

Flödesapplikation

- företagsbaserad utbildningsmodell på Internet



LUNDS
UNIVERSITET

Lunds Tekniska Högskola

LTH Ingenjörshögskolan vid Campus Helsingborg
Multimediateknik

Examensarbete:
Fredrik Jacobsson
Michael Persson

© Copyright Fredrik Jacobsson, Michael Persson

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds Universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Media-Tryck
Biblioteksdirektionen
Lunds Universitet
Lund 2007

Sammanfattning

Intern företagsutbildning via e-learning är ett aktuellt ämne, framförallt eftersom företagen ser det som en chans att bli effektivare och spara pengar. Av de företagsanpassade utbildningsapplikationer som finns på marknaden idag, är nästan samtliga skapta för att förklara för användarna (kursdeltagarna) hur saker och ting ska fungera. Deras stora svaghet ligger dock i att förklara och förmedla *varför* dessa saker fungerar, varpå man riskerar att misslyckas med att förmedla en djupare förståelse och skapa en bättre kunskapsbas. Därför finns det troligen många anställda på företag som har fått lära sig hur allt runt om kring dem fungerar, men inte varför det nu fungerar som det gör.

Denna examensrapport redogör för vårt arbete med att ta fram en webbaserad applikationsmodell, utvecklad i Adobe Flash, tilltänkt för utbildning, och som på ett enkelt men effektivt sätt synliggör hur olika typer av företagsflöden fungerar och, framförallt, varför de fungerar på detta sätt.

Vår iterativa arbetsmetod startade med en analys av företaget Informator och dess verksamhet. Därefter fortsatte vi med att studera flöden hos Informator och den tilltänkta målgruppen för applikationsmodellen. Sammantaget fick vi tillräckligt med underlag för det arbete som nu lett fram till en modell som kan synliggöra både administrativa, tekniska och vardagliga flöden.

Applikationsmodellen har anpassats dels efter Informators syfte och mål, dels efter den tilltänkta målgruppen (verksamhetschefer och projektledare) och dennes prioriteringar. Informationsinnehållet har sedan anpassats efter de yttre förutsättningarna, såsom tid och plats, för användandet.

Kärnan i arbetet har varit att ta fram ett regelverk, bestående av en uppsättning komponenter och definitioner för dessa, för att kunna synliggöra olika flöden med en gemensam struktur. Flertalet lösningsförslag för en sådan struktur har tagits fram och vägts emot varandra, innan vi kunnat göra vårt val – ”Halvlinjär struktur med vägval och processblock”. Denna lösning har visat sig vara mycket användbar och gjort det möjligt att presentera flöden på ett enhetligt sätt i applikationsmodellen.

För att säkerställa arbetets kvalitet har omfattande testningar genomförts under hela arbetsprocessen. Vad dessa testningar visat är att mycket av det vi gjort är bra – allt från flödesstrukturen till informationsinnehållet. Själva upplägget för presentationen kunde dock ha gjorts på ett annorlunda sätt. Användarna hade gärna önskat ett enklare upplägg för att man ska till fullo ska kunna ta till sig de abstrakta och ibland svårförståeliga saker som man presenteras för.

Nyckelord: *multimedia, IT, utbildning, e-learning, applikationsmodell, Flash, Internet, företagsprocess.*

Abstract

Internal education and e-learning is a subject of current interest for most companies, particularly since they are seeing the opportunity to become more efficient and strengthen their economy. The main problem is that almost all business related applications are designed with the purpose to explain how cases work. Explaining why these cases works as they do, are their weakness, and the risk of users failing to get a deeper knowledge is considerable. Therefore, it's likely to find employees at companies all over the world, which have been told how to do things, but lacks the understanding of why they're supposed to do just like that.

This essay deals with our work of creating a concept for e-learning, developed in Adobe Flash for online purposes, which in a plainly but effectively manner can show and explain the function of various types of company processes.

Our work started out with an analysis of our case company, Informator, and their activities. We then continued studying the work processes at Informator and the target group for the application we were going to create. Together they created the foundation of the work that has led to an application model, able to handle administrative, technical and everyday processes.

The application model has been brought into line with Informator's goal, as well as the target group (project managers) and their priorities. The information presented in the application has also been suited according to conditions such as term and place for application usage.

The very essence of the work is the core process structure and its solution of presenting different processes in a similar way in the application model. Several ideas and proposals for such a structure has been discussed, among with for- and drawbacks for each of those. This information has been taken into consideration before the structure, that now builds the strong foundation of our work, could be selected.

Several function tests have been executed throughout our project, to assure the quality of our work. The result of these tests shows the strength in our work; from the core process structure to the information presented in the application model. Other things, mainly the set-up of the presentation itself, could have been made in a different way to fully satisfy the users needs, so that they could learn the things they're presented to, in a more efficient and understandable way.

Keywords: multimedia, IT, education, e-learning, application model, Flash, Internet, company process.

**”Sätt aldrig likhetstecken mellan information
och kunskap”**

Anonym

Förord

Det är med stor glädje och lättnad som vi med dessa förord sätter punkt för vårt examensarbete, ett arbete som omfattat 15 högskolepoäng och ungefär sex månaders heltidsstudier. Arbetet har varit kul och lärorikt, om än stundtals mycket frustrerande då vi känt oss hopplöst efter i vår tidsplanering. Resan fram till denna rapport har resulterat i framförallt en insikt; att engagemang är den viktigaste drivkraften för god inläring.

Vi vill tacka Henrik Thorsell och de anställda på Informator i Malmö för att vi fått följa det dagliga arbetet, och fått möjligheten att utföra vårt projekt i en skarp miljö. Vi uppskattar att det upplåtits arbetsutrymme åt oss trots att arbetet drog ut på tiden.

Ett stort tack riktas till vår handledare Lise Jensen samt till våra vänner Louise Baaz och Andreas Feuk, som alla varit till stor hjälp med korrekturläsning och som gett oss värdefulla synpunkter på vårt arbete med rapporten.

Helsingborg, november 2007

Fredrik Jacobsson
Michael Persson

Innehållsförteckning

1 Inledning	1
1.1 Problemformulering	1
1.1.1 Pedagogiska utmaningar	1
1.2 Syfte och effektmål.....	2
1.3 Frågeställningar	3
1.4 Avgränsningar.....	4
1.5 Disposition.....	4
2 Metod.....	6
2.1 Metodstegen.....	6
3 Förstudie	10
3.1 Om Informator	10
3.2 Förutsättningar för arbetet.....	11
3.2.1 Informators syfte och mål med applikationsmodellen	11
3.3 Flöden och flödesapplikation – vad menas?	12
3.4 Framgångsrecept för lyckad e-learning.....	13
3.5 Målgruppen för applikationsmodellen - definition och analys	14
3.5.1 Undersökningsresultaten.....	16
3.5.2 Andra potentiella målgrupper	18
3.5.3 Typiska användare ur målgruppen.....	18
3.5.4 Målgruppen och dennes inflytande på applikationsmodellen	19
3.6 Företagsflöden hos Informator	21
3.7 Kravspecifikation	25
3.7.1 Projektkrav	25
3.7.2 Systemkrav.....	25
3.7.2.1 <i>Flödesstrukturen</i>	25
3.7.2.2 <i>Applikationsmodellen</i>	25
3.7.3 Kvalitetskrav	26
3.8 Testspecifikation.....	27
3.8.1 Teststruktur	27
3.8.2 Regressionstest.....	28
3.8.3 Löpande test.....	28
3.8.4 Sluttest.....	29
4 Flödesstrukturen	30
4.1 Val av representativa flöden att basera strukturen på	30
4.2 Principiellt olika lösningsförslag för flödesstrukturen.....	34
4.2.1 Utvärdering och val av lösningsförslag	39
4.3 Regler för hur strukturen ska se ut och fungera	41
4.3.1 Allmänna regler	41
4.3.2 Komponentregler	41
4.4 Genomgång av lösningen för flödesstrukturen.....	44

4.4.1 Utvärdering av genomgången	44
4.5 Förändringar och förbättringar utifrån genomgången.....	45
4.6 Slutgiltig flödesstruktur	47
5 Applikationsmodellen	50
5.1 Val av flöden att presentera i applikationsmodellen	50
5.2 Analys av befintliga, närbesläktade, utbildningsapplikationer	50
5.3 Olika tänkbara lösningsförslag för flödespresentationen	51
5.3.1 Utvärdering och val av lösningsförslag för flödespresentation	54
5.4 Riktlinjer för applikationsmodellen	55
5.4.1 Allmänna riktlinjer	56
5.4.2 Riktlinjer för visualiseringsmomentet.....	56
5.4.3 Riktlinjer för förståelsemomentet	57
5.5 Storyboard och detaljskisser	57
5.5.1 Walkthrough.....	61
5.6 Applikationsmodell 0.1 – en första version.....	61
5.6.1 Applikationens uppbyggnad.....	62
5.6.2 Gränssnittet – färg och form.....	64
5.6.2.1 <i>Det multimediala innehållet</i>	64
5.7 Informationspresentation och pedagogik	65
5.8 Metafor för flödespresentationen	65
5.9 Löpande test och regressionstest.....	68
5.9.1 Testutvärdering och testernas inflytande på arbetet.....	69
5.10 Applikationsmodell 1.0	71
5.11 Sluttestning.....	72
5.11.1 Testscenarion.....	73
5.11.2 Resultat, utvärderingar och förändringar	75
5.11.2.1 <i>Systemtest</i>	75
5.11.2.2 <i>Scenariotest</i>	76
6 Resultat.....	79
7 Slutsatser och diskussion	81
7.1 Vidareutveckling och rekommendationer	83
7.1.1 Administrationsgränssnitt.....	84
7.1.2 Andra användningsområden.....	85
8 Referenslista.....	86
8.1 Tryckta källor	86
8.2 Elektroniska källor	87
Bilaga 1 – Enkät till anställda vid Informator.....	88
Bilaga 2 – Enkätsammanställning	91
Bilaga 3 – Representativa flöden.....	94
Bilaga 4 – Presentationsmaterial för flödesstrukturen.....	99

Bilaga 5 – Storyboard	100
Bilaga 6 – Detaljskisser	104
Bilaga 7 – Resultat av scenariotestning.....	106

1 Inledning

Informator Utbildning Svenska AB¹ är ett utbildningsföretag som säljer kundanpassade projektledningskurser och IT-utbildningar, främst till andra företag.

Enligt vår handledare, affärsutvecklingschef Henrik Thorsell, har flera av Informators kunder idag ett behov av att internutbilda sin personal, så att de får en förståelse för hur olika flöden inom det egna företaget ser ut och fungerar. Därför vill nu Informator ta fram ett utbildningskoncept i form av en applikationsmodell som kan illustrera detta.

1.1 Problemformulering

Problematiken ligger i att skapa en applikationsmodell som inte enbart visar hur olika flöden inom ett företag fungerar, utan framförallt *varför* de fungerar på ett visst sätt. Med hjälp av en sådan applikationsmodell ska användaren sedan själv utforska olika flöden, i syfte att ge en ökad förståelse för hur saker och ting hänger ihop inom företaget.

Konkret innebär detta att fyra huvudsakliga delproblem föreligger:

- Att undersöka några typfall av flöden inom ett företag och utifrån dessa välja ut de som kan anses representativa för grupper av likartade flöden.
- Att skapa en flödesstruktur som kan ligga till grund för en enhetlig flödespresentation i applikationsmodellen.
- Att skapa en applikationsmodell med Internet som distributionskanal.
- Att studera vad i applikationsmodellen som kan användas på generell basis för andra företag än Informator, d.v.s. visa om det är möjligt att, utifrån ett specifikt företagsfall, skapa en generell applikationsmodell.

1.1.1 Pedagogiska utmaningar

Ur ett mer pedagogiskt perspektiv finns det en rad andra problem som måste lösas vid skapandet av en e-learningapplikation, såsom:

¹ Benämns i kommande text som enbart Informator.

- Vilken nivå ska man lägga sig på när det gäller applikationsmodellens informationsinnehåll och utformning för att nå målgruppen, samtidigt som man uppfyller företagets krav och förväntningar?
- Hur ska man kunna förmedla denna information på ett tydligt och effektivt sätt utifrån målgruppens och arbetsplatsens förutsättningar?
- Hur ska man kunna motivera användaren att vilja ta till sig informationen som förmedlas?

Just att nå användarna och få dem att vilja ta till sig det förmedlade budskapet är inte helt enkelt. Som Karin Mårdsjö och Pär Carlshamre tar upp i boken *Retorik kring tekniken* [9] så finns det nämligen ett problem just när det gäller anvisningar, handböcker och bruksanvisningar (m.a.o. starkt knutet till vår typ av applikationsmodell). Det finns risk för att en konfliktsituation uppstår mellan kraven att uppmuntra till användning, inläring och läsning, och samtidigt ge klara och precisa instruktioner. Det hela kan i värsta fall resultera i att man inte skapar kontakt med sin användare, utan snarare det omvända – distans.

1.2 Syfte och effektmål

Vårt syfte med projektet är att ta fram ett målgruppsanpassat utbildningskoncept för e-learning², i form av en applikationsmodell. Denna modell ska på ett enkelt men effektivt sätt synliggöra för användaren hur olika flöden i företaget fungerar och, framförallt, varför de fungerar på just detta sätt.

Vi har inte som avsikt att, som många andra, förbättra och effektivisera ett flöde. Vår enda uppgift är att visa varför det befintliga fungerar som det gör. Ett effektmål av applikationsmodellen kan vara att, om man förstår exakt varför det fungerar som det gör, skapa bättre underlag till att hyra in någon som kan förbättra och effektivisera. Man kan tydligare se sådana problemområden och knutar som kan förbättras eller ändras för att göra verksamheten mer förståelig och möjligtvis också mer lönsam.

² För vårt arbete; ett samlingsnamn på företagsbaserade kurser på Internet.

1.3 Frågeställningar

Med utgångspunkt från problemformuleringen kan ett antal större och mindre frågor formuleras:

- *Vad är det som ska läras ut?*
Vad menas med flöden och en flödesapplikation? För vilka typer av företagsflöden är applikationsmodellen tilltänkt, d.v.s. vad är det som företaget vill lära ut genom applikationsmodellen?
- *Vilken är målgruppen och vad har den för behov och prioriteringar?*
Vilken målgrupp är det som ska använda applikationen? Vilken är den minsta gemensamma nämnaren för målgruppen när det gäller datorvana, arbetssituation, erfarenheter av e-learning etc? Vilka andra målgrupper skulle kunna tänkas ha användning för applikationen?
- *Vilka flöden kan användas?*
Vilka flöden finns inom Informator, och vilka av dessa flöden är relevanta för vårt arbete? Hur ser flödena ut och vad har de gemensamt? Finns det flöden som kan anses fungera som representanter för andra, likartade flöden, och som vi kan skapa en flödesstruktur utifrån?
- *Hur kan man skapa en struktur utifrån de valda flödena?*
Vilka likheter och skillnader finns mellan de utvalda representativa flödena? Vilka olika komponenter och egenskaper har de olika flödena? Går det att finna en gemensam struktur dem emellan?
- *Hur ska flödena presenteras på ett lämpligt sätt?*
Hur ska de utvalda flödena presenteras i applikationsmodellen, så att användarna förstår syftet och varför saker och ting förhåller sig som de gör? Hur ska det grafiska gränssnittet för flödespresentationen se ut och fungera för att passa målgruppen och dennes behov och prioriteringar?
- *Vilket innehåll ska förmedlas, och hur ska detta förmedlas?*
Vilken konkret information i de olika flödena är det som ska förmedlas till målgruppen? Ska information presenteras med text och bild, eller bara text? Ska ljud och video användas?
- *Vilken detaljeringsnivå gäller för flödena och dess innehåll?*
Hur pass detaljerat ska ett flöde och dess innehåll presenteras – översiktligt eller grundligt?

