som de inneburit (Nielsen, 2000; Krug, 2006). Genom att avgränsa oss på detta sätt hoppades vi istället att få en omfattande och djup genomgång av programvarans olika delar och funktioner.

4.3.1 Presentation av urvalsgruppen

Nedan följer en kort presentation av samtliga informanter, vilka hålls anonyma under vår framställning. Informanterna varierade i ålder i spannet 30-60 år. Kön presenteras inte då vi inte anser det relevant för de framkomna resultaten.

Informant 1: Användbarhetsspecialist, arbetar som projektledare för ett applikationsutvecklingsteam.

Informant 2: IT-projektledare, ansvarig för att utveckla generella projektmodeller för alla projekt inom organisationen.

Informant 3: Systemutvecklare, men har senaste åren arbetat som chef i olika roller. Jobbar just nu som sektionschef för 22 anställda.

Informant 4: Verksamhetsutvecklare. Arbetar just nu med att förvalta system inom organisationen.

Gemensamt för samtliga testpersoner är att de lägger ca halva arbetstiden på att utföra arbetsuppgifter med hjälp av dator. Dessa arbetsuppgifter omfattar främst mail, dokumenthantering och förvaltning. Samtliga spenderar även ett antal timmar varje dag vid datorn på sin privata tid och där är mail den vanligaste sysselsättningen.

En informant hade flyktigt fått ”Barium Live!” visat för sig och två hade fått produkten beskriven för sig men inte sett den. En hade aldrig fått någon presentation av produkten överhuvudtaget. Ingen av informanterna hade själv arbetat med eller navigerat i produkten.

Dessa testpersoner skulle med stor sannolikhet kunna stöta på ett verktyg som "Barium Live!" i deras arbete tack vare deras yrkesroller inom organisationen.

Mot bakgrund av detta bedömdes samtliga informanter uppfylla de uppsatta kraven.



5. Resultat

Vår undersökning har resulterat i en rad generella reflektioner om användningen av en SaaS-self-serviceprodukt och även konkreta förslag från informanterna på hur olika situationer i användandet i deras tycke borde ha designas. Genom de observationer och intervjuer vi utförde fick vi en bild av de saker som användarna ansåg vara centrala och indikationer på vilka designgrepp som bör inkluderas och undvikas i vårt kommande ramverk.

Vi kommer nedan presentera resultatet av undersökningen i form av citat och iakttagelser under samma teman som vi presenterat gällande design i teoridelen (se kap 2). Intill varje citat har vi i parantes angivit om det kommer från en intervju eller observation samt vem av de fyra informanterna som kommenterat.

5.1 Layout

Vissa av informanterna hade svårigheter med att förstå den övergripande strukturen i "Barium Live!" och förhållandet mellan platser och processer. Detta visade sig när informanterna kom till startsidan och blev konfunderade när de fick en lista med platser presenterade för sig. Detta gjorde att en av informanterna skapade en ny plats istället för att gå in i den befintliga platsen när denna skulle skapa en ny process. Informanten uttryckte sig så här när denna reflekterade över detta;

"Jag kanske skapar en plats och inom den platsen kan jag skapa många processer, jag vet inte."
(observation, informant 4)

Samma informant fördjupar sedan sina tankar kring detta och vad som var svårt med modelleringsuppgiften;

"Det var ju inte momentet, men man måste ju förstå vad en plats är och vad en process är, man måste förstå begreppen." (intervju, informant 4)

Flera av informanterna påpekade att objekt som de trodde utförde liknande saker fanns placerade på olika ställen i produktens olika delar. En informant beskriver den övergripande placeringen av liknande objekt i de olika vyerna så här;

"Det har ju varit väldigt inkonsekvent, det borde man ju verkligen kolla igenom så att allting ligger på samma sida på varenda skärmbild. Och att skärmbilderna är uppbyggda på samma sätt så långt det går åtminstone." (intervju, informant 1)

Flera av informanterna hade t.ex. svårt att hitta "skapa ny process"- knappen (se figur 3.4) när de skulle börja modellera en process. Några av informanterna beskriver det så här;

"Det var ju inte omedelbart jag fick syn på den där borta, det tog en stund innan jag hittade den skapa process där borta. Jag kände själv att ögonen flackade runt en del innan jag hittade rätt."
(intervju, informant 2)

"Inte så lätt att hitta" (observation, informant 3)

"Tyckte inte knappen för ny process var så bra placerad" (observation, informant 2)

"Men nytt tycker jag det borde vara på ett lite mer centralt ställe och alltid på samma ställe när man ska börja med någonting." (intervju, informant 1)

"Första gången var det ju det här med börja rita processer då låg de ju längst ut till höger och man hittar dom inte intuitivt, direkt så ser man dom inte." (intervju, informant 2)

Även funktionen för att skapa nya listor var för de flesta av informanterna svårt att hitta. Flera av dem upplevde att detta menyalternativ var felplacerat då det finns långt till höger i den aktuella vyn, placerat i en drop-down meny (se figur 3.9). En av informanterna beskriver det så här;

"Den [funktionen för att skapa nya listor] borde ligga mycket mer centralt, [...] bara rena turen att jag råkade trycka på denna först här nu" (observation, informant 1)

Samtliga informanter tyckte att "klar"-knappar i formulärvyerna låg ologiskt placerade. De menade att istället för att placera dessa precis över formuläret borde de finnas placerade under detta, vilket de menade skulle ge ett mer naturligt flöde. Informanterna menade att det är onaturligt att först fylla i ett formulär uppifrån och ned, för att sedan behöva gå upp igen för att skicka ärendet vidare. En informant uttrycker sig så här;

"Ja, på något sätt om man tänker som man jobbar uppifrån och ner, eller från vänster till höger, så skulle man ju komma ner på slutet och så skulle man inte gå upp igen, utan 'avsluta ärendet' ska ju ligga sist på ditt formulär." (intervju, informant 1)

Samtliga informanterna hade svårt att i modelleringsdelen av "Barium Live!" rita ut pilar mellan objekt. De förväntade sig att pilarna, precis som alla andra tillgängliga objekt, skulle finnas i paletten med BPMN-objekt till vänster (se figur 3.6). En av informanterna fick frågan om denna hade föredragit att rita ut pilarna som man gör i dagsläget eller om man skulle kunna välja ett alternativ i paletten till vänster, varpå denna svarade:

"Om jag skulle välja på de här två metoderna så skulle jag ju föredra just det här att linjen kommer när du har valt ut den att rita, sen är det ju bra när man kan verktyget att du vet att bara klicka i mitten och nästa så länkar den de två." (intervju, informant 1)

En annan informant kommenterar detta moment;

"Pilarna var inte helt lätta att hitta." [...] Det här var ju ändå irriterande, hoppas jag är inne på linjer jag, är ju inte helt hundra. [...] Så här svårt ska det inte vara att hitta en pil" (observation, informant 2)

När informanterna fått veta hur momentet gått till, uttryckte de bl.a. följande reflektioner om detta moment;

"Det är inget som framgår att det går till på det sättet." (observation, informant 2)

"Väldigt bra när man lärt sig det, ment inte intuitivt första gången" (observation, informant 4)

"Det här att man väljer i mitten på de här pilarna. Det var ju inte van vid" (intervju, informant 3)

"Dom hade jag ju velat att de låg där med de andra objekten. Det är ju liksom samma grej för mig [...] jag tycker att pilarna också är en symbol som man ritar så jag förstår inte varför dom skulle ligga på ett helt annat ställe." (intervju, informant 1)

Ett annat moment som var problematiskt för informanterna gällande att rita pilar mellan objekt, var att få pilarna böjda på det sätt som uppgiften manade till. Fler av informanterna försökte dra i pilarna på olika sätt eller leta i pilobjektens alternativ för att där hitta rätt funktion. Några av informanterna hittade rätt menyval men förstod då inte vad termerna betydde och valde därför inte att testa några av dessa. En av informanterna beskriver just denna osäkerhet så här;

"Men jag hade ju inte koll på begreppen heller riktigt då, vad "polyline" var och sånt där.. och det, hade man vetat det så hade det kanske vart enklare. Men jag tycker ändå att det var lite, jag hade velat kunna dra dem fram och tillbaka, inte hålla på och byta vad det är för slags linje utan bara ta tag i hörnet och dragit dom dit ner och så blir det en sån pil istället." (intervju, informant 1)

Ingen av informanterna klarade detta moment utan vägledning. Några utav dem beskriver momentet så här:

"Svårt att få pilarna böjda, inte intuitivt" (observation, informant 3)

"Vanligtvis drar man ju i de där fyrkanterna för att ändra, det kan man tydligen inte göra här" (observation, informant 4)

Användningen av en menypalet i modelleringssteget gav blandad feedback från informanterna. De flesta förstod att objekten där placerades på ritytan genom en "drag-and-drop" princip, samtidigt som en av informanterna tyckte att det hade varit naturligt om man också kunde välja dem i paletten och sen måla ut dem på ritytan.