1.4 Avgränsningar

Informator är endast intresserat av en webblösning. Vi avgränsar oss därför till att inte göra någon undersökning som kan visa vilken som egentligen är bästa distributionskanalen för applikationsmodellen.

Vi genomför inte heller någon undersökning som visar vilken utvecklingsplattform som är mest lämpad för applikationsmodellen. Informators företagsstandard när det gäller utveckling av nya webbapplikationer är Adobe Flash, därmed gäller detta även för oss.

Informator tilldelar oss en målgrupp för applikationsmodellen, vi undersöker alltså inte om denna målgrupp är den mest lämpliga eller ej.

Endast ett begränsat antal (max fem stycken) flöden presenteras i applikationsmodellen. Dessa flöden ska vara av olika typer för att kunna påvisa applikationsmodellens dynamik.

Vi jobbar mot målet att skapa en fullt fungerande och presentabel applikationsmodell som visar vad vi kommit fram till under arbetet. Fokus ligger inte på att skapa en kodmässigt eller grafiskt snygg lösning.

Vi avgränsar oss också till att inte skapa någon form av administratörsfunktion som gör att man t.ex. kan lägga till, uppdatera eller ta bort flöden. Detta anser vi skulle kunna vara ett separat examensarbete, som vi därför väljer att inte gå djupare in på. En kortare diskussion hur en sådan administratörsfunktion skulle kunna fungera tas emellertid upp i kapitel 7.1.1 – Administrationsgränssnitt.

1.5 Disposition

Rapporten inleds med en kort redogörelse för projektets bakgrund och utformning, samt hur vi tänkt oss att lösa eventuella problem. Därefter följer en ingående redogörelse av projektets genomförande och våra tillvägagångssätt. Avslutningsvis presenterar vi våra resultat och de slutsatser vi kommit fram till.

1. *Inledning*: här tar vi upp bakgrunden till vårt arbete, de frågor som dykt upp, och de avgränsningar vi valt att göra.
2. *Metod*: här berättar vi om vår metod, som vi tog fram innan arbetet började för att veta hur vi skulle gå till väga.
3. *Förstudie*: här definierar vi vårt arbete och förutsättningarna för detta.

4. *Flödesstrukturen*: här berättar vi om arbetet med att ta fram ett regelverk för att kunna presentera olika flöden på ett enhetligt sätt.
5. *Applikationsmodellen*: här tar vi upp och redogör för arbetet med framtagningen av den grafiska flödespresentationen.
6. *Resultat*: här tar vi upp och redogör för de resultat som framkommit av arbetet.
7. *Slutsatser och diskussion*: här diskuterar vi vad vi kommit fram till utifrån resultaten, vad som kunde ha gjorts annorlunda, samt tankar och idéer inför framtiden.
8. *Referenslista*: här presenterar vi de olika källor vi använt oss av under arbetets gång.
- *Bilagor*: här presenteras sådant material som använts under projektets gång, såsom enkätsammanställning, flödesscheman och storyboard.

2 Metod

För att strukturera upp projektet arbetar vi fram ett antal metodsteg som ska följas kronologiskt. Förutom att tydligt definiera ordningsföljden på vårt arbete är syftet med de 10 stegen, som metoden består av, att ge en ökad förståelse för omfattningen av projektet.

Övergripande kan sägas att vår 10-stegsmetod (se figur 2.1) följer ett logiskt upplägg där vi börjar med kontextinsamling, för att sedan gå över i analys av kontexten, till att slutligen ta fram en flödesstruktur och skapa vår applikationsmodell. Avslutningsvis görs en utvärdering som ska visa hur applikationsmodellen förhåller sig till företagets krav och syften.

2.1 Metodstegen

1. **Inhämta kunskap om företaget och dess verksamheter**

Eftersom arbetet ska vara faktabaserat startar vi med att skaffa oss en förståelse för Informators verksamhet, detta för att vi ska kunna sätta allt i sitt sammanhang. Både en kvalitativ och kvantitativ strategi används för att samla in fakta om företaget och vilka flöden som finns, utifrån vilka vi senare kommer att skapa vår flödesstruktur. När vi sedan känner till företaget och kopplar samman detta med dess syfte och mål med applikationsmodellen, kan vi gå vidare och undersöka vilka målgrupper som finns.

2. **Specificera, kartlägga och inhämta kunskap om målgrupp**

Utifrån information som inhämtats om företaget, tillsammans med det uttalade syftet som företaget har med applikationsmodellen, vet vi nu vilket målgruppssegment som applikationsmodellen ska rikta sig mot. Är detta segment för stort eller ospecifikt, exempelvis ”alla som arbetar på ett företag”, måste det först preciseras för att vi sedermera ska kunna göra en mer djupgående analys.

3. **Analysera målgrupp och dennes prioriteringar**

Vår utgångspunkt i detta steg blir att välja ut ett antal personer från den tilltänkta målgruppen, för att sedan analysera deras prioriteringar och i vilket sammanhang de ska använda applikationsmodellen. För att få denna förståelse, sätter vi upp ett antal kriterier för vad vi behöver veta mer ingående om målgruppen. Detta kan t.ex. handla om att fråga dem om deras arbetssituation eller deras tidigare erfarenheter av e-learning. Målgruppsprioriteringarna vägs mot företagets egna behov och syften

vilket resulterar i ett antal krav för applikationsmodellens framtida utformning. Dessa krav utgör grunden för en kravspecifikation.

4. Analysera flöden och göra representativa val

I detta steg gör vi en grundläggande analys av de flöden som finns på företaget. Utifrån insamlad företagsinformation gör vi sedan en mer ingående undersökning som ska visa vad flödesstrukturen kan och ska baseras på, d.v.s. vilka flöden som finns, hur dessa är uppbyggda och fungerar.

Efter en ingående analys sorteras och kategoriseras de olika funna flödena för att möjliggöra ett representativt val. Av den totala mängd flöden vi funnit och valt att studera närmare, väljer vi ut ett eller ett par representanter ur varje grupp med likartade flöden. Genom att göra på detta viset kan strukturen skapas utifrån en bra mångfald av olika flödestyper. Detta medför att vår flödesstruktur kan användas för flöden av olika uppbyggnad och egenskaper.

5. Skapa regler för flödesstruktur och applikationsmodell

När vi slutligen har all information vi behöver om företaget, flödena och målgruppen kan vi färdigställa kravspecifikationen, som innefattar regler gällandes det fortsatta arbetet med flödesstrukturen och applikationsmodellen. Genom detta fås en klarare bild av uppbyggnaden och struktureringen för flödesstrukturen, samt utformningen för applikationsmodellen.

6. Skapa en teststruktur

Inför de första testerna mot målgrupp skapas en testspecifikation där det framgår vilka tester som ska genomföras, vad som ska testas, hur vi ska testa samt hur vi ska ta hand om testresultaten. Förutom ordinarie regressionstester så kommer applikationsmodellen att testas löpande. När skapandet av applikationsmodell sedan börjar närma sig sitt slutskede kommer dessutom ett omfattande slutttest att göras.

7. Bygga en specifik flödesstruktur

Utifrån flödesanalysen och den framtagna kravspecifikationen, innehållandes regler för hur strukturens komponenter ska fungera, kan vi bygga en specifik flödesstruktur. Det är denna struktur vi sedan baserar applikationsmodellen på. Genom att i tidigare steg ha valt ut specifika flöden, möjligtvis av olika karaktär, kan vi nu anpassa flödesstrukturen på ett sätt som möjliggör hanteringen av olika flödestyper.

När en färdig struktur sedan finns ska den presenteras för företaget.

Detta för att det ska stå klart för alla parter vad som är flödesstrukturen och vad den har för syfte i framtagandet av applikationsmodellen.

8. Bygga en applikationsmodell

När flödesstrukturen är färdig kan vi börja bygga själva applikationsmodellen, vilken ska vara anpassad utifrån den information vi fått av de tidigare stegen i processen. I applikationsmodellen ska användarna kunna utforska ett flöde för att därigenom få en förståelse för hur innehållet i detta hänger ihop, och framför allt varför det är på detta vis. Applikationsmodellen föranleds av en grundlig storyboard som visar på applikationsmodellens olika moment, som vi sedan arbetar utifrån. Tidigare liknande applikationslösningar kommer att studeras.

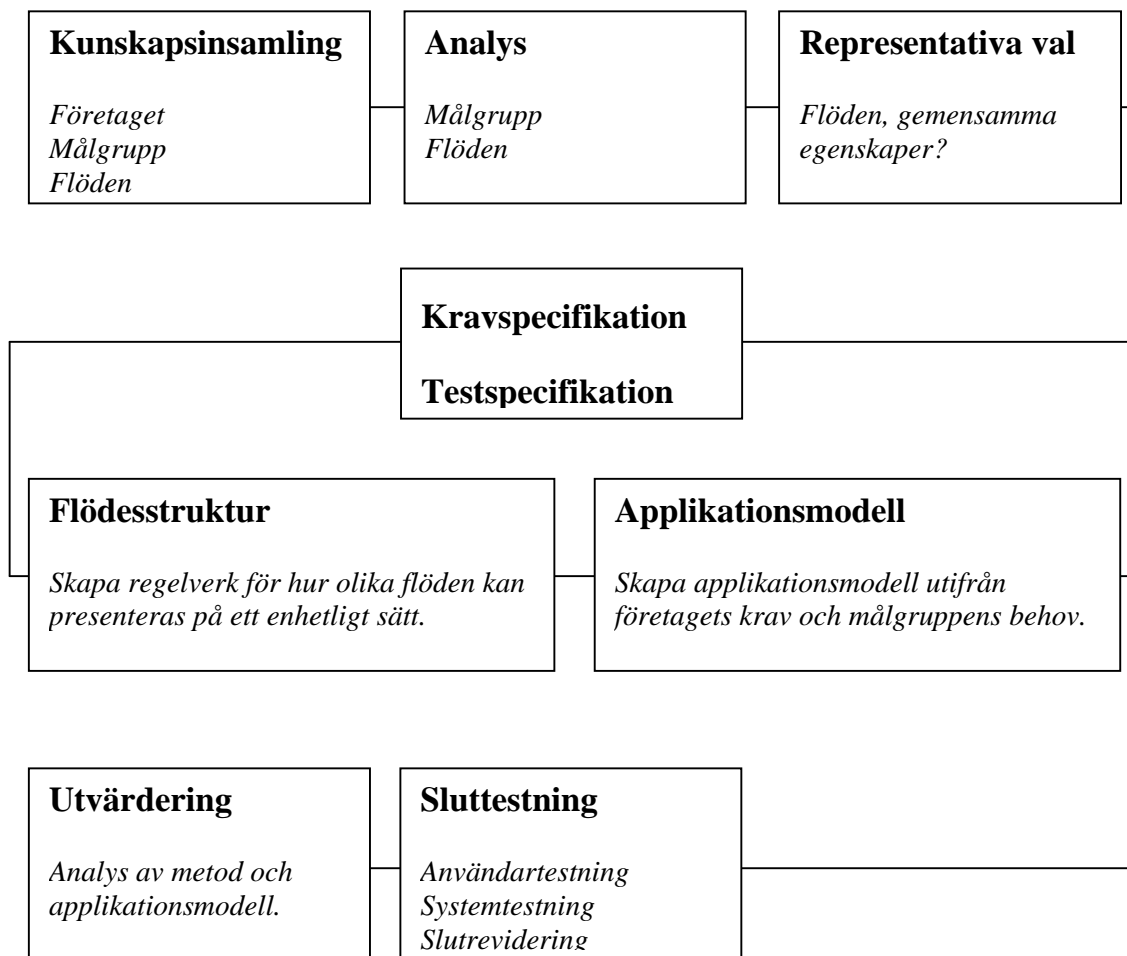
9. Sluttesta och justera applikationsmodellen

När vår applikationsmodell är färdig måste den genomgå sluttestning för att undersöka om nedtecknade krav uppfylls. En viktig del är även att analysera interaktionen mellan applikationsmodellen och målgruppen. När eventuella problem upptäcks kommer dessa att åtgärdas i applikationsmodellen, för att sedan testas återigen. Målet med denna fas är att, genom mer omfattande sluttestningar, optimera applikationsmodellen och ta fram underlag för en utvärdering.

10. Utvärdera

Som en sista del av projektet ska applikationsmodellen utvärderas för att se om denna uppfyller sitt syfte. Denna utvärdering ska sedan ligga till grund för diskussion huruvida applikationsmodellen kan och bör vidareutvecklas eller inte. Den fastslagna kravspecifikationen avgör huruvida applikationsmodellen fungerar på ett tillfredsställande sätt gentemot Informators syfte och målgruppens prioriteringar.

Andra relevanta frågor som ska besvaras i utvärderingen: Vilket värde har applikationsmodellen? Vad kan lyftas ut och bli till en mer generell modell som är företagsoberoende?



Figur 2.1 – Övergripande bild av arbetsmetoden för projektet.

3 Förstudie

Med anledning av att applikationskonceptet är nytt, både för oss och för Informator, måste vi börja från grunden med att definiera begrepp, sätta upp förutsättningar för projektet. Därefter måste vi sammanställa och analysera den kontext som är väsentlig för vidare arbete.

Förstudien genomförs utifrån både ett kvalitativt (intervjuer med utvalda anställda) och kvantitativt (enkätundersökning till samtliga anställda i företaget) arbetssätt. Den främsta anledningen till detta är att samla in så mycket information som möjligt om företaget samt den direkta respektive potentiella målgruppen för applikationsmodellen, för att därigenom skapa en bra grund att stå på.

3.1 Om Informator

Den 1 april 2007 köptes Sigma Education upp av Informator som ett led i deras arbete med att etablera sig i Öresundsregionen. Sigma Education var vid det tillfället ett utbildningsföretag som specialiserat sig på kundanpassade produktlösningar, främst inom e-learning. Samtidigt som kund- och kontaktnätet breddades, så ledde uppköpet till att Informator nu är etablerade i Sveriges tre största städer - Stockholm, Göteborg och Malmö. Om än med mindre avvikelser så jobbar samtliga lokalkontor mot samma mål;

”Informator arbetar med kunskapsöverföring som hjälper organisationer och individer att utveckla rätt kompetens. Vi ser det som en del av våra kunders verksamhets- och affärsutveckling. Syftet med ny kunskap är att organisationen ska fungera bättre. Det åstadkommer vi genom att utbilda en individ i taget.

När en individ utvecklas, får han/hon ut mer av sitt arbete och företaget får ut mer av varje medarbetare. Det leder till ökad lönsamhet, vilket ger utrymme för ännu mer utveckling.” [17]

Arbetsuppgifterna för de åtta anställda vid Malmöavdelningen, till vilken examensarbetet är förlagt, är i stort sett oförändrade sedan tiden innan uppköpet. Med tyngdpunkt på ordet kundanpassning, tar man fram och utvecklar IT-lösningar och utbildningar som är specifika från fall till fall. Den stora skillnaden mot kontoren i Stockholm och Göteborg är just specialiseringen med e-learning och utbildningar som är specifika från kund till kund. Man arbetar långsiktigt och enbart med kunden i centrum. En stor

och viktig del av arbetet är att bygga kundrelationer och aktivt förhöra sig om utbildningsbehovet hos företag, för att utifrån detta sedan utveckla och sälja nya utbildningskoncept.

3.2 Förutsättningar för arbetet

Med tanke på tidpunkten för projektstarten har semester- och tentamenstider medfört att arbetet fått anpassas därefter. Att Informator i Malmö dessutom är ett relativt litet kontor med få anställda, har sammantaget medfört att vi aktivt behövt söka den information vi behöver och inte fått något ”serverat på silverfat”.