"Man kan oftast placera saker på ritytan bara genom att klicka på dem." (observation, informant 4)

En informant upplevde de tomma flikarna i anslutning till paletten i processmodelleringen ("Process data objects" och "Object templates", se figur 3.6) som störande då denne öppnade flikarna i hopp om att där hitta de verktyg denna sökte. Informanten uttrycker sig så här;

"Ska de vara tomma? Meningslöst att de ligger här och stör!" (observation, informant 3)

En liknande reflektion kom från en informant som letade efter ett menyalternativ för att modifiera ett objekt som ritats ut på ritytan i modelleringssteget. Denne valde att testa "konfigurerings"-knappen varpå inget hände. Informanten tittade runt i gränssnittet för att se vad som ändrats men upptäckte då att ingenting har hänt.

"vad ska man ha den till om den inte gör någonting?" (observation, informant 1)

Samtliga informanterna hade i modelleringssteget svårt att placera BPMN-objekten swimlanes i pooler³ på rätt sätt. Informanterna släppte felaktigt lanes mitt i de pooler de skapat vilket gör att de blir felplacerade. Ingen av informanterna förstod på egen hand att man för att utseende och funktion ska bli rätt måste släppa nya lanes över poolens namn. En av informanterna letade länge efter vart man ändrar detta utseende;

"Det här var inte intuitivt får jag säga [...] varför ska det vara så" (observation, informant 1)

Flera av informanterna hade synpunkter på hur listor används i de olika delarna av "Barium Live!". Dels uttrycktes det åsikter om själva layouten av listorna, dess funktionalitet och några av informanterna visade också tecken på att inte riktigt förstå vad listorna innehöll. De upplevde att det fanns bristfälliga förklaringar över vad listorna innehöll. Detta visades tydligt när flera av informanterna, efter att öppnat "mina aktiviteter vyn" (figur 3.9), stannade upp för att, efter uppmaning från oss, ta sig tid att försöka förstå sig på vad dess funktion var. En informant uttryckte sig så här när denna försökte förstå vyns funktionalitet;

"Hur den ska användas kan man inte intuitivt förstå. [...] Dom som jobbar i samma flöde lägger information på samma ställe?" (observation, informant 3)

Informanten ansåg inte vyn vara självförklarande och hade olika resonemang om både den aktuella vyn men också om de andra listornas funktion och betydelse i "Barium Live!". Flera av informanterna hade svårt att avgöra vad de olika listorna faktiskt innehöll. En av informanterna letade i en lista bestående av befintliga processer (figur 3.4) för att hitta till kör-delen av barium, då denna inte förstod vad listan visade;

"trodde det var hit jag gick" (observation, informant 2)

Samma osäkerhet kring vad listorna innehåller beskriver en annan informant;

"Nej det var inte heller så speciellt tydligt. Dom var ju rätt lika varandra. Så det hade man ju kunnat märka ut lite bättre." (intervju, informant 1)

Några av informanterna resonerade kring aspekter av utformningen av "Barium Live!" som är relaterade till produktens tillgänglighet för användare med fysiska begränsningar. En av informanterna har svårt att i sitt dagliga arbete använda musen för att utföra moment vid sin dator, pga. musarm och använder därför i så hög grad det stöds snabbkommandon istället. Informanten försökte följaktligen utföra delar av momenten i användartestet genom snabbkommandon som denna var van vid att använda i liknande produkter och märkte då att vissa fungerade likadant i "Barium Live!". Informanten resonerar kring detta;

"Det behövs kortkommandon, det kan hända att dom har det men om dom inte har det är det otroligt viktigt" (observation, informant 1)

³ BPMN-notationen påbjuder att man ritat upp sin process som en "pool" innehållandes avgränsade "swimlanes" eller "lanes" för varje användartyp som är involverad i processen. Se t.ex. Silver (2011).

En annan informant med nedsatt syn hade svårt att läsa den lilla texten som fanns på flikarna i modelleringssteget. Informanten försökte öka storleken på dem, men ökar du storleken på ritytan istället och hade därför svårt att förstå vart vissa verktyg var placerade i paletten (se figur 3.6).

Flera av informanterna hade åsikter om de namngivningar som används på objekt och menyalternativ i ”Barium Live!”. En informant menar att begreppen ”poster” och instanser” som används i produkten inte är logiska för den målgruppen produkten idag har;

”De begreppen är inte anpassade till processteori eller verksamhetsmänniskor” (observation, informant 4)

En annan informant var inte helt säker på vad några av objektens namn betydde, eller motsvarade inom BPMN. Därför hade denne svårt i modelleringssteget att avgöra om det var objekt som innefattades av den aktuella uppgiften. Ytterligare en annan menade att benämningar på funktioner som används för att spara ibland byttes ut trots att de hade samma ikoner och utförde, i informantens ögon, likande saker. Informanten reflekterar så här om dessa funktioner;

”Det var inte konsekvent på något sätt, utan ibland hette det spara och ibland hette det starta” (intervju, informant 1)

5.2 Navigering

Samtliga informanter upplevde stora problem med navigeringen i applikationen. Detta berodde framförallt på att de saknade de en tydlig översikt över applikationens olika delar då den existerande drop-downmenyn inte upplevdes som tydlig nog. Vad som också blev uppenbart under observationen var behovet av en tydlig indikation på i vilken del av applikationen de för tillfället befann sig. Några kommentarer löd:

” Vad finns det för olika delar i hela produkten? Är det bara de här tre eller? [Pekar mot ikonerna Modellera – Bygg – Kör på skärmen] [...] de skulle ha legat överst i menyn eller på sidan här så att man alltid kan komma dit, vart man än är i gränssnittet” (intervju, informant 1)

”Det [applikationen överlag] var inte speciellt intuitivt, det var väldigt svårt att hitta tyckte jag [...] Jag tänkte att det borde ligga någon fast meny någonstans här, men det gjorde det ju inte. Då fick man liksom bara försöka scanna uppifrån och ner här och se.” (intervju, informant 2)

”Jag tycker ju verktyget var, för mig iallafall, lite svårt att navigera i.” (intervju, informant 3)

Informanterna klarade utan större problem att från startguiden för platser (se figur 3.2) ta sig till modelleringssteget med hjälp av de tre ikonerna ”Modellera – Bygg – Kör”. Dessa ikoner upplevde de alla som begripliga och överskådliga;

”Ja, den är ju bra, den hade man ju där... Modellera, bygg och kör. Den fanns ju intuitivt och då använde de visuella symboler också och då underlättar det ju redan ...” (intervju, informant 4)

När de sedan fick uppgiften att ta sig från processmodelleringen till ”Kör”-steget fastnade samtliga informanter. Alla utom en försökte leta sig tillbaka till startguiden för platser för att byta steg. En informant upplevde att det var;

”Väldigt svårt att hitta tillbaka till den första bilden vi var på... det hade varit bra om man nådde de här tre stegen [pekar på de tre ikonerna på startguiden för platser] även inne från processen” (observation, informant 2)

Detta var en åsikt som stöddes av samtliga informanter;

”Jag skulle vilja ha att dom [”Modellera – Bygg – Kör”-ikonerna] hängde på hela tiden, att man alltid såg dem i någon övergripande meny.” (intervju, informant 1)

Två informanter använde sig upprepade gånger av applikationens (inte webbläsarens) bakåtknapp för att försöka komma till startvyn och därifrån välja ”Kör”-ikonen. Denna bakåtknapp finns i anslutning till en form av breadcrumbs som visar var i hierarkin man befinner sig. Dock ledde den bara till processöversikten för den just modellerade processen (en vy som informanten i det läget ännu inte hade varit i) och när de tryckte bakåt ytterligare en gång kom de återigen till modelleringsvyn för processen (ett steg fram i hierarkin), inte ”Startguiden för platser” som informanterna hoppats. Detta skapade förvirring kring var de egentligen befann sig och var på väg;

”Tycker inte jag kommer bakåt... jag är fast i ritverktyget” (observation, informant 3)

”Det måste ju finnas någon form av förstasida?!” (observation, informant 1)

Bara en informant lyckades, efter en stunds letande, använda drop-downmenyn (som ger tillgång till applikationens alla delar, se figur 3.3) på det avsedda sättet. Trots att flera av informanterna klickade på menyn och fick fram navigeringsalternativen valde ingen av de övriga att navigera sig med hjälp utav den. Detta kommenterades, efter att vi påtalat dess existens och funktionalitet, så här;

”Ja, men det förstår man ju inte först. Det måste man ju jobba några veckor innan... Ja, där finns de ju. Men jag tycker att det är osynligt för mig. Alltså... det måste vara tydligare. Det är ju olika; vissa människor är text och andra är ju bild. Jag är ju bild, det har jag märkt här också” (intervju, informant 1)

”Det såg jag ju inte omedelbart allfall, så det var väl därför jag bara backade då.” (intervju, informant 3)

”Det var inte heller speciellt logiskt.” (intervju, informant 2)

När den informant som under observationen hittade drop-downmenyn under intervjun uppmanades att ta sig från den vy systemet för tillfället befann sig i till ”Modellera” lyckades denne inte ta sig fram och hittade inte menyn igen;

”... till modellera... ja... nej... det är kört... [skratt] Men jag kanske får stänga den och sen får jag kanske... [läser] ”Mina aktivitetslistor”... nä... jag har svårt för det. Så att det är inte intuitivt...” (intervju/observation, informant 4)

En informant misstog knapparna ”Spara” (som sparar ifyllt data) och ”Utför åtgärd” (som klarmarkerar uppgiften) i ”Kör”-steget och förstod då inte vilken del denne befann sig i;

”När man var inne i den här körlistan så ena gången när man hade gått igenom grejerna då hamnar man i en vy och nästa gång när man gått igenom grejerna då låg det bara kvar och så fick man gå upp där på ’gör åtgärd’ för att komma vidare. Den här listan där man hela tiden såg att det här är gjort och det här är väntande, den hade man ju velat komma till varenda gång.[...] Det var väldigt snurrigt här så man viste inte vad man skulle... Vad man skulle bege sig någonstans.” (intervju, informant 2)

5.3 Hjälp

Trots att alla informanter i vid något tillfälle i olika delar av systemet körde fast och inte kunde komma på hur de skulle lösa vissa problem var det ingen som använde länken till hjälpdokumentationen som finns i alla skärmbilder med rubriken ”(?) Lär dig mer om den nya Barium Live! releasen” (se figur 3.1-3.2 & 3.4-3.10). Inte heller när vi uppmanade dem att leta efter hjälp i gränssnittet hittade de länken. När vi visade den för informanterna under intervjuerna kommenterades det;

”Där var ju ett frågetecken, låg den med på alla bilder? ... [Läser] ’den nya Barium Live! releasen’ ... Det är ju inte det man förväntar sig ... Det var ju konstigt för nu när jag ser frågetecknet där så hade jag nog... men jag tänkte inte på det...” (intervju, informant 2)

”Jag letade efter den men jag fick inte ögonen på den. Det var ju där vid navigeringshjälpen, jag försökte ju rita var det nog, försökte hitta hjälp men fick inte ögonen på dem helt enkelt.” (intervju, informant 3)

En hade noterat länken under processmodelleringen men motiverade sitt val att inte klicka på den med följande:

”Jag ville ju ha ett avsnitt kring just det specifika om linjen och då vill man ju oftast när man pekar på objektet få fram hjälpen här då i så fall. När jag klickar på ett objekt vill jag kunna se hjälp, exakt specifikt. Angränsat. Här var det ju generellt, antar jag. Så det här var ju inte anpassat för när man modellerar. För när man modellerar är det specifika hjälpavsnitt som man behöver titta på, inte det generella.” (intervju, informant 4)

Även en annan informant uppmärksammade länken men avfärdade snabbt dess betydelse då denna tolkade det som att där bara fanns information om den nya releasen och inget annat;

”Varför finns information om releasen där? Helt onödigt ju!” (observation, informant 1)

Alla informanterna framhöll hur de i lägen där de ska lära sig att hantera program på egen hand utan utbildning föredrar att prova sig fram istället för att först sätta sig in i dokumentation och instruktioner kring applikationen, för att först söka hjälp när problem väl uppstår. Några av informanterna resonerar;

"Går jag på utbildning så får man ju alltid använda manual och sånt där men... och då går man ju igenom det. Men om jag själv bara börjar jobba i ett system så kollar jag inte i hjälpen förens jag känner att jag behöver." (intervju, informant 2)

"Jag ger mig nog helst på verktyget tror jag. Jag har nog för lite tålmod ibland. Så jag ger mig på verktyget i regel och testar. Ofta finns ju såna här hjälpfunktioner i alla system, men de är så in i helsike invecklade och om sig så man klarar inte av att fånga upp alla de där... så att... de är inte enkla dom. Så därför är det viktigt att när man står i den funktionen eller den åtgärden man ska göra – att man får upp just den informationen som är giltig just då, så att man inte får upp något generellt så att man måste navigera. Det går inte. Man måste få upp det som är... det kan till och med vara så att man bara står där så kommer det upp en sån där liten "bluppa" ibland." (intervju, informant 4)

"Man provar själv och så ser man - det här fattar jag inte - och så läser man om det området då." (intervju, informant 3)

Att minimera användandet av hjälpdokumentation var ett gemensamt mål för informanterna, vilket understöds av resonemangen ovan. Det visade sig att detta var pga. att informanterna tyckte att det ofta är svårt att hitta det man söker i hjälpdokumentationen och att den ofta är allt för omfattande och svår att sätta sig in i. Några av informanterna uttrycker dessa åsikter så här;

"...för jag tycker inte att hjälp brukar vara så bra. [...] Jag tycker ofta att det är så svårt att hitta det man vill ha hjälp med så att man blir bara irriterad på den där hjälpfunktionen." (intervju, informant 2)

"för läser man manualen får man ju läsa om allt." (intervju, informant 3)

Informanterna fick svara på frågan om på vilket sätt de helst skulle vilja ta till sig hjälp om de själva fritt fick välja. De flesta tryckte på betydelsen av inbäddad hjälp, som man inte aktivt behöver aktivera utan som visas för en och som är relevanta för det man håller på med för tillfället. Så här svarade några av informanterna på hur de skulle vilja ta till sig hjälp;

"Jag hade nog skummat quick start guiden först och sen hade jag nog försökt gå in på den riktiga hjälpen, den mer detaljerade hjälpen och sökt då, på pilen t.ex." (intervju, informant 2)

"I vissa lägen hade jag nog önskat mig det [hjälp som "poppar" fram i gränssnittet], när man ritar... 'vill du koppla två objekt', typ... Det exemplet kunde man ju tänka sig att man kunde få som förslag, men har man det som poppar upp så ska det ju kunna stängas av för det blir ju irriterande när man har lärt sig verktyget. Men som ny användare känner man absolut att det hade nog vart ett stöd, att få pop-ups då, som föreslår åtgärder." (intervju, informant 3)

En informant tog också upp introduktionsfilmer där väsentliga funktioner går igenom;

"Det brukar jag titta på om det finns, men inte alltid. Och om jag behöver hjälp så ska det ju finnas där. Och det ska vara bra producerat. Det är jätteviktigt." (intervju, informant 1)

Flera av informanterna återkommer till att det kan finnas ett behov av olika slags hjälp i olika lägen. De syftar då både till olika moment i användningen samt hur länge de har använt sig av produkten. En informant sammanfattar behovet av olika typer av i hjälp;

"Det borde finnas hjälp i flera nivåer"(observation, informant 1)

Gemensamt är att informanterna vill ha hjälp som mer eller mindre automatiskt kommer fram när de utför specifika uppgifter i processmodelleringen, men betonar vikten av att det aldrig får kännas som att de är i vägen.

De tycker också att det bör finnas en omfattande manual eller instruktion som man kan vända sig till om man verkligen kört fast eller om det är övergripande logiska kopplingar man inte förstår, men påpekar samtidigt att de ofta upplever dem problematiska och svårlästa;

"Hade det vart senare och jag hade jobbat med det i flera månader så hade jag ju inte gått till den här lathunden. För då förväntar man ju sig att man kan det som står där. Då hade jag nog gått på hjälp direkt." (intervju, informant 2)

"Det ena utesluter ju inte det andra egentligen. Det måste kanske finnas fullständig dokumentation så man kan sätta sig på sin kammare och läsa för att förstå på djupet... men egentligen ska man inte behöva det... egentligen ska man inte göra det." (intervju, informant 4)

"Givetvis ska du ju kunna slå upp också men nackdelen när du går upp och slår upp, då ibland är det ju som i Word eller Excel, då måste du komma ihåg de begreppen dom beskriver..." (intervju, informant 3)

"Ni behöver ju ha hjälp precis bredvid den ritobjektet då. Får ni ju ha en sån liten ikon också kan det ju komma upp en ruta med hjälp då. Det är ju lokal hjälp, men så borde ni ju ha nån mer övergripande hjälp för sidan också nån yttre hjälp för hela gränssnittet eller för hela kommunikationen." (intervju, informant 1)

En av informanterna sammanfattar hur denne anser att hjälpfunktioner ska fungera;

"Det ska ju vara något som är snabbt, måste det vara. Också ska det vara, liksom nåt man kan ta till sig snabbt. När jag gick interaktionsdesign så hade läraren nån presentation så kunde han bläddra så här [illustrerar hur hon bläddrar genom en hög med papper] 'Jag förstår ju', sa han, 'Jag förstår ju om jag bläddrar igenom den här så här, vad den innehåller'. Och så menade han att det skulle vara. Det ska ju vara så att man inte behövde anstränga sig, utan bara får in det." (intervju, informant 1)