Företagsflödena som applikationsmodellen baseras på, hämtas från Informators egen verksamhet. Situationen är därmed lite konstlad eftersom arbetet vi utför på Informator, åt Informator, i förlängningen även är tilltänkt andra företag. Vi måste därför, under tiden som vi utgår från Informators verksamhet, ha i åtanke att applikationsmodellen är tänkt att vara generell och passande för andra företag. Det specifika faktainnehållet ska baseras på Informators verksamhet, men den bakomliggande flödesstrukturen och applikationsmodellens utseende och funktion ska dock vara mer företagsberoende.

3.2.1 Informators syfte och mål med applikationsmodellen

Företaget har en lång och gedigen lista på kunder, och enligt Henrik Thorsell har i stort sett samtliga av dessa någon form av behov att kunna utbilda sina anställda hur arbetsflöden inom det egna företaget fungerar.

Informators syfte är därför att vi, genom vårt arbete, ska ta fram en modell som kan illustrera både administrativa flöden och tillverkningsflöden i rena konfigureringssyften. Grundtanken är att skapa någon form av mall som sedan kan användas oberoende av vilken typ av flöde som ska visas, d.v.s. ta fram en applikationsmodell som är företagsspecifik i sitt innehåll men som i förlängningen kan tillämpas på ett företag, vilket som helst.

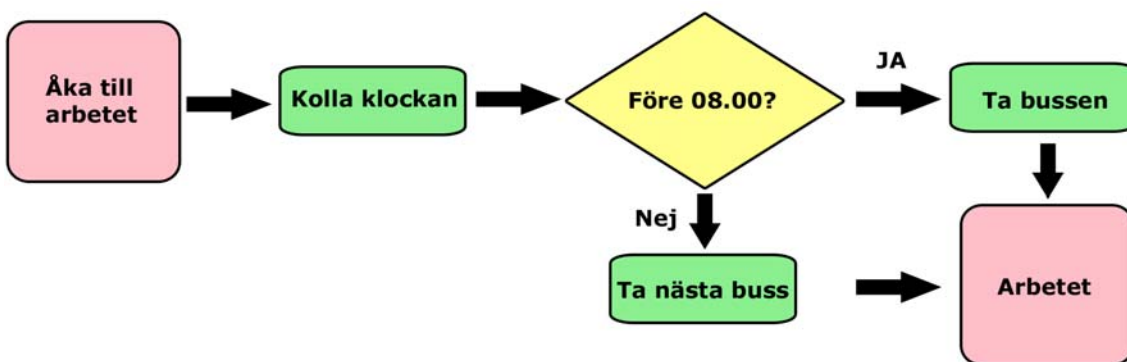
Tre konkreta huvudmål finns uppställda från företagets sida när det gäller arbetet:

1. Företaget vill ta fram en applikationsmodell som kan simulera både tillverkningsflöden och administrativa flöden, och sedan tydliggöra dess logiska kopplingar för användarna.
2. Företaget vill att applikationsmodellen ska vara utvecklad för webbanvändning.

3. Företaget vill att applikationsmodellen i förlängningen ska vara oberoende i den mån att den ska vara tillämpningsbar på ett flöde eller företag vilket som helst.

3.3 Flöden och flödesapplikation – vad menas?

Ett flöde i detta arbetets sammanhang är en rad uppgifter på ett företag som på något sett är beroende av varandra och som utförs i en viss sekvens. Start och mål finns, exakt väg däremellan är dock beroende på användarens val.

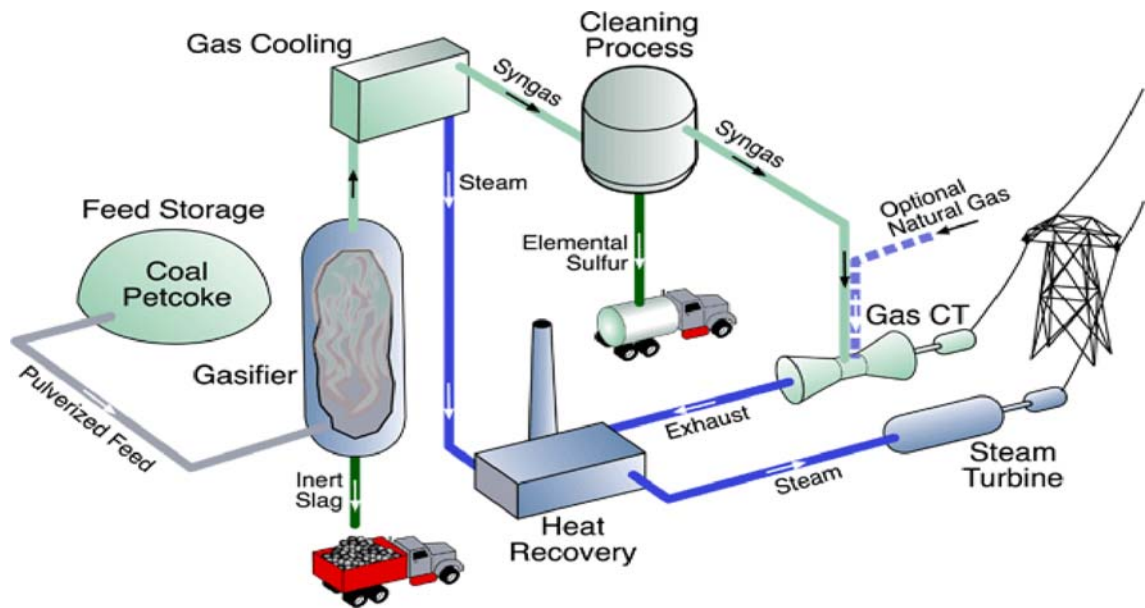


Figur 3.1 – Typexempel på ett vardagligt flöde.

Ett annat, kanske vanligare, ord för flöde är process. Mer exakt handlar det om en normativ³ process eftersom vi redan skapar färdiga val vilka användaren kan välja för att nå till ett visst mål.

”En process är en serie aktiviteter som förädlar en vara eller tjänst. Den har en väl definierad början och ett väl definierat slut. Det måste alltid vara klart definierat vad processen ska ta emot, vad den ska åstadkomma, hur det ska ske samt vilka de förväntade resultaten ska vara.” [18]

³ Normativ – av latinets ord *Norma*, rättesnöre, ett mönster varefter man i sitt handlande rättar sig efter en given struktur.



Figur 3.2 – Typexempel på en företagsprocess. [16]

En flödesapplikation är enligt vår definition därmed ett gränssnitt i vilket användaren sedan på ett interaktivt sätt ska kunna utforska flöden.

3.4 Framgångsrecept för lyckad e-learning

För att försäkra sig om att den e-learningapplikation man vill skapa verkligen ska fungera och engagera, finns det en del viktiga nyckelpunkter som man bör ta hänsyn till. Punkterna nedan är fritt översatta från en artikel i tidskriften *Performance Improvement* [15] och handlar om att skapa lyckade e-learningapplikationer.

- Är användarna i målgruppen bekanta med e-learning sedan tidigare?
- Vilka teknologier finns tillgängliga för att främja applikationens innehåll?
- Har användarna tillräckligt med kunskap för att kunna använda dessa teknologier?
- Vilka förkunskaper är det bra för användarna att ha?
- Hur lång tid har man tänkt att användarna ska ägna åt inlärningsprocessen?
- Hur lång tid har användarna själva tänkt ägna åt inlärningsprocessen, d.v.s. hur mycket är var och en beredda att engagera sig?

- Vilka konkreta resultat vill man uppnå genom aktiviteten?
- Vilka andra målsättningar har man med aktiviteten?

Artikeln tar upp och beskriver hur man kan göra traditionella, ibland tråkiga, inlärningsmetoder för skolundervisning mer meningsfulla och interaktiva. Genom att ta metaforer från verklighetstroga klassrumssituationer och blanda detta med de teknologier som erbjuds, kan man skapa e-learningkurser som exalterar eleverna och får dem att engagera sig. Detta kan i mångt och mycket direkt liknas vid vår projektsituation med att skapa en applikationsmodell för ett företag. Klassrumsmiljön är inte helt olik en arbetsmiljö där tiden för inläring är begränsad. De enda skillnaderna är att målgruppen inte är studenter utan företagsanställda, och själva klassrummet istället är arbetsplatsen.

3.5 Målgruppen för applikationsmodellen - definition och analys

Informators tänkta målgrupp för applikationsmodellen i detta utvecklingskede är anställda med en ledande befattning, t.ex. verksamhetschefer och projektledare. Det finns även en annan potentiell målgrupp, nämligen nyanställda inom företaget. Vi väljer dock att inte fokusera på den sistnämnda målgruppen under vårt projekt eftersom det inte finns några nyanställda på Informator.

Utifrån den givna målgruppen söker vi upp matchande personer inom företaget och studera dessa närmare genom en målgruppsanalys. Ett problem uppstår dock här då det är ganska få anställda vid varje kontor, vilket medför att de personer som jobbar på det lokala kontoret i Malmö inte kan anses som representativt för hela målgruppen.

Med anledning av detta genomför vi en fallstudie i form av ett enkätformulär via Internet (se bilaga 1), i vilken vi vill få svar på mer allmänna frågor inför skapandet av en kravspecifikation. Vi adresserar enkäten till samtliga anställda inom företaget, för att få ut så täckande information som möjligt. En bieffekt är dessutom att detta ger oss en bättre bild av hela företaget och personalsituationen på Stockholms- och Göteborgskontoren.

Att samla in information av olika karaktär, för att skapa en bra helhetsbild av det aktuella fallet, är en vanlig och beprövad företeelse. När det som i detta fall handlar om en arbetsplats bör både intervjuer, observationer och enkäter kombineras [10].

Genom denna första kvantitativa undersökning får vi enkätsvar som till stor del består av rena fakta, och som spelar en viktig roll när det gäller de tekniska krav som ska ställas upp för vår applikationsmodell. Ur mängden information vi får in, lyfter vi ut de svar som är väsentliga för vårt fortsatta arbete. Exempel på detta är skärmupplösning och möjlighet till ljuduppspelning på arbetsdatorn.

Nackdelen med en sådan enkät är dock att svaren är ganska opersonliga. Vi får heller ingen förståelse för hur personen verkligen känner i en viss situation. Med anledning av detta kompletterar vi vår kvantitativa undersökning med en mer kvalitativ. Denna kvalitativa undersökning går ut på att helt enkelt intervjuar personer som tillhör målgruppen, samt att genom observationer få en förståelse för målgruppens tyckande, tänkande och handlande i olika situationer.

”Syftet med kvalitativa undersökningar är att skaffa en annan och djupare kunskap än den fragmentiserade kunskap som ofta erhålls när vi använder kvantitativa metoder.” [13]

Undersökningarna underlättas genom att vi befinner oss på samma arbetsplats som målgruppen under vårt arbete. Allt efter behov kan vi därför genomföra observationer, som generellt sätt är ett bra redskap för att komplettera information som man får in via intervjuer och enkäter [4, 12]. Detta ligger sedan till stor grund för applikationsmodellens utformning.

Sammantaget är upplägget för målgruppsundersökningarna sådant, att alla anställda får svara på samma frågor och får delge sina personliga synpunkter och erfarenheter. Frågorna är över lag öppna ställda, endast sådana frågor som spelar en viktig roll för applikationsmodellens tekniska krav har mer eller mindre fasta svarsalternativ. Eftersom både enkäten och intervjun till stor del består av öppna frågor, så är undersökning således standardiserad men av låg struktureringsgrad (se figur 3.3).



Figur 3.3 – Exempel på olika typer av intervjuer och enkäter beroende på hög eller låg grad av standardisering och strukturering. [11]

3.5.1 Undersökningsresultaten

Cirka 40% av de anställda har svarat på 90% av de ställda frågorna (se bilaga 2). Siffran är visserligen för låg för att ge en statistisk säkerhet till utfallet, men sammantaget med den information vi fått av Henrik Thorsell, och av våra egna erfarenheter från arbetssituationen på Informator i Malmö, så väljer vi att behandla underlaget som representativt för företaget.

I samband med den kvantitativa undersökningen har en del tekniska fakta framkommit, fakta som vi kommer att använda i den målgruppsspecifika delen av kravspecifikationen:

- Samtliga använder något operativsystem från Microsoft. Vanligast är Windows XP.
- Hög skärmapplösning på arbetsdatoren, minimum 1024*768 bildpunkter.
- Samtliga har möjlighet att spela upp ljud på arbetsdatoren.
- Vanligaste webbläsaren är Internet Explorer 6.
- Vanligaste programmen är de som finns i Officepaketet. Abalon, Informators eget administrationsprogram, används också flitigt.

Vi vet nu också att de flesta i vår målgrupp är män i 40-årsåldern. De flesta har arbetat på samma position i flera år och torde därför vara bra insatta i den verksamhet de jobbar inom.

När vi frågar vilken datorvana de anser sig ha, på en skala från ett till fem, svarar de flesta fyra eller fem. Om man jämför detta med vilka program som används, och deras svar om operativsystem etc. kan man dra slutsatsen att den uppskattade datorvanan mestadels gäller de program de hanterar dagligen. Däremot är den allmänna datorvanan troligen mer medelmåttig. Någon visste exempelvis inte vad ett operativsystem är, eller vilket som fanns på arbetsdatorn.

Vi försöker också få en bättre insyn i målgruppens allmänna e-learningkunskap, genom att fråga om deras tidigare erfarenheter, synpunkter och reflektioner i ämnet. Det visar sig att ungefär hälften har någon form av tidigare erfarenhet, antingen genom att de själva utvecklat eller genomfört en sådan utbildning. När det gäller dem utan tidigare erfarenhet har de flesta en någorlunda klar bild av att e-learning är någon form av utbildning som sköts via Internet, där användaren kan lära in något i egen takt. Att målgruppen har en allmän kunskap om e-learning är en värdefull aspekt som vi tar hänsyn till när vi utformar applikationsmodellens upplägg.

Genom intervjuer har de anställda också fått berätta lite om deras arbetssituation och vad de rent konkret arbetar med, varvid vi kunnat bilda oss en uppfattning om vilka flöden som finns och som vi kan ha användning för. Det visar sig att flera medarbetare, på samma position i företaget, inte har samma bild av vad deras arbete går ut på. Det framkommer också att det finns stora skillnader mellan de olika kontoren inom företaget. Bland annat är det lokala Malmökontoret specialiserat på utveckling av e-learningtjänster, något som inte återfinns på de övriga kontoren.

Även olika former av observationer har gjorts. Bland annat fick en medarbetare visa oss hur administrationsverktyget Abalon fungerar. När personen i fråga satt och jobbade med detta, märktes tydligt vilka svårigheter som fanns, och hur personen reagerade på en viss typ av information, såsom hjälptexter etc. Uppenbara problem förelåg i att förstå varför allt fungerade som det gjorde.

En slutsats av vår undersökning är att Informator och dess verksamhet har stora likheter med andra företag. Organisationen, med administration, försäljning, produktion, projektledning etc, och rådande arbetssituation speglar i mångt och mycket hur det ser ut inom andra företag. Detta antagande är också något som Henrik Thorsell, med sina erfarenheter, går i god för. Med

detta som underlag anser vi därför att den målgruppsundersökning vi gjort kan anses som representativ både för Informator och för många andra företag.

3.5.2 Andra potentiella målgrupper

Under vårt arbete riktar vi in oss på målgruppen verksamhetschefer och projektledare. Även andra människor i organisationen skulle dock kunna ha nytta av applikationsmodellen, då det med all säkerhet finns ett stort allmänintresse i att förstå varför man jobbar på så sätt som man gör. Därför finns det andra målgrupper som också kan tänkas nyttja applikationsmodellen. Den mest uppenbara är kanske en nyanställd som snabbt och lätt ska få en bra bild av hur arbetet fungerar, för att sedan kunna börja arbeta på ett effektivare sätt. Man kan också tänka sig att en befintlig anställd ska byta position inom företaget, och behöver lära sig något nytt. Då kan vår applikationsmodell komma till nytta.

Eftersom applikationsmodellen är tänkt att vara anpassad för olika sorters flöden inom både produktion och administration, så skulle den kunna passa både till dem som jobbar vid ett löpande band på ett tillverkningsföretag, eller de som skriver journaler på ett sjukhus. I båda fallen skulle det kunna vara möjligt för användaren att lära sig varför något gått fel i produktionen respektive administrationen.

3.5.3 Typiska användare ur målgruppen

Följande genomsnittsanvändare utgör en beskrivande bild över målgruppen för applikationsmodellen:

1. Typexempel – säljchef

Magnus Forslund, 37 år, är säljchef på ett bemanningsföretag i Stockholm. Hans viktigaste arbetsuppgift är att gå på kundmöten, för att skapa bra relationer till kunderna. Magnus tror att han är bra på det här med datorer, men använder i själva verket bara Outlook för att hålla koll på sina möten. Någon enstaka gång skriver han också ihop ett Word-dokument.

2. Typexempel – administrationsansvarig

Kristina Andersson, 32 år, har precis kommit tillbaka till arbetet efter mammaledigheten och har fått en ny befattning som ansvarig för administrationen. Detta innebär att hon ansvarar för att boka in diverse möten med kunder och säljare. Kristina är inte helt bekant med de nya administrationsprogrammen och hon råkar göra fel gång på gång.

3. Typexempel – projektledare

Anton Karlsson, 24 år, jobbar som produktutvecklare och tillika projektledare. Han arbetar i mindre projekt där han programmerar och skapar nya hemsidor. Anton har en mycket bra datorkunskap både när det gäller hård- och mjukvara.

3.5.4 Målgruppen och dennes inflytande på applikationsmodellen

Rent allmänt kan sägas att Informators syfte och krav fungerar som någon form av filter för målgruppens prioriteringar. Samtidigt som vi tar hänsyn till det som framkommit av målgruppsundersökningen, måste vi rätta oss efter de krav som ställs från Informators sida, eftersom de är huvudintressenten i arbetet.

Det kan tänkas att konflikter och motsättningar uppstår när företagets syfte inte stämmer överens med målgruppens prioriteringar. I vårt fall är detta dock inget problem eftersom vi har ganska fria händer både vad gäller applikationsmodellens funktion och dess innehåll. Samtliga kravförslag som vi har lagt fram utifrån målgruppsundersökningen har godkänts av Henrik Thorsell. De enda kraven som ställts från företagets sida är att applikationsmodellen ska kunna visa olika typer av flöden samt att allt ska vara utvecklat i Flash för webbanvändning.

Applikationsmodellens utformning kommer därmed till stor del att baseras på våra undersökningsresultat, och därmed målgruppens prioriteringar. Förutom det som framgår av kapitel 3.7 – Kravspecifikationen, så tar vi hänsyn till följande saker:

- **Passande och relevant applikationsupplägg**
Stilrenhet och funktionalitet är vad som efterfrågas av målgruppen. Applikationsmodellen med dess olika funktioner ska vara intressant, men för den skull inte nödvändigtvis rolig. Till detta ska det finnas en lagom kombination av teori och praktik. Det allmänna utseendet och ev. metafor som används för att visa handlingsförloppen för de olika flödena, ska vara relevant och passande för ett företag.
- **Grundläggande funktionalitet**
Målgruppen har en grundläggande datorvana. De är dock generellt sätt inga experter på övriga datorområden och applikationsmodellens funktion bör därför inte vara mer komplicerad än de vanligaste programmen som används på arbetet. Det ska t.ex. inte finnas ett flertal menyer som användaren ska behöva navigera sig igenom och riskera att ”gå vilse i”.

- **Enkel att komma igång med**

Då applikationsmodellen ska kunna användas på en arbetsplats så antas tiden för användandet stundtals vara mycket begränsad. En fullt tänkbar arbetssituation, där användaren snabbt måste kunna starta upp och komma igång, är den när chefen kommer in och säger: ”Om en halvtimme ska vi börja lägga in budgeten, precis som vi gjorde vid den här tidpunkten i fjol. Gå in och gör en snabb repetition av flödet i ekonomisystemet så att du vet hur du ska göra – och just det, du ska även kunna förklara för Nils från bemanningsföretaget hur allt hänger ihop eftersom han kommer att hjälpa oss.” Genom att då plocka bort onödiga valmöjligheter och inställningar som måste göras vid uppstart, så gör man det enkelt för användaren att komma igång. På så sätt kan man även i tidspressade situationer känna att man hinner med att köra igenom ett eller ett par flöde, istället för att man blir stressad av att ödsla dyrbara minuter på inställningar.

- **Relevant och personligt presenterad informationstext**

Det ska vara lätt att läsa och ta till sig texten som presenteras. Därmed bör texten som målgruppen efterfrågar vara kortfattad men innehållsrik. För de som är intresserade ska det sedan finnas en möjlighet att läsa vidare om det finns mer ämnesinformation, exempelvis genom externa länkar. Textinnehållet ska även vara både intressant och relevant.

- **Minimera behov av hjälpfunktioner**

E-learning i sig medger inte någon fysisk hjälp, eftersom själva tanken med detta utbildningssätt just är att något ska kunna läras ut på distans via Internet. Därmed är det också av största vikt att sträva efter att skapa en så självförklarande applikationsmodell som möjligt, som kan användas med minimal hjälp. Erfarenheten från våra observationer säger oss nämligen att användare inte har tid eller lust att sitta och studera en mängd hjälpfunktioner för att förstå hur man ska göra, särskilt inte i stressade miljöer eller situationer.

- **Olika former av multimedia och interaktivitet**

Bilder, ljud och video ska om möjligt användas som komplement till text. Detta gör vi för att anpassa applikationsmodellen både till dem som tycker om att ta till sig informationen i skriven form, och för dem som hellre vill lära sig genom att studera bilder eller någon videosekvens. Interaktivitet använder vi för att kunna presentera innehållet på ett intressant sätt, i syfte att bibehålla användarnas intresse och skapa en ökad förståelse av det som presenteras.

Applikationsmodellen ska därför medge att användaren aktivt får prova olika handlingsförlopp när det gäller de olika flödena, och få utforska

flödets olika delar på sitt eget sätt, för att på så vis få en djupare förståelse för hur allt hänger ihop [5].

3.6 Företagsflöden hos Informator

För att veta vilka flöden vi har att tillgå, och då kunna studera dessa närmare på egen hand, utgår vi från de vi finner runt om i vår omgivning på Informator.

Flertalet av de funna flödena, såsom certifiering eller bokning av lärare, har vi funnit och sammanställt efter intervju av målgruppen. Några flöden har vi själva skrivit ned efter observationer av det dagliga arbetet, som t.ex. att skicka fax eller att ringa ett telefonsamtal.

Eftersom det är vår egen bild av flödena som vi antecknat, behövs det på något sätt också verifieras att de verkligen ser ut och fungerar som vi hade tänkt. Vi väljer därför att noggrant rita upp flödena och förklara dem i ord, för att sedan gå igenom dem tillsammans med Henrik Thorsell.

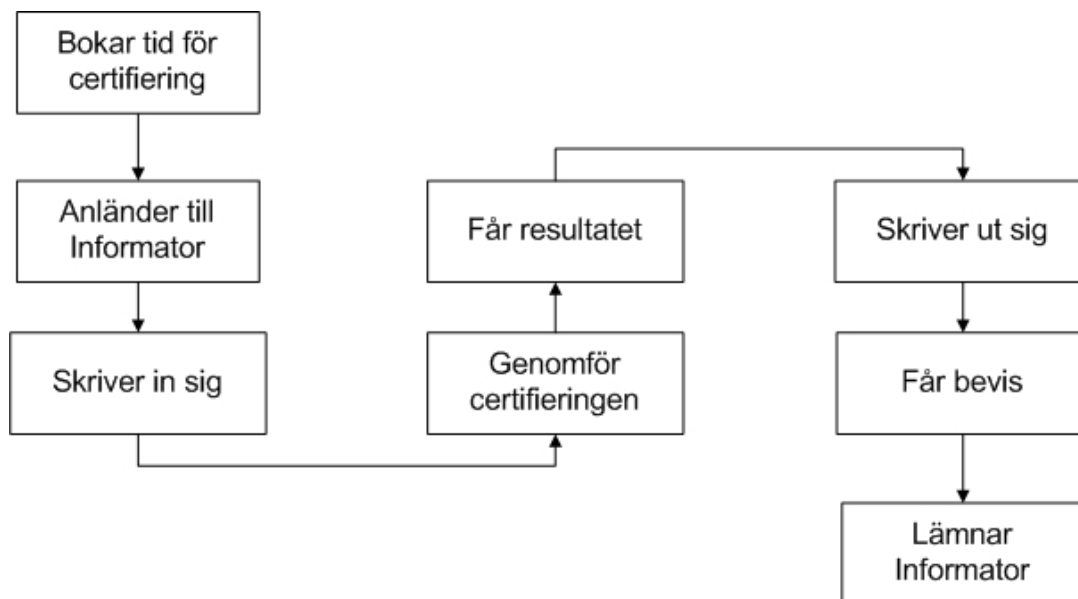
Av de flöden vi funnit så är det totalt nio stycken som vi väljer att ta en närmare titt på. Dessa delar vi in i tre huvudgrupper:

- Administrativa
- Tekniska
- Vardagliga

Några av dessa flöden är snarlika, medan många har tydliga skillnader både i sitt innehåll och i sin uppbyggnad. Vår applikationsmodell ska kunna presentera flöden från samtliga tre huvudgrupper.

- **Certifiering**

På Informator kan man bli certifierad (se figur 3.4), vilket innebär att man får ett bevis på att man kan hantera ett visst program eller ett större IT-område. Certifieringen sker genom olika prov framför en dator. Förfarandet vid certifieringen, ur ett kundperspektiv, består i att; deltagaren anländer till Informator, skriver in sig, gör provet, får resultatet, skriver ut sig och lämnar Informator igen. Detta är ett bra exempel på ett helt linjärt flöde, där varje steg måste göras i exakt denna ordning.



Figur 3.4 – Certifieringsflöde hos Informator ur ett kundperspektiv.

- **Affärsadministration**

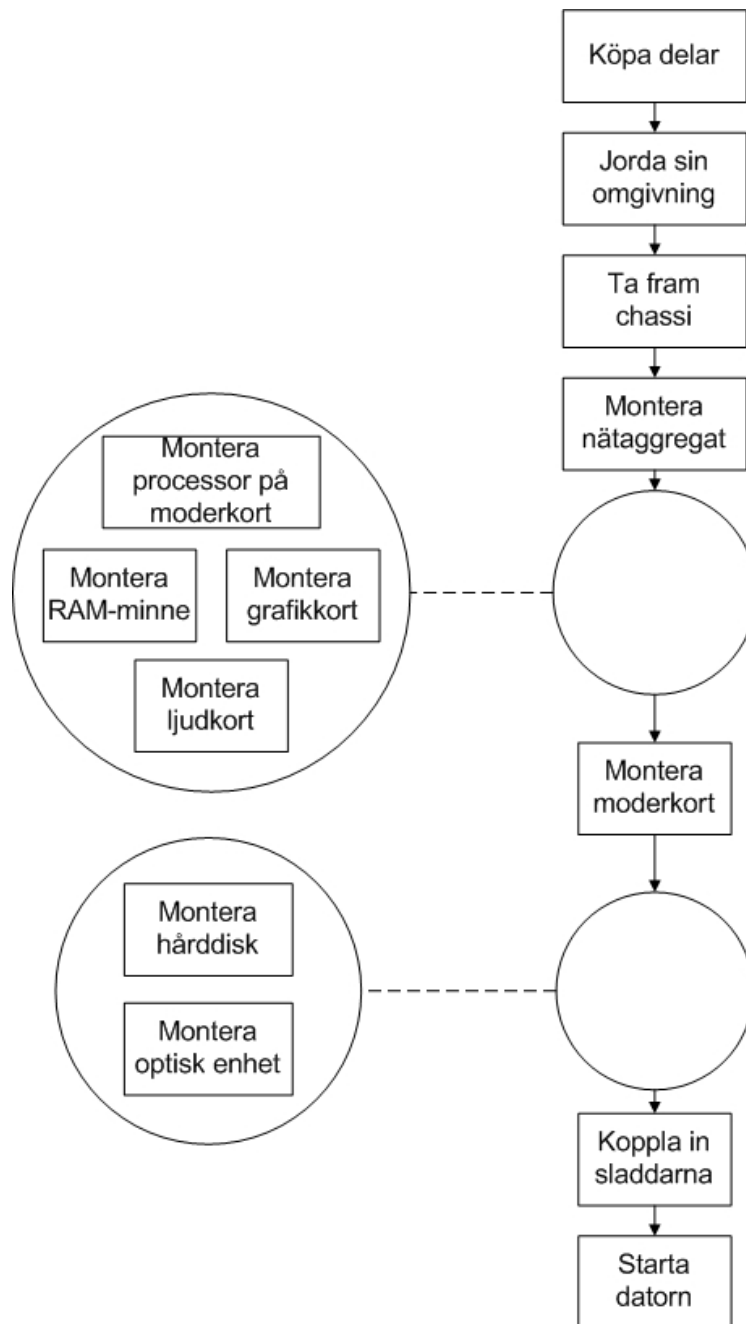
Detta administrativa flöde beskriver hur Informators affärsidé fungerar. Genom att först träffa kunden får Informator reda på vad kunden vill ha. Sedan skickas en expert till kunden för att undersöka vad det är för utbildning kunden behöver, vilket dokumenteras och sammanställs. Man kommer överens med kunden om upplägget för kursen och skickar en offert. Om denna godkänns startar man utbildningen, annars får man diskutera vidare. Utbildningen slutförs, en utvärdering görs och fakturan skickas till kunden.

- **Kursadministration**

Kursadministrationsflödet kan ses som en del i affärsflödet, och handlar i princip om att administrera själva utbildningsdelen till en kund. Detta görs genom att boka in tid och plats samt att hitta en lämplig lärare, som kunden sedan får godkänna. Material skickas därefter till läraren och kursen genomförs som planerat. Slutligen görs en kursutvärdering.

- **Datorkonstruktion**

Detta är ett bra exempel på ett configurationsflöde där man visar hur en fysisk produkt är uppbyggd. Datorbyggandet erbjuder möjligheten att sätta ihop vissa komponenter i olika ordning, och ändå uppnå samma slutresultat. Dessutom finns det, precis som i tidigare beskrivna flöden, även vissa moment som måste göras i en viss ordning. Exempelvis bör man inte sätta igång strömmen innan datorn är färdigbyggd. Däremot är det fullt möjligt att montera minnen på moderkortet före processorn, eller tvärtom. Detta illustreras genom de stora cirklarna i figur 3.5.



Figur 3.5 – Ett möjligt flöde över hur man bygger en dator. Oberoende valmöjligheter symboliseras av de stora cirklarna.

- **Kopiering**

Ett kopieringsflöde är ett vardagligt flöde, som beskriver hur man tar en kopia i en kopieringsmaskin. Kopieringsflödet är linjärt, och har endast ett fåtal steg. Det fungerar på så sätt att man lägger i sitt original, för att sedan trycka på startknappen. Kopieringen utförs och sedan tar man ur kopiorna och sitt original.

- **Ringa ett samtal**

Att ringa ett samtal är en enkel uppgift att genomföra. Flödet är linjärt och består i att lyfta luren, få kopplingston, slå en nolla för att komma ut på linjen, sedan slå telefonnumret och komma fram till den man ringer till. Man skulle dock kunna tänka sig andra vägar att gå, exempelvis att man slår numret innan man lyfter luren, eller att man måste ringa igen ifall man inte kommer fram på linjen.