Ingen av informanterna nämnde Wikis eller communities (se 2.3.2) på frågan om vilken hjälp de föredrar, däremot nämnde flera att de skulle fråga en kollega om de körde fast med applikationen och det fanns någon de kunde vända sig till;

"Jag skulle nog gå på hjälp... i verktyget först... och steg två hade nog varit att fråga någon kollega" (intervju, informant 3)

Som nämnts i avsnitt 5.1 hade några av informanterna svårt att förstå vissa av objektens namn för att de inte var insatta i BPMN-termologin. En informant tyckte att man kunde börja modelleringsdelen med att beskriva de begrepp som används i systemet för att kunna förstå vad som behövs osv.:

”men det är kanske det man skulle börjat med, de här begreppen betyder följande, alltså som en sån där självstudie som man kan välja bort när man sedan kan dem då för det vill man ju inte ha varje gång.” (intervju, informant 3)

Samtliga informanter upplevde som vi såg i 5.1 någon svårighet med att rita något av de objekt som processen skulle innehålla, med att modifiera objekten till det eftersökta tillståndet och hantera ärenden i processapplikationerna. Efterfrågan efter tooltips var återkommande;

”Lite tooltips alltså, sådana små rutor som dyker upp när man håller över” (intervju, informant 2)



6. Resultatanalys

I detta kapitel presenterar vi de designprinciper som vi genom vår studie kommit fram till är extra viktiga för lyckad design av komplexa SaaS-self-serviceprodukter. Det är dessa som tillsammans utgör vårt ramverk, vilket i sin tur utgör svaret på vår frågeställning. Principerna presenteras under var och ett av de teman som gått igen i såväl litteraturgenomgång som intervjuunderlag och resultatpresentation. Var och en av principerna presenteras i tre steg; 1) Ett namn som återges med fet stil. 2) En förklaring av principen som återges kursivt. 3) Det bakomliggande resonemang som visar hur våra upptäckter från observationer, intervjuer och litteraturgenomgång stödjer principen. Denna förklaring återges i normal stil.

Vissa principer presenteras här under ett tema medan de motiveras och förklaras helt eller delvis med hjälp av upptäckter som i resultatpresentationen togs upp under andra teman. Detta är för att problemen kan vara något som t.ex. upplevdes som ett problem i layout eller navigering, men vi väljer att lösa problemet med någon form av hjälp, eller vice versa.

6.1 Layout

Skapa konsekventa gränssnitt

Skapa ett konsekvent uppbyggt gränssnitt, där likadana gränssnittsobjekt placeras likadant i alla vyer. Genom att säkerställa att gränssnittsobjekt placeras konsekvent genom applikationens alla delar ökar man möjligheterna för att användaren själv snabbt förstår alla vyer i applikationen.

I resultatet såg vi hur flera informanter upplevde att vissa objekt som de uppfattade som väsentligt lika, t.ex. ”skapa”-knappar för processer och listor, placerades på olika platser i gränssnittet i olika vyer (se avsnitt 5.1). Schneiderman och Plaisant (2010) beskriver att det är viktigt att hålla gränssnittet konsekvent för att användare ska kunna känna igen sig i alla delar av en produkt och få möjligheten att bli familjära med vad funktionerna gör och betyder (se avsnitt 2.3.2). Komplexa SaaS-produkter kan innehålla omfattande funktionalitet där vissa funktioner rent betydelsemässigt liknar varandra. Med tanke på att användaren så starkt relaterar dessa funktioner som lika bör de placeras på samma ställe och på så sätt, i användarnas ögon, skapa ett mer konsekvent gränssnitt.

Skapa ett naturligt flöde i applikationen

Genom att, på ett naturligt sätt, visa för användaren vad denna ska utföra i nästa steg av en uppgift minskar man den totala ansträngning som användaren upplever vid utförande av uppgiften. De steg som användaren måste gå igenom för att slutföra en uppgift ska vara utformade så att de utgör ett naturligt flöde där användaren tydligt ser var interaktionen börjar och slutar samt upplever att denne går rakt mellan dessa punkter - inte fram och tillbaka mellan olika steg.

Flera av informanterna upplevde att de i vissa situationer var tvungna att slutföra uppgifter på ett onaturligt sätt, genom att gå tillbaka till en del av vyn som de redan passerat. Som en informant beskriver det är det naturligt att man arbetar från höger till vänster, samt uppifrån och ned. Som

vi visat i teoridelen (se avsnitt 2.3.2) stämmer Tidwell (2011) in i detta och menar att man bör placera all interaktion som ska ske i gränssnittet i en visuell bana för att försäkra att användaren inte behöver gå fram och tillbaka i det för att slutföra en uppgift. Därför ska funktioner för att inleda en uppgift vara stora och tydliga och placeras i början av den visuella banan, medan funktioner för att avsluta uppgifter alltid placeras sist i den visuella banan. Som vi sett i vår fallstudie kan avancerade funktioner innefatta ett flertal steg som användaren måste utföra för att slutföra en uppgift. Därför blir det, för att uppnå goda förutsättningar för self-service i produkter som tillhandahåller avancerad funktionalitet, viktigt att all interaktion följer denna visuella bana.

Använd språk som passar den tilltänkta användargruppen

Namngivningar och beskrivningar av objekt i gränssnittet ska vara utformade för de tilltänkta användarna. Genom att anpassa dessa utefter användarnas erfarenheter och kunskaper blir applikationen mer självförklarande och lättare för användarna att förstå.

Som resultatet visar fanns det vid flera tillfällen oklarheter bland informanterna om vad vissa objekt var för något. Bl.a. hade samtliga informanter svårt att hitta ett sätt att böja pilarna som uppgiften uppmanade till. Denna svårighet visade sig bero på de namngivningar som används i produkten för att skapa böjda pilar. Den namngivning som användes motsvarade inte vad någon av informanternas förväntade sig och därför användes inte denna funktion. Ytterligare namngivningar som var okända för de flesta informanter var användningen av instans och post, vilket inte motsvarade deras tidigare erfarenheter och gjorde att de hade svårt att förstå deras betydelse. Därför är det, precis som visats i avsnitt 2.3.2, viktigt för användare när de ska lära sig nya gränssnitt att man i applikationen använder begrepp som stämmer överens med tidigare erfarenheter från andra gränssnitt och tidigare kunskap om den aktuella domänen (Schneiderman & Plaisant, 2010; Sharp et al., 2011). Alltså; Anpassa benämningar och namngivningar efter de tilltänkta användarna och deras erfarenheter

Följ konventioner

Följ befintliga konventioner vid utformande av applikationens layout och funktionalitet. Genom att ta till vara på accepterade konventioner och best-practice säkerställer man att användare som har erfarenhet av liknande produkter snabbt känner igen sig i den nya applikationens layout och snabbt kan förstå och börja använda dess funktionalitet.

Flera av informanterna visade både under observationerna och under intervjuerna att de försökte tillämpa erfarenheter från tidigare system de använt, när de försökte lära sig funktionaliteten i "Barium Live!". Bl.a. visade detta sig genom att flera informanter använde gängse kortkommandon som de var vana att bruka i andra applikationer, för att utföra funktioner i "Barium Live!". Detta tyder på att det finns en konsensus bland informanterna om vad som är naturligt att man kan göra i denna typ av produkter, vilket i sin tur tyder på att det finns en vida accepterad konvention mellan applikationer att låta användare använda samma kortkommandon för att utföra liknande funktioner. Vikten av detta stöds, som vi visat i avsnitt 5.1, av en informant som under observationen uttrycker att det är vanligt att kortkommandon stöds av

liknande programvaror och att det är viktigt att det finns tillgängligt. En informant försökte klicka in objekt på ritytan i modelleringssteget (se figur 3.6) istället för att dra in dem, vilket inte fungerade. Informanten tog senare upp detta under intervjun och påpekade att det var konstigt att man inte kunde klicka in objekt, då det enligt dennes uppfattning är vanligt i liknande program. Som visats i resultatdelen hade flera informanter svårt att se länken till hjälpdokumentationen som finns i flera vyer av applikationen (se figur 3.1). Detta beror i vår mening till stor del på att denna länk var gråfärgad, vilket i liknande applikationer innebär att en funktion är tillfälligt inaktiverad. Valet att gråmarkera länken följer alltså inte den praxis som finns vid design av liknande produkter. Som presenterats i avsnitt 2.3.2 ökar användningen av konventioner användarnas effektivitet, pga. att de kan utföra invanda moment som fungerar i de flesta applikationer (Tidwell, 2011). Men de kan också, som visats i exemplen ovan, stjälpa en användare i de tillfällen då konventionerna inte följs. Användaren blir i dessa lägen förvirrad och tvingas fundera på hur moment som annars görs på vana egentligen utförs (ibid.). Som resultatet visar upplevde också samtliga informanter att vissa objekt var ologiskt placerade och hade därför svårt att hitta dem. Placering av gränssnittsobjekt omfattas också av rådande konventioner och påverkar vart användare förväntar sig att hitta objekt i ett gränssnitt. Därför bör man sträva efter att även följa de konventioner som finns för placering av gränssnittsobjekt, för att uppnå ett logiskt uppbyggt gränssnitt, vilket är fundamentalt för att uppnå god self-service.

6.2 Navigering

Översikt + positionsmarkering

För att användaren ska förstå var denne befinner sig i applikationen och vad hon kan göra i den måste applikationen förses med en tydlig och lättåtkomlig översikt över alla applikationens delar eller, i sekventiella gränssnitt, alla dess steg. Denna översikt ska vara fullt synlig oavsett var i applikationen användaren befinner sig. Lika viktigt är att användarens nuvarande position i applikationen märks ut i förhållande till denna översikt.

Som resultatpresentationen visade hade samtliga informanter svårt att navigera i applikationen. Detta hade till stor del att göra med att informanterna inte kunde hitta någon del i gränssnittet som tydligt visade applikationens hela funktionalitet. Den drop-downmeny (se figur 3.3) som finns idag upplevdes osynlig och otydlig. Ingenting i den markerade heller vilket steg informanten för tillfället befann sig i. Allt detta rimmar dåligt med vad Koblanck (2003), Krug (2006), Schneiderman och Plaisant (2010) och Tidwell (2011) alla var eniga i; att användaren hela tiden måste veta 1) Var denne befinner sig, 2) Vad den aktuella applikationen innehåller och 3) Hur användaren kan ta sig till det den söker (se avsnitt 2.3.2).

För att en komplex produkt ska kunna fungera enligt en self-serviceprincip är det av största vikt att användarna snabbt och enkelt kan finna sin väg i den. Om inte kommer produkten att kännas övermäktig och svårpenetrerad för nya användare vilket, som vi sett i avsnittet om SaaS (se avsnitt 2.1), är ödesdigra egenskaper för en SaaS-produkt.

Konsekventa och tydliga namn och skyltar

Använd alltid samma benämning för samma funktion och kalla alltid varje sida för ett och samma namn. Verifiera dessutom alltid att användaren hamnat där den förväntat sig att hamna med tydliga skyltar.

Medan vi i ovanstående avsnitt lyfte fram vikten av konsekventa gränssnitt för att användare ska kunna klara av att hantera produkten fokuserar vi här på konsekventa namngivningar för att användaren ska kunna följa med i applikationens navigationsflöde. Vid flera moment i ”Barium Live!” händer det att man klickar på en länk med ett namn och kommer till en sida med ett annat namn. Klickar man t.ex. på ”Modellera” i ”Startguiden för platser” (se figur 3.2) kommer man till en sida med rubriken ”Processer” (se figur 3.4). Detta bidrog i stor utsträckning till informanternas förvirring kring var de befann sig i applikationen. Om vi återupptar Krugs (2006) liknelse mellan en applikation och en affär (se avsnitt 2.3.2) blir denna problematik uppenbar; om du hittar skylten ”Mejeriprodukter” och under den finner juice (vilket är fallet i vissa matbutiker) blir du lätt förvirrad. Likaså om du letar efter juice kommer det vara svårt att hitta den eftersom den står under skylten ”Mejeriprodukter”. För att kunna implementera ett self-servicetänk i produkten måste namngivningarna och uppmärkningen av varje sida vara tydlig och logisk och överensstämma med användarens förväntningar, annars kommer användaren aldrig hitta det den söker.

Bakåt ska alltid leda hela vägen hem

*Förse varje sida med en ”bakåt”-knapp som stegar tillbaka genom applikationens hierarki. Se till att ”bakåt”-knappen leder **hela** vägen till förstasidan.*

”Barium Live!” är försett med bredcrumbs som visar var i hierarkin man för tillfället befinner sig (se figur 3.6). I anslutning till dessa finns en ”bakåt”-knapp. Som vi berättade i resultatpresentationen ledde denna ”bakåt”-knapp bara ett steg bakåt i hierarkin – och sedan framåt igen. Detta skapade stor förvirring för informanterna och förstörde deras känsla för riktning i applikationen. Tidwell (2011) lyfter, som vi sett i avsnitt 2.3.2, fram vikten av bakåtknappar och att de leder användaren tillbaka till ”säker mark”. Rätt utförd fungerar ”bakåt”-knappar som en effektiv nödutgång för användarna. I det utförande de finns i ”Barium Live!” idag fungerar de närmast som en fälla som lurar användaren. Med self-serviceaspekten i åtanke är detta kritiskt då det försvårar användarnas möjligheter att förstå relationen mellan applikationens olika delar och hitta bland dem. Därför; ”Bakåt”-knappar *måste* leda tillbaka till förstasidan.

6.3 Hjälp

Ge en introduktion

För att användarna ska förstå domänspecifika detaljer och applikationens förutsättningar; förse applikationen med tydligt introduktionsmaterial som går igenom grunderna. Placera detta material så att det inte går att missa, om man inte verkligen vill.

Den typ av komplexa SaaS-applikationer som är föremål för den här studien kan förväntas att i samtliga fall figurera i ett sammanhang eller domän som har betingande inverkan på hur applikationen kan och bör utformas. I fallet med ”Barium Live!” är det t.ex. BPMN-standarder och processmodellerings best-practice som är styrande. Detta medför en uppsättning uttryck och konventioner som kan ha betydande vikt för förståelsen av produkten. Vill man uppnå en grad av self-service måste man säkerställa att dessa grundstenar är uppfattade och förstådda av

användarna. I "Barium Live!" måste man t.ex. för att kunna skapa en fungerande processapplikation av processmodellen börja sin modellering med att dra ut en "pool" och fylla den med "lanes" till vilka olika användarroller associeras. Detta var något som ingen av informanterna till fullo lyckades göra i rätt ordning, vilket visar på vikten av introduktion. Flera av de misstag som tog längst tid för informanterna att bemästra tror vi enkelt hade kunnat undvikas om de fått en sådan introduktion.

Schneiderman och Plaisant (2010) framhåller "tutorials" (se avsnitt 2.3.2) som ett effektivt sätt för användarna att introduceras till en ny applikation och menar att det svarar mot den generella önskan hos användarna att lära sig genom att göra istället för att läsa som såväl alla våra informanter (se avsnitt 5.3) som Krug (2006, se avsnitt 2.3.1) vittnar om. Mot denna bakgrund och det faktum att applikationen ska fungera enligt self-serviceprincipen rekommenderar vi "tutorials" som introduktionsmaterial för komplexa SaaS-self-serviceapplikationer.

Schneiderman och Plaisant (2010) lyfter också fram "Snabbstartsguiden" (se 2.3.2) som introduktionsmaterial med korta exempel och mycket grafiska presentationer av interaktionen med gränssnittet. Detta var något som också efterfrågades av flera av våra informanter, som en slags lathund att luta sig emot den första tiden av användandet. I ett sådant utförande ser vi snabbstartsguiden som ett bra komplement till "tutorials" med ett mer frekvent användande än en omfattande manual (vilken dock fortfarande fyller sin funktion, se *Tillbandabåll kompletterande, sökbar hjälp* nedan).

Placeringen av detta introduktionsmaterial (tutorials+snabbstartsguide) måste placeras i uppstarten av applikationen så att det blir den naturligt första punkten för interaktion (se *Skapa ett naturligt flöde*, avsnitt 6.1). För vana användare bör detta första steg gå att gömma genom att aktivt välja att göra så.

Tydlig och relevant hjälp när den behövs, där den behövs

Tillbandabåll hjälp och tillbandabåll den generöst - men - packa inte på användaren massor utav den vid ett och samma tillfälle, utan låt hjälpen komma till dem när och där den behövs.

Komplexa applikationer rimmar per definition dåligt med self-servicetänket och vi har sett hur bl.a. van Beuningen et al (2009, se avsnitt 2.2) vittnar om höga felfrekvenser i användandet av self-serviceprodukter. Men det går att råda bot på. Krug (2006) menar, som vi sett i kapitel 2, att komplexa webbsidor inte går att göra uppenbara, men att man bör sträva efter att göra dem självförklarande. Det samma gäller, menar vi, för komplexa SaaS-applikationer. För att göra något komplext självförklarande krävs (förutom god design i enlighet med principerna i avsnitt 6.1-6.2) att man ger användarna alla tips och råd som denne behöver för att lösa ett visst momentet. Som Krug (2006, se avsnitt 2.3.1) och samtliga av informanterna vittnat om är dock inte en massiv manual den bästa lösningen på problemet. Istället behövs kort och precis information som handlar om exakt det som sker på skärmen just för tillfället.