- **Skicka ett fax**

Att skicka ett fax är i princip samma flöde som att ringa ett samtal. Skillnaden är att ytterligare ett steg tillfogas processen, nämligen att man matar in papper i faxen för sändning.

- **Koka kaffe**

Att koka en kanna kaffe är till synes ett väldigt enkelt flöde. Flödet har stora likheter med t.ex. affärsflöde eller datorbygge, men är betydligt kortare. I princip består flödet av att fylla maskinen med vatten och kaffe och trycka på startknappen. Även här finns det dock saker som kan utföras i olika ordning, men som ändå leder fram till samma slutresultat. T.ex. spelar det ingen roll om man fyller på vattnet eller kaffet först, så länge du gör båda delarna så kommer du fram till samma resultat.

- **Abalonflöde**

CRM-systemet⁴ Abalon används av några få anställda på Informator för att bl.a. lägga upp kunder, boka kurser och registrera inbetalningar. Programmet är väldigt komplext och samlar i stort sett hela företagets administration. Endast ett fåtal anställda har tillgång till programmet, och ännu färre förstår hur det ska användas.

Av de flöden som vi funnit men inte väljer att titta närmare på, kan nämnas sådana som är av mycket alldaglig och basal karaktär, exempelvis ”Gå på toaletten” och ”Gå och handla fika”. Dessa exempel visar tydligt att allt i vardagen som beskriver hur något utförs är flöden i sig. Därmed kan man (som vi också gjort) finna nästan oändligt många flöden i sin omgivning. Sådana grundläggande flöden är dock inte relevanta för vårt arbete, eftersom det inte är på denna nivå som Informator önskar att en framtida applikation ska ligga på.

⁴ CRM – Customer Relationship Management. System som går ut på att kundernas behov och önskemål ska tillgodoses bättre via samordning av tidigare splittrad information om kunderna i kunddatabaser, säljstödssystem och affärssystem. [19]

För att arbetet ska vara faktabaserat så har vi totalt sett bara kunnat välja bland de flöden vi funnit på Informator. Man kan tycka att exempelvis ett inköpsflöde är något som är relevant för vårt arbete och som borde tas med i urvalet. Någon tydlig struktur för ett inköp finns dock inte på Informator enligt Henrik Thorsell. Därmed finns det heller inget att ta fasta på när det gäller att konstruera ett sådant flöde, och i längden inget att lära ut.

3.7 Kravspecifikation

Vi har två huvudsyften med kravspecifikationen. Det första är att få en klar bild av hur applikationsmodellen ska anpassas så att vi vet hur vi ska arbeta. Det andra handlar om att vi i slutskedet verkligen ska kunna göra en utvärdering av applikationsmodellen, som ska kunna visa på om det vi tagit fram motsvarat företagets syften och krav.

Kravlistan nedan har redigerats fortlöpande då nya krav har dykt upp, eller gamla blivit inaktuella. Systemkraven, som godkänts av Henrik Thorsell, har mestadels författats utifrån målgruppsanalysen.

3.7.1 Projektkrav

1001 Projekttidsplan ska uppdateras kontinuerligt.

1002 Samtliga flöden som används i arbetet ska gå att finna på Informator.

3.7.2 Systemkrav

3.7.2.1 Flödesstrukturen

2001 Flödesstrukturen ska vara baserad på minst tre företagsflöden.

2002 Flödesstrukturen ska kunna hantera de flöden som den baserats på.

2003 Flödesstrukturen ska kunna hantera en given start- och slutpunkt.

2004 Flödesstrukturen ska kunna hantera processblock.

2005 Flödesstrukturen ska kunna hantera olika förgreningar (vägval).

2006 Lösningsförslag för flödesstrukturen ska presenteras för företaget.

3.7.2.2 Applikationsmodellen

2007 Applikationsmodellen ska göras i Adobe Flash 8.

2008 Distributionskanal för applikationsmodellen ska vara Internet.

2009 Distributionsformatet ska vara swf.

2010 Ska kunna spelas upp korrekt i Internet Explorer version 6 och senare.

- 2011 Ska kunna spelas upp korrekt i Flash Player 8.
- 2012 Databasversion är Microsoft SQL Server 2000.
- 2013 Serverprogrammeringsspråk är PHP.
- 2014 Ska fungera på upplösningen på minst 1024x768.
- 2015 Ska fungera på en dator med Microsoft Windows XP som OS.
- 2016 Applikationsmodellen ska uteslutande köras i ett webbläsarfönster.
- 2017 Applikationsmodellen ska kunna visa minst två flödesexempel.
- 2018 Minst ett administrativt flöde ska finnas som exempel.
- 2019 Minst ett tekniskt flöde ska finnas som exempel.
- 2020 Minst ett vardagligt flöde ska finnas som exempel.
- 2021 Storyboard ska göras över applikationsmodellen.
- 2022 Applikationsmodellens textinnehåll ska vara svenskbaserat.
- 2023 Programkod ska vara kommenterad.
- 2024 Tillåtna filformat för bilder är jpg, gif, png, pfd.
- 2025 Tillåtna filformat för ljud är mp3, wav.
- 2026 Tillåtna filformat för video är avi, mpeg.
- 2027 Krav utgått!
- 2028 I möjligaste mån ska allt som är klickbart innebära en förändring av muspekaren.
- 2029 Applikationsmodellen ska testas löpande av minst en person ur målgruppen.
- 2030 Obligatoriska ljudmoment ska ej finnas med tanke på användningsmiljön för applikationsmodellen.
- 2031 Krav utgått!

3.7.3 Kvalitetskrav

- 3001 Inga fel som förhindrar exekvering får finnas i applikationsmodellen.
- 3002 Alla fel som upptäcks vid slutttestningen ska nedtecknas.
- 3003 Applikationsmodellen ska uppfylla samtliga ovanstående krav vid leverans.

3.8 Testspecifikation

För att applikationsmodellen ska anses som godkänd ska vi kunna visa på att samtliga punkter i kravspecifikationen är uppfyllda. I detta kapitel anger vi därför rutiner för att, genom ett flertal tester, försäkra oss om att det löpande arbetet leder till att vi uppnår detta kvalitetskrav.

Anledningen till att vi sammanställer en testspecifikation innan vi utvecklar applikationsmodellen, är att vi redan från start kommer att vilja testa mot utvalda personer ur målgruppen. Därför måste det redan vid implementeringsstart stå klart för oss vad vi ska testa, och hur detta ska göras. I denna testspecifikation framgår det vilka test vi ska genomföra, vad vi ska testa, hur vi ska testa och hur vi tar tillvara på testresultaten.

3.8.1 Teststruktur

Samtliga tester skall utföras mot ställda krav och ska verifiera om funktioner, gränssnitt och egenskaper följer kravspecifikationen. Merparten av testerna kommer att utföras på Informator i Malmö där målgruppen befinner sig.

Totalt kommer vi att genomföra tre olika typer av tester; regressionstester, löpande målgruppstester och sluttest.

Regressionstester och löpande tester av applikationsmodellen är de vanligast förekommande. Dessa kommer att genomföras kontinuerligt under hela implementeringens gång, där vi då kontrollerar hur enstaka smådetaljer fungerar, såsom en viss knappfunktion eller videosekvens. Dessutom kommer ett fåtal utvalda personer utanför den direkta målgruppen att hjälpa oss med att testa applikationsmodellen löpande. Dessa kan då ses som en potentiell målgrupp då de skulle ha kunnat vara nyanställda i företaget.

Förutom de löpande testningarna så kommer applikationsmodellen även att genomgå sluttestning. Personer ur målgruppen kommer då att få testa applikationsmodellen utifrån nedskrivna scenarion. Sluttesterna har som syfte att testa hur helheten fungerar, genom att simulera verklighetstroga fall som kan uppkomma inom ramen för applikationsmodellens användningsområde. Ett systemtest kommer också att genomföras för att se om applikationen uppfyller ställda krav.

Genom att genomföra testerna på ovanstående sätt arbetar vi iterativt – implementering, testning, korrigerig (omarbetning), testning. Vi går från att testa mindre och enstaka implementeringar (regressionstest och löpande test av applikationsmodellen), till utvalda sammanhängande delar som anses färdigimplementerade (löpande test av applikationsmodellen), till att slutligen

testa helheten (systemtest och scenariotest). Tanken är att upptäcka ev. svårigheter eller felaktigheter snarast möjligt efter en implementering.

Testmomentet anses vara klart när applikationsmodellen har passerat sluttestet och vi, tillsammans med Informators representant Henrik Thorsell, jämfört resultatet mot kravspecifikationen.

Eventuella fel som kvarstår efter testningen ska dokumenteras i resultatsammanställningen.

3.8.2 Regressionstest

Regressionstester genomförs löpande av oss själva under hela arbetet. Dessa är viktiga för att se att nya funktioner som implementeras fungerar tillsammans med befintliga delar.

Syftet med regressionstesterna är att kunna spåra i vilket skede något gick fel, och varför det gick fel just då. Därmed är det också viktigt att testerna utförs grundligt och kontinuerligt. Resultaten diskuteras muntligt och eventuella problem åtgärdas direkt.

3.8.3 Löpande test

Att testa applikationsmodellen löpande ger oss en förståelse av hur användarna hanterar applikationsmodellen allt eftersom nya moment tillkommer. Vi undersöker exempelvis om användarna klarar själva flödesutforskningen, hur de utnyttjar den hjälp som finns att få, om de klarar av navigationen etc.

Testningen genomförs under hela arbetets gång men mest intensivt under slutfasen. Dessa löpande tester är kort utformade och genomförs för att vi ska kunna få feedback på hur målgruppen hanterar enskilda moment. Detta kan bestå i att testa om en enskild implementerad detalj fungerar och ser ut som det var tänkt, exempelvis om testpersonerna kan hantera navigeringen på rätt sätt.

Testerna utförs främst mot personer ur målgruppen, men även personer utanför den direkta målgruppen kommer att få testa applikationsmodellen löpande. Detta för att vi ska få löpande feedback från andra som använder applikationsmodellen mer förutsättningslöst.

Resultatet från de löpande testerna ligger hela tiden till grund för det fortsatta arbetet av applikationsmodellen. Om problem eller fel upptäcks under testningen, har vi möjlighet att rätta till dessa och sedan testa på nytt.

Testresultaten redovisas i en sammanställd form i kapitel 5.9 – Löpande test och regressionstest, tillsammans med ställningstaganden för vad resultaten ska leda till.

Mycket av den feedback vi får av de löpande testerna utgör underlag för skapandet av testscenarion inför slutttestningen.

3.8.4 Slutttest

När applikationsmodellen anses nästintill färdigimplementerad ska den kontrolleras och slutttestas. Detta sker i flera steg, där vi själva först ser till att vår applikationsmodell uppfyller de krav som är ställda. När detta är gjort kan vi sedan göra ett sista test av applikationsmodellen mot målgruppen, med uppställda scenarion som ska köras igenom.

Genom att testpersonerna utsätts för verklighetstroga scenarion och får en längre stund på sig att genomföra de uppgifter som ges, kan vi observera dem och få fram viktiga uppgifter om hur vår applikationsmodell fungerar gentemot användarna.

Resultaten dokumenteras i rapporten under kapitel 5.11 – Slutttestning. De problem som upptäcks skrivs ned och behandlas enskilt. Fel som gör att ett krav i kravspecifikationen inte kan uppfyllas, definieras som allvarliga och ska åtgärdas. Mindre fel och brister som finns, men som inte innefattas av kravspecifikationen, ändras om det anses nödvändigt och om tid finns.

4 Flödesstrukturen

Tanken med en flödesstruktur är att vi senare ska kunna skapa en enhetlig presentation av de utvalda flödena i applikationsmodellen. Dessa utvalda flöden måste då bygga på en gemensam struktur för att en genomförbar metod för implementering ska kunna hittas. Flödesstrukturen är alltså ett abstrakt begrepp för det bakomliggande skelettet som alla flöden är uppbyggda av. Denna struktur syns inte i själva applikationsmodellen utan används enbart för att vi ska kunna finna en bra implementeringsmetod.

Arbetet med flödesstrukturen går ut på att, utifrån de nio flöden vi studerat närmare, välja ut de som kan anses som lämpliga representanter att basera strukturen på. Dessa representativa flöden måste sedan analyseras så att vi kan finna någon form av minsta gemensamma nämnare dem emellan, en nämnare som sedan ger oss den struktur vi är ute efter.

4.1 Val av representativa flöden att basera strukturen på

När vi väljer ut de representativa flödena avser vi att ligga på en sådan nivå att vi kan visa på hur flera olika typer av flöden kan presenteras. Vår teori är att man sedan, genom de representativa flödenas strukturer, även ska kunna använda andra flöden med olika egenskaper och grad av komplexitet⁵. Detta genom att man lägger till eller ta bort de komponenter som vi visar finns att tillgå i flödesstrukturen.

Vi gör vårt representativa urval utifrån tre huvudkriterier:

- **Olika typer**
Applikationsmodellen ska kunna presentera både administrativa, tekniska och vardagliga flöden, vilket innebär att även strukturen måste klara av att hantera dessa. Detta medför att vi måste se till så att vi väljer ut flöden från samtliga tre huvudgrupper som togs upp under kapitel 3.6 – Företagsflöden hos Informator.
- **Olika utseende**
Applikationsmodellen ska även kunna presentera både linjära och icke-linjära flöden, d.v.s. flöden där man har inga (eller ett fåtal) respektive många valmöjligheter. I många konfigureringsflöden, t.ex. när man bygger en dator, har man en del moment som måste utföras, men utan

⁵ Komplexiteten definieras som en kombination av ingående komponenter sett till både typ och antal.

någon egentlig turordning dem emellan. Exempelvis måste man sätta in både ljudkort, grafikkort och hårddisk (se figur 3.5). Så länge man utför samtliga moment innan man startar datorn så spelar ordningen dem emellan ingen roll, och alla sätt är lika korrekta. I andra situationer, som t.ex. certifiering, måste alla steg däremot göras i en viss ordning. Slutsatsen av detta är att både linjära och icke-linjära flöden måste finnas med i vårt representativa urval, för att vi ska kunna införa och testa denna valfrihetsegenskap i vår struktur.

- **Olika grad av komplexitet**

Detta är det viktigaste kriteriet i urvalet. Förutom att kunna hantera olika typer av flöden, så måste flödesstrukturen också kunna hantera flöden som har olika många komponenter och olika kombinationer av komponenter, oavsett om dessa flöden tillhör samma typ eller har samma utseende. Utgår vi enbart från flöden av samma komplexa karaktär är risken att inte visar på att denna verkligen kan hantera flöden som har olika innehåll.

För att kunna välja ut representativa flöden börjar vi med att titta på samtliga flöden som vi hittat på Informator, och gruppera dessa efter typsort (administration, teknik, vardag). Inom respektive grupp delar vi sedan in flödena i delgrupper beroende på om de linjära eller icke-linjära. Slutligen graderar vi varje flöde med avseende på hur pass komplext det är (skala 1-5). Flöden som slutligen hamnar inom samma grupp och har samma grad av komplexitet anses som likartade.

Indelningen av flöden, enligt de kriterier som vi satt upp ovan, medför att vi kan välja ut representativa flöden som är principiellt olika. Ur varje grupp av likartade flöden väljer vi ut ett eller ett par som får bli våra representanter. Genom att vi väljer flöden på detta vis är vår förhoppning att vi implementerar en sådan mångfald, att sannolikheten ökar för att vår struktur verkligen fungerar för flera olika flöden.

Nedan är de flöden som vi väljer ut som representanter. Valen motiveras och graderar vi med avseende på de tre huvudkriterierna. Flödena illustreras i bilaga 3.