Ett exempel på hur detta skulle kunna användas i "Barium Live!" rör pilarna som relaterar objekt mellan varandra i processmodellen. Vi såg i avsnitt 5.1 att dessa vållade stora problem för informanterna, framförallt då de inte kunde hitta dem. Detta skulle kunna lösas

gränssnittsmässigt genom att lägga till ett pil-verktyg bland de andra processobjekten i paletten till vänster (se figur 3.6), vilket i sådana fall täcks in av principen *Följ konventioner* (se avsnitt 6.1), då konventionen för ritprogram är att verktygen finns i en sådan palett (se t.ex. Tidwell, 2011). Men eftersom informanterna, när de väl förstått hur pilarna ritas ut, tycker att det existerande sättet är mycket smidigt kan det vara bättre att använda sig av lokal hjälp. En lösning skulle då vara att det när användaren lagt ut två andra objekt på ritytan dyker fram en liten ruta som föreslår länkning mellan objekten som nästa steg och förklarar hur den länkningen går till.

Många situationer skulle kunna lösas på liknande sätt, med olika kombinationer av alla de exempel på gränssnittsintegrerad hjälp som vi lyfter upp i avsnitt 2.3.2, bl.a. förklarande och dynamiska tooltips och animationer som på dyker upp där/när de behövs och guidar användaren rätt.

Tillhandahåll kompletterande, sökbar hjälp

I tillägg till dynamisk hjälp i gränssnittet bör kompletterande hjälpdokumentation finnas tillgänglig. Låt användarna själva hitta den när de vill och gör den sökbar.

Komplexiteten i den aktuella typen av applikationer gör att all hjälp inte lämpar sig eller ryms i det format som nämndes i föregående princip. Vissa övergripande applikations- eller användningsområdesrelaterade regler måste kanske förklaras utförligare (vilket knyter an till designprincipen *”Ge en introduktion”* ovan). Detta visar på vikten av hjälp i flera nivåer, som vi också såg i resultatpresentationen att flera av informanterna efterlyste. Omfattande referensmanualer fyller sin funktion och har man resurser att skapa dem finns i vår mening ingen anledning att låta bli, men målet bör vara att göra produkten så självförklarande att de sällan behövs. Om användaren hamnar i ett läge där manualer av detta slag ändå behövs vittnar informanterna i resultatpresentationen om att användarens tålmod börjar ta slut, vilket också stöds av Krug (2006, se avsnitt 2.3.2). All generell hjälp-information måste därför finnas tydligt tillgänglig i gränssnittet, namngiven och utformad enligt våra principer om *Följ konventioner* och *Konsekventa och tydliga namngivningar* (avsnitt 6.1 resp. 6.2) så att den är lätt att hitta när den behövs. I tillägg till detta är det viktigt att den är sökbar så att användarna snabbt kan filtrera bort de delar som inte är relevanta för tillfället.

Skapa forum för en aktiv användarbas

Tillhandahåll enkla, tillgängliga och förståeliga forum för användare att själva dela kunskap mellan varandra.

Prahalad och Ramaswamy (2004) menade på att Wikis (se avsnitt 2.3.2) med användargenererat innehåll var en effektiv lösning för att skapa hjälpdokumentation med specialanpassat och tillfredsställande innehåll (se avsnitt 2.2). Ingen av våra informanter lyfte dock fram denna form som något de brukade använda sig utav (se avsnitt 5.3). Däremot uttryckte flera att de gärna vände sig till en kollega för att be om hjälp. Detta, menar vi, öppnar upp för möjligheten att Wikis och, kanske ännu hellre, communities (se 2.3.2) ändå kan vara ett relevant hjälpverktyg. Detta då de bygger på samma princip som att fråga en kollega, men frågan ställs i en onlinemiljö istället för användarens fysiska miljö. Detta förutsätter dock att användarbasen för applikationen

är tillräckligt stor och att man lyckas ta sig runt den problematik som van Beuningen et al (2009) identifierar med att få användare att bidra till informationskällorna (se avsnitt 2.2).

Förklara allt

Låt inget steg i navigeringen leda till helt tomma ytor och låt inget reglage i gränssnittet manipuleras utan effekt. Förklara vad som händer!

Vid ett par tillfällen under våra observationer hamnade informanterna i lägen där de inte förstod vad som skedde eller vad som förväntades av dem härnäst. T.ex. när de klickade på vissa flikar i processmodelleringen så möttes de av helt vita ytor där paletten tidigare låg. Detta skapade frustration och förvirring (se avsnitt 5.1). Likaså gav ändringar av inställningar i vissa reglage ingen uppenbar effekt (se avsnitt 5.1), vilket gjorde hela verktyget onödigt svårt för informanterna att känna sig bekväma med. Detta hade enkelt kunnat undvikas, t.ex. genom att i fallet med de tomma ytorna lägga till en förklarande text om vilket innehåll flikarna kan innehålla och var användaren vänder sig för att generera detta innehåll och i fallet med reglagen använda sig av tooltips (se 2.3.2) som förklarar vad de olika alternativen innebär eller pop up-rutor som förklarar den ändrade inställningen eller varför inget ändrades.

Schneiderman och Plaisant (2010) lyfter fram vikten av att system designas så att användaren vet vad denna ska göra härnäst och att detta är en förutsättning för att användarna ska kunna lära sig systemet på egenhand. Krug (2006) återkommer också till att designen måste vara självförklarande (se avsnitt 2.3.2). Huvudpoängen i denna princip bygger vidare på dessa tankar och vi menar att för att skapa en begriplig applikation som användarna kan ta till sig utan handledning krävs att det inte finns några lösa ändar. Användaren måste hållas i handen hela tiden och aldrig behöva fundera på ”vad var det som hände nu?”. Allting ska förklaras.



7. Slutsatser

Designprinciperna presenterade i föregående kapitel utgör vårt huvudsakliga resultat och svaret på vår frågeställning; *Vilka designprinciper är av särskild relevans vid design av komplexa SaaS-self-serviceapplikationer?*

Våra huvudsakliga upptäckter kan sammanfattas med att ”konsekvens” och ”tydlighet” är begrepp som återkommer i de flesta av principerna och äger stor tyngd för att self-servicetänket ska kunna implementeras i komplexa SaaS-applikationer. Med konsekvens avses såväl konsekvens inom applikationen som konsekvens mot de konventioner och begrepp som fötts i tidigare applikationer inom den domän där applikationen ska verka. Tydlighet är viktigt dels ur navigeringssynpunkt där användaren aldrig får hamna i ett läge där den inte vet hur den ska ta sig vidare och/eller tillbaka men också för att den aldrig ska behöva fundera över vad applikationens olika delar och funktioner är till för. Då de applikationer vi diskuterar i denna uppsats är av komplex natur krävs i vissa situationer ändå hjälp för att användaren ska kunna klara särskilda moment. Hjälpen ska då utformas så att den i så stor utsträckning som möjligt dynamiskt visas där och när den behövs. I tillägg till detta behövs väl utformat introduktionsmaterial som reder ut initiala frågetecken samt utförliga referensmanualer att falla tillbaka på vid behov. Lyckas man dessutom skapa forum för användarna att aktivt dela kunskap mellan varandra har man skapat goda förutsättningar för att möta användarnas varierande behov av hjälp.

Övriga insikter är att existerande designforskning täcker in stora delar av våra upptäckter och att många av de generella designprinciper som finns däri är applicerbara även på komplexa SaaS-self-serviceapplikationer och innefattade i flera av våra principer. Det visade sig också att de råd för design av traditionella enkla self-serviceprodukter som finns, vilka i huvudsak omfattar konsekvens och tydlighet, även gäller för komplexa self-serviceprodukter, med tillägget att hjälp måste finnas tillgänglig i större utsträckning och i fler nivåer.

7.1 Studiens relevans och överförbarhet

Denna undersökning och rapport är genomförd och skriven på uppdrag av Barium AB med det primära syftet att reda ut hur designen av deras produkt ”Barium Live!” bör förändras för att fungera enligt en self-serviceprincip. Vår fokusering har legat på att ta fram generella designprinciper som är applicerbara inte bara på ”Barium Live!” utan på alla produkter som delar den komplexitet och leveransmodell som ”Barium Live!” representerar. Därmed är resultatet av studien - vårt designramverk - överförbart även till andra organisationer som vill skapa komplexa SaaS-applikationer för self-serviceanvändande. Detta är relevant då, som vi sett i inledningskapitlet och vilket stöds av omfattande forskning, affärsmodellen kraftigt växer och ställer krav på leverantörerna att leverera attraktiva produkter som snabbt ger värde till konsumerande organisation.

7.2 Förslag till fortsatt forskning

Denna undersökning är endast baserad på empiriska upptäckter från användandet av en enskild applikation ("Barium Live!"). Det ramverk vi tagit fram bör därmed testas och utvärderas på ytterligare produkter för att verifiera designtankarna som byggt det. Vi gör själva inte någon utvärdering av ramverket då vårt arbete med att ta fram det riskerar, i enlighet med vad Patel och Davidssons (2011) skriver, att göra oss blinda för dess brister. Därför uppmanar vi kommande studenter att utvärdera och bygga vidare på vårt ramverk i sina uppsatser. En intressant aspekt vore att göra samma undersökning, men med en produkt som författarna har inflytande över att förändra. På så vis skulle observationerna kunna ske i iterationer där upptäckter i en iteration korrigeras i applikationen och möjliggör ytterligare/djupare upptäckter i efterföljande iteration, vilka sedan kan omsättas till nya designprinciper som kompletterar det vi kommit fram till i denna studie.