- **Certifiering**

Ett flöde av denna typ är väldigt linjärt utan några valmöjligheter. Stegen måste utföras i rätt ordning då detta handlar om ett regelstyrt testförfarande. Eftersom det är linjärt och utan valmöjlighet så är

komplexiteten låg, varvid det inte krävs så mycket av strukturen för att hantera denna typ av flöde. Flödet får därmed fungera som vårt minsta exempel, utifrån vilket vi sedan bygger på med mer avancerade.

Typ	Utseende	Komplexitet
Administrativt	Linjärt	1-2

Figur 4.1 – Gradering för flödet ”Certifiering”.

- **Skicka fax**

Precis som certifieringsflödet så är detta flöde enkelt och linjärt i sin uppbyggnad. Komplexiteten höjs dock något eftersom flödet trots allt kräver olika komponenter och innefattar en hel del steg; sätt i papper, slå mottagarnummer, vänta på signal o.s.v. Flödet får representera ett vardagligt flöde som är linjärt och mindre komplext.

Typ	Utseende	Komplexitet
Vardagligt	Linjärt	2-3

Figur 4.2 – Gradering för flödet ”Skicka fax”.

- **Datorkonstruktion**

För att kunna hantera detta flöde så måste strukturen innefatta en situation där flera olika moment ska genomföras, men där ordningen mellan dem inte spelar någon roll. P.g.a. dessa oberoende valmöjligheter så är flödet icke-linjärt, vilket i sin tur leder till att flödet kommer att se annorlunda ut och ha olika komponentegenskaper. Detta tekniska flöde är både icke-linjärt och komplext, och ingår därmed i vårt representativa urval.

Liknande tekniska flöden med oberoende val finns i många former, och det kan då likaväl handla om att sätta upp nätverk.

Typ	Utseende	Komplexitet
Tekniskt	Icke-linjärt	4

Figur 4.3 – Gradering för flödet ”Datorkonstruktion”.

- **Kursadministration**

Flödet går ut på att boka in rätt lärare till rätt kurs, och är ganska avancerat, men samtidigt också väldigt konkret i sin struktur. Flödet innehåller ändå ett antal steg och två vägvalsmöjligheter, vilket gör att graden av komplexitet är relativt hög. Att flödet är linjärt och komplext gör det till ett intressant val att ta med, för att se om det går att använda i vår struktur. Andra administrativa och centrala händelser inom ett företag, såsom inköp och försäljning, kan jämföras med detta flöde.

Typ	Utseende	Komplexitet
Administrativt	Linjärt	4

Figur 4.4 – Gradering för flödet ”Kursadministration”.

- **Affärsadministration**

Detta administrativa flöde är ganska omfattande och komplext i sin uppbyggnad, med många ingående moment, som var för sig kräver olika komponenter. Att flödet dessutom är icke-linjärt med de ingående förgreningarna, höjer komplexiteten ytterligare. Detta gör det sammantaget till en bra representant för gruppen av icke-linjära och komplexa flöden, och tas med för att visa på strukturens fulla potential.

Typ	Utseende	Komplexitet
Administrativt	Icke-linjärt	5

Figur 4.5 – Gradering för flödet ”Affärsadministration”.

Flödena ”Ringa ett samtal” och ”Kopiering” tas inte med i vårt urval eftersom de är snarlika de representativa flödena sett till både utseende och komplexitet. Detsamma gäller flödet ”Koka kaffe” då det inte finns några större skillnader mellan detta och ”Datorkonstruktion”, när det gäller utseende och komplexitet, som motiverar att vårt urval ska utökas.

När det gäller ”Abalonflödet” är det ett flöde som hade varit bra att kunna ta med, främst med tanke på dess komplexitet. Tyvärr är dock komplexiteten så pass hög att bara ett fåtal anställda klarar av och förstår det, och när dessa anställda har försökt visa oss hur det fungerar, har de stött på stora problem. Vi har dessutom inte själva möjligheten att studera programmet närmare på

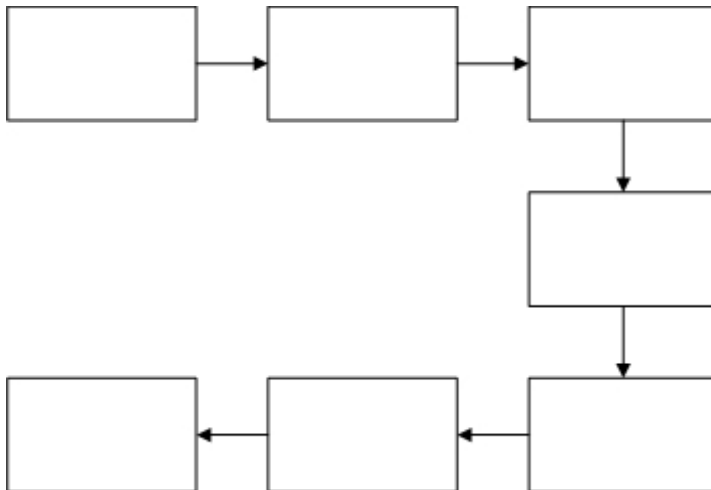
egen hand. Detta sammantaget gör att vi har svårt att se något lämpligt flöde i programmet, varvid vi väljer att inte ta med ”Abalonflödet” i vårt urval.

4.2 Principiellt olika lösningsförslag för flödesstrukturen

Det finns en stor mängd lösningar till problemet med att skapa en struktur som är anpassad för olika typer av flöden. Allt från linjära lösningar med mycket få valmöjligheter, till helt icke-linjära som i stort sett är obegränsade, kan tänkas. För att få en mer konkret bild över vad vi har att välja mellan har vi valt att sammanställa, skissera upp, och i ord beskriva de abstrakta lösningsförslagen vi kommit på. Respektive lösnings för- och nackdelar vägs mot varandra, vilket leder fram till en utvärdering och en uppställning av den slutgiltiga flödesstrukturen.

- **Strikt linjär flödesstruktur**

Denna lösning (se figur 4.6) baseras på en metod där man går från punkt A till punkt B, utan några direkta valmöjligheter. Detta kan jämföras med t.ex. ett löpandeband-flöde där allt sker i tur och ordning och på samma sätt varje gång. På så vis så är denna struktur den minst dynamiska med tanke på sitt strikt linjära upplägg och kan enbart användas för flöden såsom ”Certifiering”.



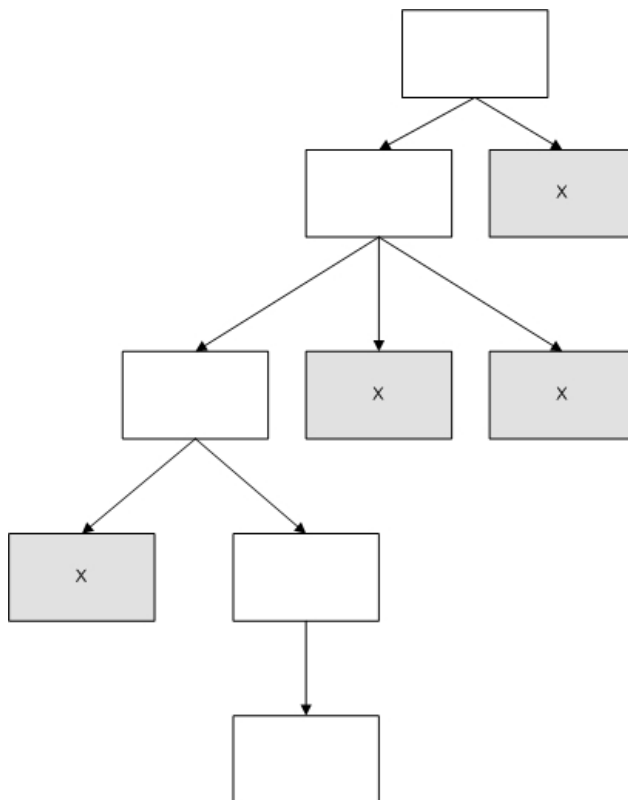
Figur 4.6 – Strikt linjär flödesstruktur.

Fördelar: Enkelt att implementera strukturens komponentegenskaper då endast rätt eller fel existerar.

Nackdelar: Strukturen är inte anpassad för alla typer av flöden då inga former av valmöjligheter existerar.

- **Linjär flödesstruktur med fiktiva vägval**

En utbyggnad av det strikt linjära flödet, där man nu kan använda sig av valmöjligheter, i form av ett vägval för t.ex. en ja- eller nejsituation. Vägvalen är dock helt fiktiva eftersom fortfarande endast ett val är korrekt och leder vidare i flödet. Stora likheter finns mellan den strikt linjära varianten och denna struktur, skillnaden är att det nu krävs en ny komponentegenskap för att skapa valmöjligheter. Strukturen kan enbart användas för helt linjära flöden såsom "Certifiering".



Figur 4.7 – Linjär flödesstruktur med fiktiva vägval.

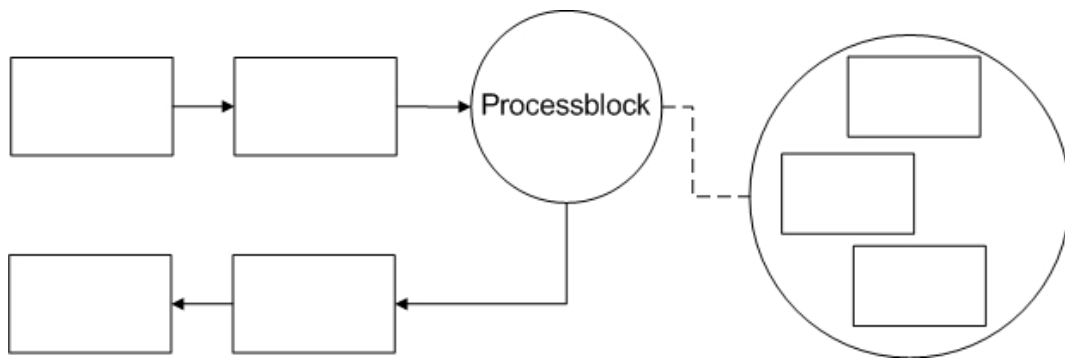
Fördelar: Relativt enkelt att implementera strukturens komponentegenskaper. Bra struktur för ett företagsflöde som måste följas till 100 % (jfr. kärnkraftverk), men som ändå ska innehålla ställningstagande.

Nackdelar: Strukturen är inte anpassad för alla typer av flöden då inga riktiga valmöjligheter tillåts.

- **Halvlinjär flödesstruktur med processblock**

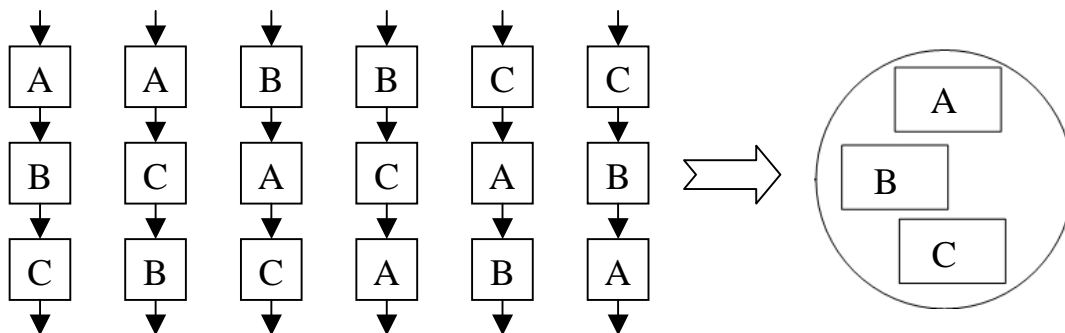
Här implementeras en större valfrihet jämfört med de ovanstående lösningarna, genom att vi inför s.k. processblocks-komponenter.

Processblocket har en egenskap som innebär att ett flöde har en grupp av komponenter som ska ligga på en viss plats, men där ordning dem emellan inte spelar någon roll (se figur 4.8). Här finns då ingen enhetlig "lösning" för ett flöde, utan en del av de ingående komponenternas placeringar kan variera. Lämpar sig för flöden såsom "Datorkonstruktion".



Figur 4.8 – Halvlinjär flödesstruktur med processblock.

Fördelar: Innehåller ett valfrihetsmoment från vilket man kommer framåt i strukturen oberoende av ordningen mellan dess ingående komponenter. Kan användas för mer icke-linjära flöden där man då kan sammanfatta förgreningar mellan flera komponenter, i en och samma komponent (se figur 4.9).

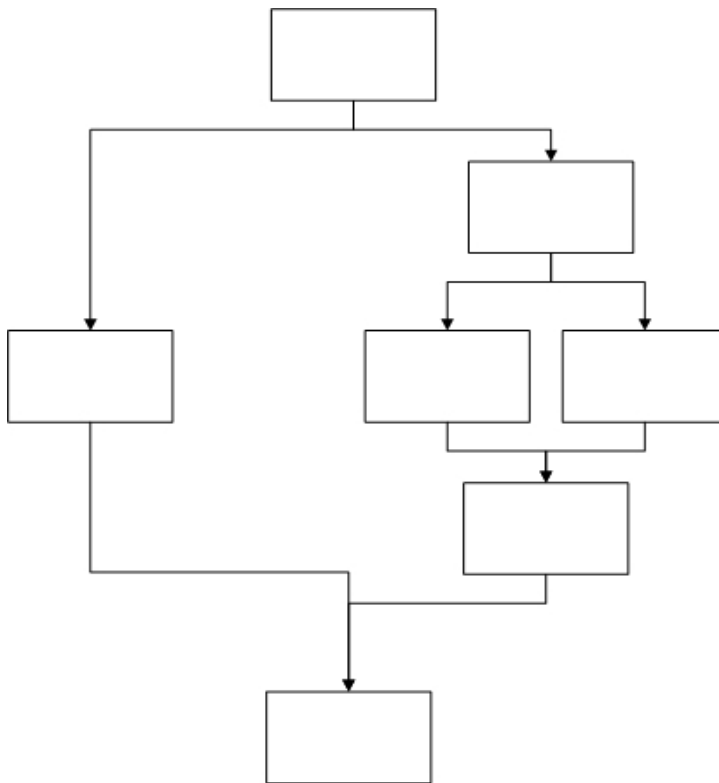


Figur 4.9 – Fördelen med processblocket; förgreningar mellan flera komponenter sammanfattas i en och samma komponent.

Nackdelar: I grundformen av denna struktur finns inga valmöjligheter i form av vägval, vilket skapar en begränsning för flöden som kräver detta.

- **Halvlinjär flödesstruktur med vägval**

Detta skiljer sig från ett icke-linjärt flöde genom tillfogandet av faktiska vägval. Strukturen är konstruerad för vägval som var för sig är oberoende, men som alla ändå slutar i samma mål. Detta innebär att strukturen kan hantera sådana flöden som kräver vägval, som i sin tur leder till förgreningar med andra valmöjligheter, men som ändå ska resultera i samma sak. Eftersom strukturens vägar leder mot samma mål så är den inte en helt icke-linjär lösning. Kan användas för både helt linjära flöden och sådana där vägval måste vara möjliga, t.ex. ”Skicka fax”, ”Kursadministration” och ”Affärsadministration”.



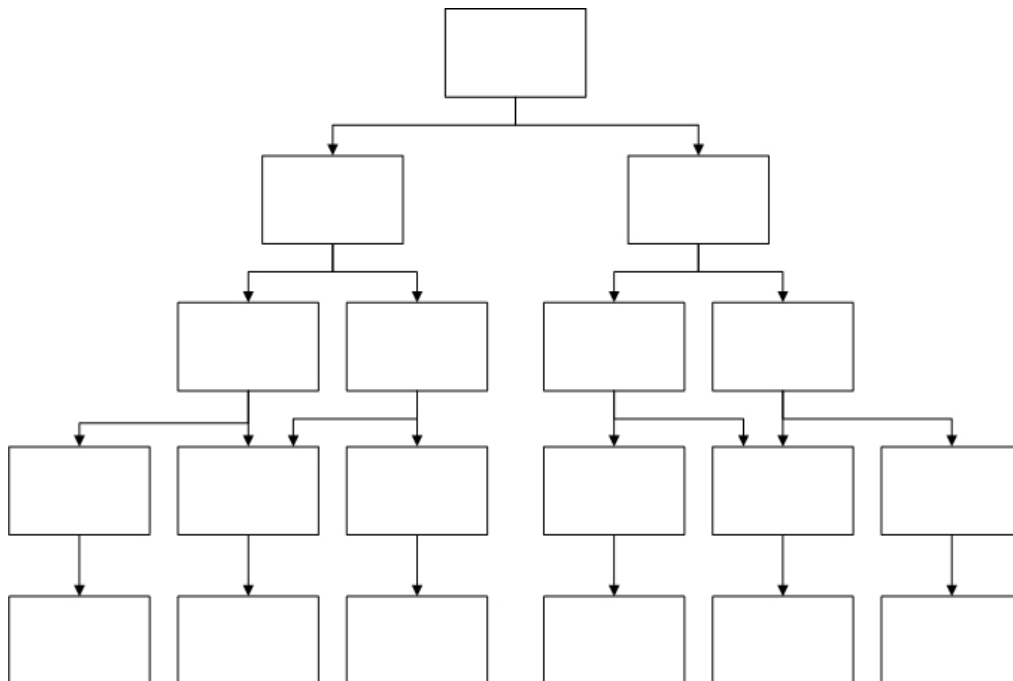
Figur 4.10 – Halvlinjär flödesstruktur med vägval.

Fördelar: Strukturen innehåller faktiska valmöjligheter. Strukturen kan hantera flöden innehållandes vägval och oberoende förgreningar, förgreningar som sedan leder mot samma mål. Strukturen är mer dynamisk och kan användas för olika typer av flöden, både linjära och sådana som kräver ett (eller flera) vägval.

Nackdelar: Svår struktur att använda när det krävs flera olika valmöjligheter. Hur ska man på ett enkelt sätt hantera t.ex. processblock med grupper av komponenter som ska ligga på en viss plats, men där ordning dem emellan inte spelar någon roll?

- **Trädstruktur (icke-linjär flödesstruktur)**

En icke-linjär lösning där de olika förgreningarna i flödet leder till olika slut. Varje komponent fungerar i sig som ett vägval där man kommer till olika ställen beroende på vad man valt tidigare. Detta kan jämföras med en webbplats där man, via länkar, kan navigera mellan olika sidor i en obestämd ordning. Varje väg är möjlig och målet i flödet är helt beroende av den väg som valts tidigare, som i sin tur är beroende av valet innan det o.s.v. Vägen *och* målet ändras alltså beroende på vägvalet! Strukturen är helt dynamisk och kan anpassas för alla våra flöden genom att förgreningar läggs till eller tas bort. Flödet "Skicka fax" kräver endast en gren, medan "Datorkonstruktion" eller "Affärsadministration" kräver betydligt fler.



Figur 4.11 – Flöde uppbyggt med en trädstruktur.

Fördelar: Dynamisk struktur. Sett till storleken så finns det inga direkta begränsningar för vilka flöden som skulle kunna hanteras av denna struktur.

Nackdelar: Varje komponent kräver sina egna vägar och mål, vilket medför att strukturen snabbt växer i takt med att fler komponenter tillfogas. Strukturen har även samma nackdel som den halvlinjära flödesstrukturen med vägval; hur ska man på ett enkelt sätt hantera t.ex. processblock med grupper av komponenter som ska ligga på en viss plats, men där ordning dem emellan inte spelar någon roll?

Sammantaget så krävs det nästan att man redan i förväg sätter upp en gräns för antalet komponenter och vägar som får ingå. Lösningen tappar då sin funktion, eftersom man begränsar en struktur som från början är framtagen för att vara obegränsad.

4.2.1 Utvärdering och val av lösningsförslag

Grundtanken med den applikationsmodell som ska tas fram är att man ska kunna presentera olika typer av flöden på ett enhetligt sätt. En struktur som inte klarar av en mångfald av flöden och egenskaper kommer inte att kunna användas för våra representativa flöden, eftersom dessa är av olika typer med sina individuella strukturer och innehåll. Sett till de fem representativa flöden så finner vi tre konkreta krav som ett lösningsförslag måste kunna uppfylla och som vi måste ta med som riktlinjer när vi väljer ett lösningsförslag:

- **En startpunkt och en slutpunkt**

Samtliga representativa flödena har klart uppställda start- och slutpunkter som definierar vad som ska göras och vilket resultatet är. Strukturen måste därför kunna användas för flöden som har en given start- och slutpunkt. Startpunkten ska svara på frågan ”vad är det som ska uppnås i detta flöde?”. Slutpunkten ska i sin tur visa på det tilltänkta resultatet av flödet.

- **Oberoende valmöjligheter**

Flödet ”Datorkonstruktion” (kunde lika gärna ha handlat om hur man bygger ett kontorsnätverk) visar tydligt att det måste finnas möjligheter för vissa flödeskomponenter att ligga i oberoende ordning, annars är risken att situationen i figur 4.11 uppkommer. Strukturen måste därför kunna hantera ett eller flera processblock för att inte varje steg i flödet ska bli helt beroende av de tidigare, och därmed riskera en trädstruktur.

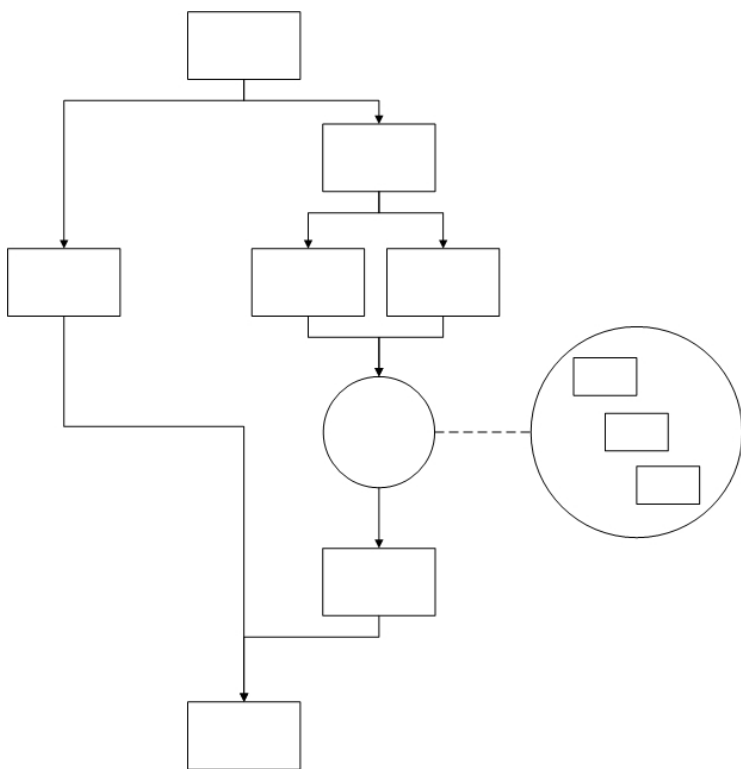
- **Olika förgreningar**

Strukturen måste dessutom kunna hantera att ett flöde har olika förgreningar, vilka var för sig bildar olika oberoende vägar från start till mål. Mer komplexa flöden såsom ”Affärsadministration” består av valmöjligheter där ett val leder till olika oberoende vägar, men som i slutändan ändå leder fram till samma resultat.

Eftersom inget av de upptagna lösningsförslagen helt passar in på de parametrar vi anger ovan, krävs det att vi hittar en annan och mer flexibel variant. Vi har därför kommit fram till att vi måste kombinera ett helt linjärt med ett helt icke-linjärt flöde, och använda nödvändiga komponenter från de olika lösningsförslagen. Det vi söker är en kombination mellan ”Halvlinjär

flödesstruktur med vägval” och ” Halvlinjär flödesstruktur med processblock”, som skulle ge en lämplig lösning där vi använder processblocken för att skapa en mer dynamisk variant av den halvlinjära strukturen med vägval. Detta innebär att vi då kan påvisa samtliga de nyckelmoment som hittats i de representativa flödena; start, slut, vanliga processer, processblock och olika vägval.

Kombinationslösningen, även kallad ”*Halvlinjär struktur med vägval och processblock*” (se figur 4.12), blir i grund och botten en form av icke-linjär struktur, innehållande valfritt antal oberoende vägar, men som ändå blir linjär genom en given start- och slutpunkt. Trots att den innehåller valfrihetsmoment blir den kortare till utseendet än en trädstruktur, genom att vi inför processblocken med sina oberoende valmöjligheter. Resultatet är att strukturen kan användas för de flöden som har en start- och en slutpunkt, olika förgreningar och moment där flera komponenter ska sättas på plats oberoende av varandra.



Figur 4.12 – Halvlinjär struktur med vägval och processblock. Vår kombinationslösning med de komponenter som ska ingå.

Valet av lösningsförslag, som visas i figur 4.12, gör det nu möjligt att beskriva och strukturera samtliga våra utvalda flöden på ett enhetligt sätt. Därmed skapar vi nu också förutsättningen för att applikationsmodellen sedan ska kunna presentera olika typer av flöden på ett enhetligt sätt.

Om de övriga lösningsförslagen kan nämnas att många inte räcker till för att visa det som de utvalda flödena kräver, de som är helt linjära kan t.ex. omöjligt visa de nödvändiga valmöjligheterna som ibland måste finnas i ett flöde. Flöden såsom ”Strikt linjär flödesstruktur” eller ”Linjär flödesstruktur med fiktiva vägval” faller ganska snabbt bort p.g.a. avsaknaden av valfrihetsmoment.

När det gäller den helt icke-linjära lösningen (trädstrukturen) så räcker den visserligen till för att visa alla utvalda flöden. Den stora nackdelen är dock att denna struktur växer exponentiellt i takt med fler valmöjligheter som tillfogas. Detta problem ser vi tydligt efter att ha studerat en mängd flödesexempel på olika företags webbplatser. Flöden innehållandes valmöjligheter tenderar snabbt att svälla och bli stora och komplexa. Detta resulterar nästan alltid i stora trädstrukturer där varje steg i flödet blir helt beroende av det tidigare steget. Ur ett företags- och programmeringsmässigt perspektiv så är denna lösning med en beroendekedja inte att föredra som bas för en enhetlig flödespresentation. Ett enda extra steg i strukturen skulle nämligen innebära att ett mycket stort antal nya vägval måste tillfogas, något som torde bli både tidsödande och dyrt.

4.3 Regler för hur strukturen ska se ut och fungera

Efter att vi nu valt en lösning som bygger på en kombination mellan ”Halvlinjär flödesstruktur med vägval” och ”Halvlinjär flödesstruktur med processblock”, behöver vi också definiera några regler för hur denna lösning och dess ingående komponenter ska fungera.

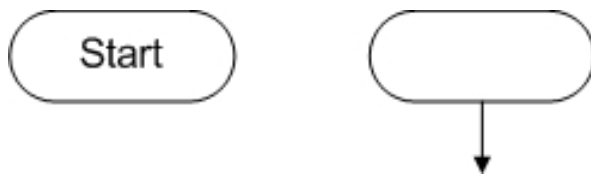
4.3.1 Allmänna regler

- Start och slut på flödena ska finnas, som visar att man startar med att göra något, och slutar när man kommer till ett visst mål. Ungefär; ”detta har du givet, och hit ska du komma”. Start och slut ska markeras med speciella symboler i strukturen för att påvisa att dessa inte är delar i själva flödet.
- Förutom start och slut ska komponenterna process, processblock och vägval kunna hanteras av strukturen. Beskrivning av dessa komponenter följer nedan.

4.3.2 Komponentregler

- **Start**
Ingen väg in, en väg ut (som leder till första delen i flödet). Denna komponent har inget med själva flödet att göra, utan existerar endast för

att definiera vad som ska uppnås. Komponentens symbol har formen av en avrundad rektangel i flödesstrukturen.



Figur 4.13 – Symbol för startkomponent.

- **Slut**

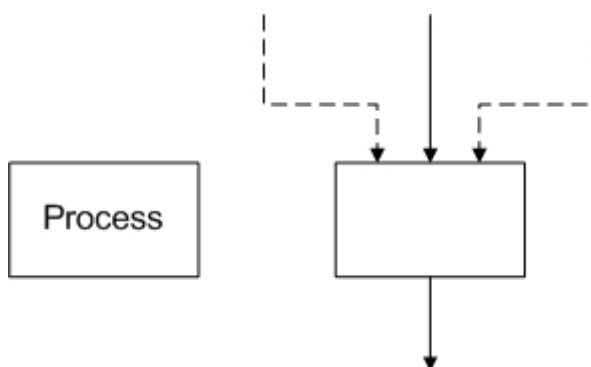
En väg in (som kommer från den sista delen i flödet), ingen väg ut. Denna komponent har inget med själva flödet att göra, utan existerar endast för att definiera dess resultat. Komponentens symbol har formen av en avrundad rektangel i flödesstrukturen.



Figur 4.14 – Symbol för slutkomponent.

- **Process**

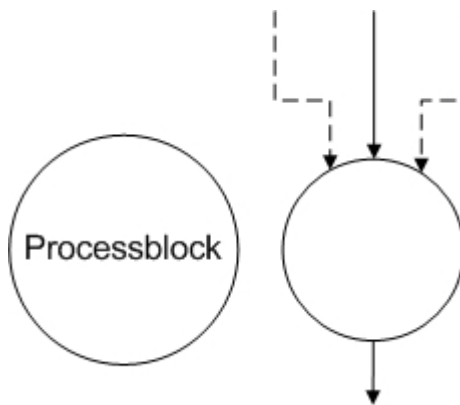
Minst en väg in, endast en väg ut. Endast en väg ut eftersom den är enkel i sin uppbyggnad och inga valmöjligheter tillåts. Flera vägar kan leda till en och samma processkomponent, eftersom olika valmöjligheter kan leda till samma flödessituation. Komponentens symbol har formen av en rektangel i flödesstrukturen.



Figur 4.15 – Symbol för processkomponent.

- **Processblock**

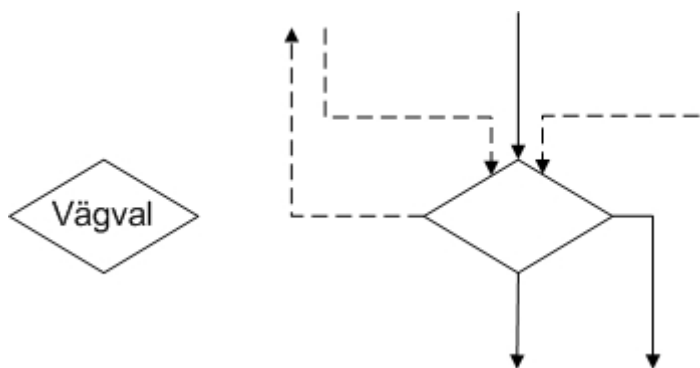
Minst en väg in, endast en väg ut. Samma regler för in- och utvägar gäller som för den vanliga processkomponenten. Skillnaden ligger i att processblocket består av många olika delprocesser i sig, som ska sammanfogas och bilda en helhet (själva blocket). Dessa ingående delprocesser kan då vara placerade i en oberoende ordning inom blocket. Syftet är att korta ned valmöjligheter som skulle leda till en trädstruktur med många förgreningar. Komponentens symbol har formen av en cirkel i flödesstrukturen.