Ytterligare ett intressant forskningsområde, stort nog för en egen uppsats, är hur man kan övervinna svårigheterna med att få användare att dela med sig av sin kunskap och skapa den aktiva användarbas som efterfrågas bland annat i designprincipen "*Skapa forum för en aktiv användarbas*".



Källor

- Anselmsson, J. (2001) *Customer-perceived Service Quality and Technology-based Self-service*, dss., Lund Business Press, Lund, Sverige.
- Avery, P. (2009) *Four Strategies for Kiosk Application Design*. [Elektronisk] NetWorld Alliance LLC. Tillgänglig: http://global.networldalliance.com/downloads/white_papers/Olea_WP_Application%20Design_To%20Launch.pdf, [2012-04-25]
- Bikers, J; Slawsky, R (2010) *Intelligent Kiosk Design*. [Elektronisk] NetWorld Alliance LLC, Tillgänglig: http://global.networldalliance.com/downloads/white_papers/KMC_G_Intelligent-Kiosk-Design_To-Launch.pdf [2012-05-11]
- Dabholkar, P.A. (1994) Technology-based Service Delivery: A Classification Scheme for Developing Marketing Strategies, i Swartz; T.A.; Bowen, D.E.; Brown, S.W. (Red.), *Advances in Services Marketing and Management*, vol.3, ss. 241-271.
- Dubey, A.; Wagle, D. (2007) Delivering software as a service, *The McKinsey Quarterly*, Web exclusive, maj 2007. [Elektronisk] Tillgänglig: <http://www.mckinseyquarterly.com> [2012-04-02]
- Hoogvliet, M.T. (2008) *SaaS Interface Design - Designing web-based software for business purposes*, [Elektronisk] Tillgänglig: http://www.catalystresources.com/pdf/whitepaper_maartenhoogvliet_saasinterfacedesign.pdf [2012-03-14]
- Klein, G. (1998) *Sources of Power: How People Make Decisions*, The MIT Press, Cambridge, USA.
- Koblanck, H. (2003) *Typografi, bild och grafisk design*, Sanoma Utbildning, Stockholm, Sverige.
- Krug, S. (2006) *Don't make me think - A Common Sense Approach to Web Usability* (2:a uppl.), New Riders Publishing, Berkeley, Kalifornien, USA.
- Ladik, D.M. (1999) A Technology-based Self-service Delivery System: An Exploration Adoption Model in a Supermarkt Delicatessen, *In Proceedings of Annual Meeting of the Society Marketing Advance*, Atlanta GA, USA.
- Nielsen, J. (2000) *Why you only need to test with 5 users*. [Elektronisk] Tillgänglig: <http://www.useit.com/alertbox/20000319.html> [2012-05-10]
- Papazoglou, M.P.; Ribbers, P.M.A. (2006) *e-Business - Organizational and Technical Foundations*, John Wiley & Sons, Ltd, Chichester, West Sussex, England.
- Patel, R.; Davidson, B. (2011) *Forskningsmetodikens grunder – Att planera, genomföra och rapportera n undersökning*, Studentlitteratur, Lund, Sverige.
- Prahalad, C. K.; Ramaswamy, V. (2004) Co-creation Experiences: The Next Practice in Value Creation, *Journal of Interactive Marketing*, 18 (Summer), 5-14.

- Silver, B. (2011) *BPMN Method and Style with BPMN Implementer's Guide - A Structured Approach for Business Process Modeling and Implementation Using BPMN 2.0* (2:a uppl.), Cody-Cassidy Press
- Simon, H. (1957) *Models of Man: Social and Rational*, Wiley, Oxford, England.
- Sharp, H.; Rogers, Y.; Preece, J. (2011) *Interaction Design - Beyond Human-Computer Interaction* (3:e uppl.), John Wiley & Sons, Ltd, Chichester, West Sussex, England.
- Shneiderman, B.; Plaisant, C. (2009) *Designing the User Interface - Strategies for Effective Human-Computer Interaction* (5:e uppl), Pearson Education, New Jersey, USA.
- Tidwell, J. (2011) *Designing Interfaces - Patterns for Effective Interaction Design* (2:a uppl.), O'Reilly Media, Inc., Sebastopol, Karlifornien, USA.
- Vella, J. (2011) *Determinants in the Adoption of a Customer Relationship Management System: A Study among Bank Managers*, diss., Luleå Tekniska Universitet, Sverige.
- Verma, G. (2011) Software as a Service, *Pragyaan: Journal of Informaion Technology*, vol. 9(1), ss.32-37.
- van Beuningen, J.; de Ruyter, K.; Wetzels, M.; Streukens, S. (2009) Customer Self-Efficacy in Technology-Based Self-Service: Assessing Between- and Within-Person Differences, *Journal of Service Research*, vol. 11(4), ss. 407-428.
- Vignelli, M. (2008) *The Vignelli Canon* [Elektronisk] Tillgänglig: <http://www.vignelli.com/canon.pdf> [2012-04-25]
- Wang, J.; Namen, J. (2004) *Customer Adoption of Technology-Based Self-Service – A Case Study on Airport Self Check-in Service*, D-uppsats, Luleå Tekniska Universitet.
- Wong, J.Y.M.; Chan, C.C.H.; Siu, K.W.M. (2010). *User Centered Design for Self-Service Point in MTR*. Conference paper delivered at The 12th International Conference on Mobility and Transport for Elderly and Disabled Persons (TRANSED) held in Hong Kong on 2-4 June, 2010.

Bilaga 1 – Användbarhetstestmanus

Anpassat från Steve Krug, <http://www.sensible.com/downloads-dmmt.html> [Hämtat: 2012-04-02].

- Ha webbläsaren öppen med Google eller annan “neutral” sida framme.**

Hej, _____. Jag heter _____ och jag kommer att leda dig igenom den här sessionen och det här är _____ som kommer att ta lite anteckningar.

Innan vi börjar så har jag lite information som jag kommer att läsa för att vara säker på att vi får med allting.

Du vet säkert redan ungefär varför vi är här, men jag ska gå igenom det lite snabbt ändå. Vi skriver en uppsats om design av komplexa system och som en del av det ber vi folk att försöka använda ett processmodelleringsverktyg som heter Barium Live! som ert företag använder sig av för att försöka se hur det kan förbättras för att bli enklare att använda. Idag är det inte gjort för att kunna användas utan utbildning, men vi har anpassat scenarierna som vi ska gå igenom efter det. Det hela kommer att ta ungefär en timme och sen kommer vi att ha en intervju med dig för att fördjupa oss i upplevelsorna från testsessionen, vilket kommer att ta ca en halvtimme.

Vi vill poängtera att det är produkten vi testar och inte dig. Du kan inte göra något fel, faktiskt så är det här kanske ett av få tillfällen där man inte behöver oroa sig för att göra några misstag över huvudtaget!

När du använder produkter så vill jag be dig att i så stor utsträckning som möjligt försöka tänka högt; säg vad du tittar på, vad du försöker göra och vad du tänker. Det kommer att vara till stor hjälp för oss.

Och kom ihåg att du inte behöver oroa dig för att såra våra känslor – vi har egentligen ingenting med den här produkten att göra och allt vi gör här idag är för att hitta områden där den kan förbättras, så vi är i stort behov av dina ärliga åsikter!

Om du har några frågor medan vi håller på är det bara att ställa dem. Jag kanske inte kommer att kunna svara på dem direkt, eftersom vi är intresserade av att se hur människor hanterar produkten när ingen sitter bredvid dem. Men om du fortfarande har frågor när vi är klara ska jag försöka besvara dem då. Om du behöver ta en paus vid något tillfälle är det bara att säga till.

Med ditt medgivande kommer vi att spela in skärmbilder och ljud. Inspelningen gör att vi inte behöver ta så mycket anteckningar och kommer bara att användas som underlag för slutsatser i vår uppsats samt eventuellt av Bariums utvecklingsavdelning för att förbättra produkten. Ingen utomstående kommer att

se eller höra inspelningen och du kommer att vara helt anonym i allt vi skriver om sessionen.

Om det är okej så kan du bara läsa och skriva under det här pappret som säger precis det jag just sa.

- Ge användaren inspelningsmedgivandet och en penna**
- Medan de skriver, STARTA skärminspelningen.**

Har du några frågor än så länge?

OK. Innan vi tittar på produkten ska jag bara ställa några snabba frågor;

Vad är ditt yrke? Vad gör du en vanlig arbetsdag?

Om du skulle uppskatta, grovt, hur många timmar skulle du säga att du spenderar vid datorn på en vecka eller dag?

Vad gör du till största delen vid datorn? Internet, e-mail, spel etc.

Har du sett eller använt Barium Live! tidigare?

OK, bra. Vi är klara med frågorna och kan börja titta på produkten.

- Be användaren skriva in adressen www.bariumlive.com**

Till att börja med ska jag be dig att bara titta på sidan och berätta vad du tänker om den; Vad slår dig? Vad kan man göra? Kolla bara runt lite och berätta vad än som kommer upp i huvudet.

Om du vill så kan du scrolla upp och ner, men klicka inte på något än.

- Låt detta pågå ett par minuter, max 3-4.**

Tack. Nu kommer jag att be dig utföra några specifika uppgifter. Jag läser var och en högt och du får en utskriven kopia.

Och återigen – Det är oss till stor hjälp om du försöker att tänka högt så mycket som möjligt medan du utför uppgifterna!

- Ge användaren det första scenariot och läs det högt.**
- Låt användaren hålla på tills du känner att det inte ger något mer av värde eller tills användaren blir väldigt frustrerad.**
- Upprepa för varje uppgift eller tills tiden tar slut.**

Tack! Det här har varit till stor hjälp för oss.

Har du några ytterligare frågor för användandet [andra undersökaren]?

- Ställ eventuella ytterligare frågor och undersök så djupt som behövs**

Har du några frågor nu när vi är klara med observationen?

- Stoppa skärminspelningen och spara filen!**
- Ta rast innan uppföljningsintervju.**

Bilaga 2 - Användbarhetstestuppgifter

Logga in

Användarnamn: ██████████

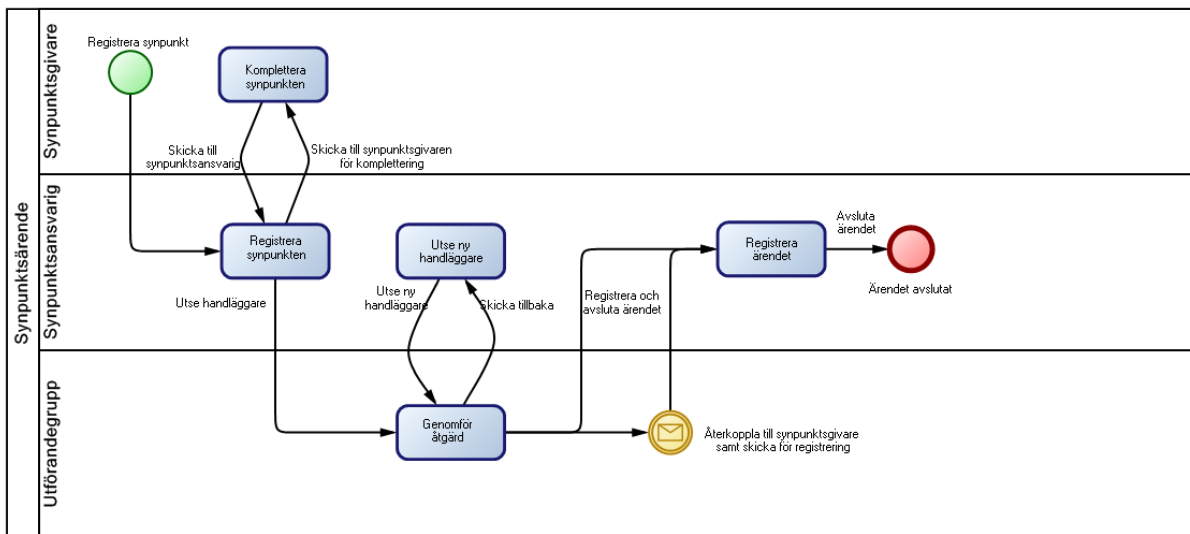
Lösenord: ██████████

(Ditt namn i systemet är Kalle Karlsson)

Uppgift 1 - Modellera

Din första uppgift är att modellera en process. Processen i fråga handlar om att lämna och hantera synpunkter i din organisation. Den ska se ut som bilden nedan.

Kom ihåg att hela tiden "tänka högt" och berätta vad du gör och varför!



Uppgift 2 - Arbeta i processen

Efter att man har modellerat en process kan man göra den till en fungerande applikation, ett "program", där man kan utföra de olika uppgifterna som man modellerade. Detta sker i huvudsak genom att man kopplar olika formulär, knappar och fält till varje steg i processen.

Vi har förberett en sådan applikation baserad på den process du just modellerat. Nu ska du testa att arbeta i den applikationen. För att du ska få känsla för flödet i processen så kommer du i den här uppgiften att ha alla roller, dvs. vara Synpunktsgivare, Synpunktsansvarig och Utförandegrupp. Det innebär att när du utfört den eller de uppgifter som du ska göra i en roll, som t.ex. Synpunktsgivare, så kommer ärendet att komma tillbaka till dig, men du hanterar det nu i en annan roll.

Tänk på att hela tiden "tänka högt" och berätta vad du gör och varför!

- Lämna en synpunkt (=starta en ny processinstans)
- Hantera synpunkten i alla steg/roller tills ärendet är avslutat.

Uppgift 3 – Listor

- Gå till “Mina aktivitetslistor”.
- Titta på sidan och berätta vad du tänker om den;

Vad är det första du kommer att tänka på?

Vad kan man göra här?

Titta bara runt lite och berätta om de tankar som kommer upp!

Tänk på att hela tiden ”tänka högt” och berätta vad du gör och varför!

Du ska nu göra följande uppgifter:

1. Skapa lista

- Skapa en egen lista som du namnger med “Mina arbetsuppgifter” och ger valfri beskrivning.
 - Listan ska sortera ut dina arbetsuppgifter från applikationen “Synpunktprocessen” och från processen “Synpunktsprocessen”.
 - Listan ska bara innehålla arbetsuppgifter från de senaste 7 dagarna och bara vara aktiva arbetsuppgifter.
 - Listan ska bara visa kolumnerna “Namn”, “Instans” och “Applikation”.
 - Listan ska delas ut till användaren “Christian Renlund” som ska kunna redigera listan.
 - Spara din nya lista
 - Öppna din lista och beskriv hur du upplever den? Hur fungerar den?

2. Redigera listan

- Du ska nu redigera din lista så att den också visar kolumnen “Initierad av” och så att hela listan grupperas efter kolumnen “Namn”.
- En ny kolumn ska läggas till i din lista som hämtas från formuläret och heter “Behövs åtgärd?”.
- Din lista ska också markeras som “Standard”, så att den alltid visas först när man öppnar fönstret för “Mina aktivitetslistor”.

3. Ta bort din skapade lista

Bilaga 3 – Inspelningsmedgivande

Inspelningsmedgivande

Tack för att du deltar i vår användbarhetsstudie.

Vi kommer att spela in ljud och skärmbilder från den här sessionen för att kunna gå tillbaka och studera olika moment vid senare tillfälle i vårt uppsatsarbete samt ge anställda vid Barium AB möjlighet att ta del vad som skett här idag.

I uppsatsen kommer du att behandlas anonymt.

Var vänlig och läs nedanstående kommentar och skriv under om du samtycker.

Jag förstår att min testsession kommer att spelas in.

Jag tillåter Daniel Alteborg och Christian Renlund att använda inspelningen som underlag för sitt uppsatsarbete vid Göteborgs universitet, vårterminen 2012 samt Barium AB att använda inspelningen internt i syfte att förbättra designen av produkten ”Barium Live!”.

Signatur: _____

Namnförtydligande: _____

Datum: _____

Bilaga 4 - Intervjufrågor

Tema 1: Upplevelse från observationen

- Tankar om produkten som helhet
- Ta fram registrerade händelser från observationen*
- Vad var det som gjorde momentet svårt?
 - Vad skulle du vilja haft för hjälp?
 - Var hade du förväntat dig att hitta funktionerna (som du inte hittade)?

Tema 2: Navigering och Layout

- Hur upplever du det sättet man tar sig mellan produktens delar?
- Hur upplever du placering av verktyg och menyer i de olika delarna? Logiska? Lätt att hitta? Naturliga?
- Tycker du att namn på olika delar och funktioner stämmer överens med det du letade efter/förväntade dig?

Tema 3: Allmänt om hjälp

Om informanten inte använt hjälpfunktionerna:

Säg du hjälpfunktionerna? I så fall varför inte använda dem? Och hur skulle hjälpfunktionerna vara för att de skulle använda dem?

Om informanten använd hjälpfunktionerna:

Vad tyckte du om innehållet?

Placeringen?

Mer generellt om hjälpfunktioner:

- Har du använt och i så fall hur har du upplevt
 - Wikis
 - Manualer
 - animeringar (som pilar etc. som visar var man ska göra saker)
 - instruktionsfilmer
 - Diskussionsforum
- När föredrar du hjälp?
 - Innan du börjar använda produkten?
 - I början?
 - Allteftersom du håller på?
 - När det blir fel?
 - När man kommer tillbaka nästa gång (start-up tips)
- Vilken typ av hjälp föredrar du i dessa situationer?