Figur 4.16 – Symbol för processblocks-komponent.

- **Vägval**

Minst en väg in, minst två vägar ut. Har som syfte att presentera ett vägval i flödet, antingen framåt eller bakåt. Komponentens symbol har formen av en diamant i flödesstrukturen.



Figur 4.17 – Symbol för vägvalskomponent.

4.4 Genomgång av lösningen för flödesstrukturen

Syftet med genomgången är att kontrollera ifall lösningsstrukturen kan hantera linjära och icke-linjära flödesexempel, liksom flöden som är mer eller mindre komplexa. Samtidigt kontrolleras ifall strukturen kan hantera de olika typer av flöden som vi valt ut, d.v.s. både administrativa, tekniska och vardagliga flöden. För att strukturen, och dess ingående komponenter, ska ses som godkänd ska den kunna hantera samtliga representativa flöden.

Strukturen ska fungera både för enklare och mer komplexa flöden, varvid vi väljer att konstruera vår lösning stegvis utifrån flöden av olika komplexitetsgrad. Lösningsstrukturen skapas med utgångspunkt i certifieringsflödet som ligger på nedre gränsen på skalan för komplexitet. Efterhand byggs den sedan på med komponenterna från de andra representativa flödena, varpå de framtagna reglerna appliceras på strukturen allt eftersom. Tanken med detta är att, precis som när det gäller testningsförfarandet, starta i liten skala och sedan lägga på steg för steg.

Första steget är att rangordna våra representativa flöden med avseende på hur komplexa de är; certifiering, skicka fax, kursadministration, bygga dator, affärsflöde. Därefter börjar vi med att se om det minst komplexa flödet (certifiering) kan struktureras enbart utifrån våra komponenter och deras respektive regler. Detta görs genom att vi ritar upp flödet på papper, helt efter de regler och definitioner vi har satt upp för de olika komponenterna, och sedan stegar igenom det för att se om vi logiskt kan gå från start till mål. Om detta är möjligt går vi vidare och gör om samma procedur med nästa flöden, som då är lite mer komplext.

För att se hur ett mycket komplext exempel skulle se ut, skisserar vi upp ett fiktivt flöde som består av de gemensamma nämnarna mellan de representativa flödena. Avslutningsvis stegas även det fiktiva, komplexa, flödet igenom från start till mål. Detta kan ses som en form av regressionsgenomgång där vi kontrollerar så att samtliga komponenter fungerar ihop, för att på så sätt försäkra oss om att det även skulle vara möjligt att illustrera större och mer komplexa flöden än våra representativa.

4.4.1 Utvärdering av genomgången

Genomgången visar att de representativa flöden vi arbetar utifrån går att strukturera och följa logiskt med hjälp av vår lösningsstruktur. Genom att lägga till eller ta bort de tillgängliga komponenterna så kan samtliga flöden stegas igenom från start till mål. Lösningen som sådan får därmed ses som godkänd.

En viktig detalj som framkommit under genomgången är skillnaden mellan val och val. Efterhand som flödena strukturerats upp så har vi märkt att vägvalskomponent har olika egenskaper gällande sina utvägar. Några går bakåt i flödet, medan vissa förgrenade sig och tar olika vägar, fast enbart framåt i flödet. Efter diskussion har vi kommit fram till att detta är väsentligt för kommande arbete och att vi därför måste skilja dem åt på något sätt.

Under genomgången har vi också insett att vissa processer som ingår i ett flöde även skulle kunna representera flöden i sig. Man skulle kunna säga att vissa processkomponenterna fungerar som delflöden till det huvudsakliga flödet som man visar. Ett bra exempel är administrationsflödet, som kan ses som ett delflöde till affärsflödet. Även om det i grund och botten handlar om vilken detaljnivå man lägger sig på när man försöker illustrera flödet, så anser vi ändå att det vore bra att kunna representera delflöden i vår applikationsmodell på något sätt.

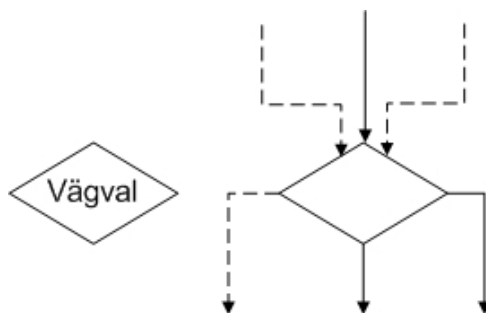
En sista reflektion av genomgången är att symboliken för processblocken inte blir tydlig nog. I princip så är processblocket en vanlig processkomponent, som i sig består av andra processkomponenter. Vi anser därför att det är bättre att cirkelsymbolen för processblocket ändras så att man kan associera den med processkomponentens funktionalitet.

4.5 Förändringar och förbättringar utifrån genomgången

En första förbättring är att förtydliga skillnaden mellan de olika valen. Vi väljer att dela upp den tidigare vägvalskomponenten i två olika komponenter; iterationsval respektive vägval.

- **Vägval**

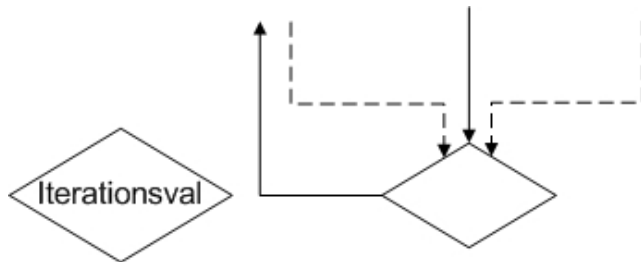
Minst en väg in, och minst två vägar ut. Har som syfte att presentera framåtskridande vägval för användaren. Komponentens symbol har formen av en diamant, med pilar som går framåt.



Figur 4.18 – Ny symbol för vägvalskomponent.

- **Iterationsval**

Minst en väg in, och en väg ut. Har som syfte att presentera iterativa vägval för användaren, dvs. sådana val som i någon form måste upprepas genom att man stegar tillbaka i flödet tills det att något villkor är uppfyllt. Komponentens symbol har formen av en diamant, med en eller flera pilar som går bakåt.

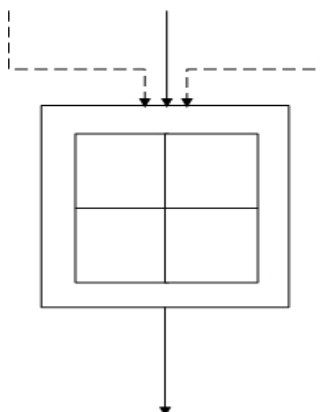


Figur 4.19 – Ny symbol för iterationsvals-komponent.

Symboliken för processblocket ändrar vi för att man ska kunna förstå att denna och den vanliga processkomponenten har något gemensamt. Båda två har samma egenskaper vad gällande in- och utvägar. Skillnaden är att processblocket är en processkomponent uppdelad i flera oberoende delar.

- **Processblock**

Minst en väg in, endast en väg ut. Samma regler för in- och utvägar gäller som för den vanliga processkomponenten. Skiljer sig genom att det här är många olika delprocesser som ska sammanfogas och bilda en helhet. Dessa delprocesser kan då läggas i en ordning som är oberoende av varandra. Syftet är att korta ned många valmöjligheter, som annars skulle leda till en trädstruktur med många förgreningar. Komponentens symbol har formen av en rektangel innehållandes mindre rektanglar.

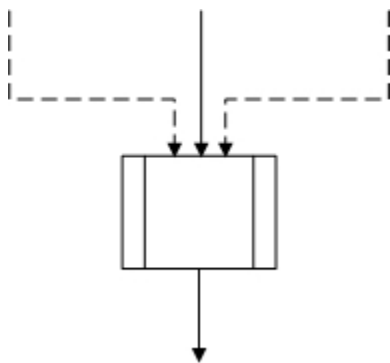


Figur 4.20 – Ny symbol för processblocks-komponent.

Eftersom vi insett att det kan finnas möjliga delflöden i ett huvudflöde väljer vi att skapa ytterligare en komponent. Denna komponent har precis samma egenskaper som en process, men ritas lite annorlunda bara för att markera att den innehåller ett underliggande flöde.

- **Delflöde**

Minst en väg in, endast en väg ut. Komponenten markerar att den i sig innehåller ett eget, självständigt, flöde (delflöde). Flera vägar kan leda till en och samma processkomponent, eftersom olika valmöjligheter kan leda till samma flödessituation. Komponentens symbol har formen av en rektangel innehållandes en rektangel.



Figur 4.21 – Symbol för delflödeskomponent.

4.6 Slutgiltig flödesstruktur

Genomgången av strukturen har visat på att enbart några mindre förändringar har behövt göras av den ursprungliga lösningen. Några komponenter har behövt läggas till, andra har delats upp eller förtydligats.

Figur 4.22 innehåller samtliga komponenter som krävs för att visa våra fem utvalda flöden, vilka representerar grupper av flöden med olika karaktär. Figuren visar också på de olika komponenternas logiska kopplingar, och är ett exempel på hur ett avancerat flöde skulle kunna vara uppbyggt.

För det fortsatta utvecklingsarbete är det av största vikt att alla parter verkligen förstår vad den framtagna flödesstrukturen är för något, och vad denna har med arbetet med applikationsmodellen att göra. Detta är anledningen till att vi väljer att göra en kort presentation (se bilaga 4) av vårt lösningsförslag för Henrik Thorsell. Presentationen har tre konkreta syften:

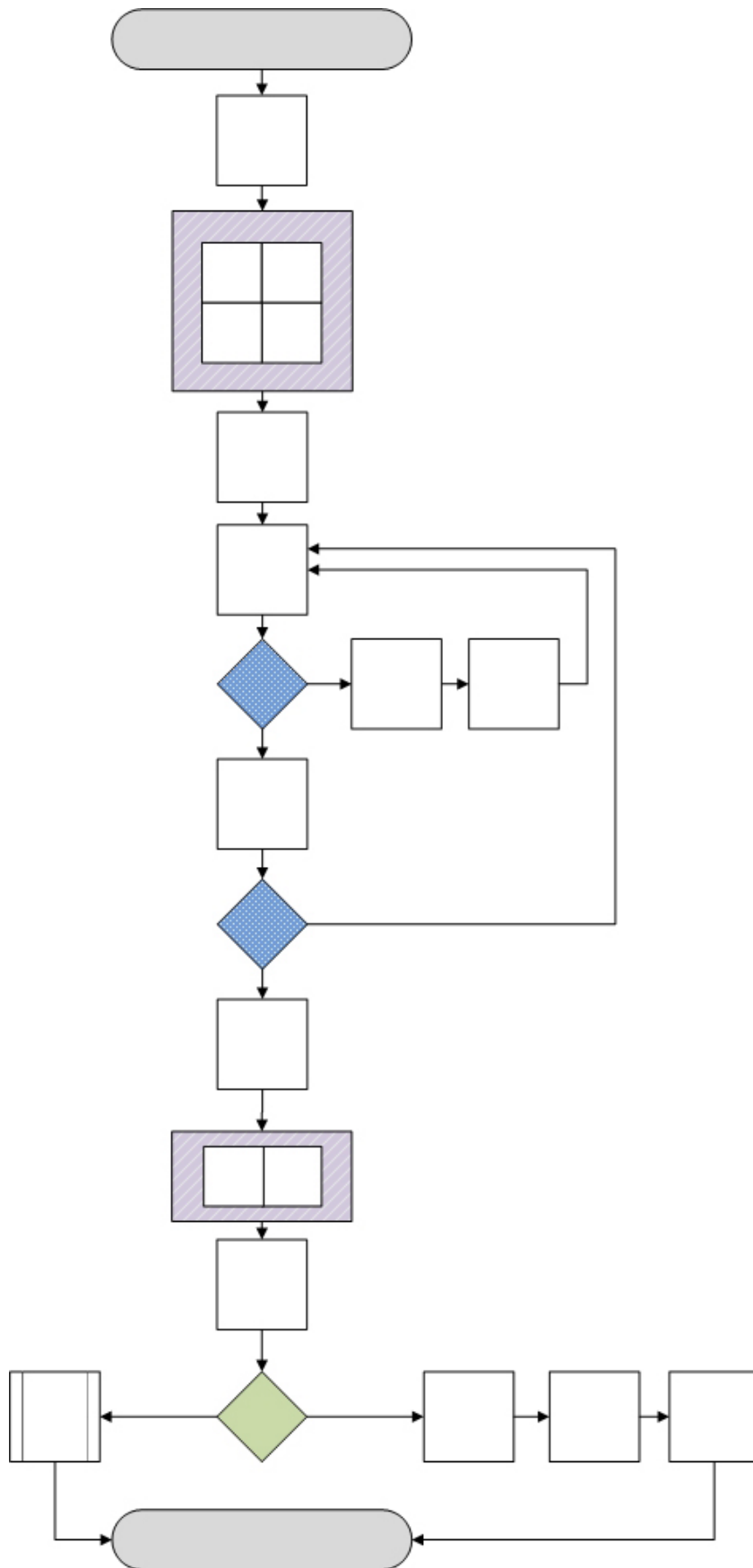
- Visa på skillnaden mellan flödestruktur och flödespresentation

- Visa flödesstrukturens komponenter, hur dessa ser ut och fungerar
- Visa på möjligheterna med flödesstrukturen och komponenterna

I detta skede presenteras inte något som har med applikationsmodellen att göra. Däremot vill vi visa och förklarar innebörden av strukturen och de bakomliggande komponenterna, eftersom det är det viktigt att alla förstår hur dessa ser ut, vad de har för syfte och hur de fungerar rent tekniskt.

Både vi och Henrik Thorsell är överens att man inte med säkerhet kan veta ifall alla sorters flöden kan visas med vår struktur. Detta är tanken, men vi kan inte veta eftersom vi inte kan testa alla förekomster av flöden inom ramen för vårt arbete. Ifall ett annorlunda flöde upptäcks och det visar sig att detta inte kan skapas utifrån vår struktur, så har man två vägar att gå; undersöka om flödet i sig är ologiskt uppbyggt och utifrån detta besluta att inte ta med det (om man inte förändrar det). Alternativt får man se om strukturen kan anpassas ytterligare så att även det nya flödet kan skissas upp.

I en förlängning där applikationsmodellen ska byggas ut med andra företagsflöden, kan man tänka sig att strukturen kan användas för att skickas ut till företag som vill lägga in sina flöden. De som ska administrera applikationen och lägga in nya flöden kan då säga till kunden; ”skissera upp era flöden enligt denna flödesstruktur, så kan vi skapa en presentation av dessa i applikationen”. Med andra ord så kan flödesstrukturen fungera som en form av mall för de flöden som man sedan vill ta med i applikationen.



Figur 4.22 – Skiss på det slutgiltiga lösningsförslaget, med samtliga komponenter som kan ingå.

5 Applikationsmodellen

Applikationsmodellen är den grafiska presentationen av det vi vill lära ut, d.v.s. hur och varför ett visst företagsflöde ser ut som det gör.

Applikationsmodellen ska ses som en prototyp som ska presentera en metod för hur man skulle kunna lära ut flöden för användarna, och som ska visa vilka möjligheter som finns inför en framtida framtagning av en fullständig e-learningkurs.

Vi startar med att undersöka om det finns några liknande applikationslösningar, och hur dessa i så fall ser ut och fungerar. Samtidigt väljer vi ut de flöden som ska tas med i applikationsmodellen och tittar på olika tänkbara lösningar och metaforer. När lösningen är bestämd, görs en storyboard och en walkthrough för att få en bild av hur applikationsmodellen ska se ut och fungera. Arbetet med applikationsmodellen utförs på ett iterativt sätt genom implementering, testning, förändring, ny testning o.s.v.

5.1 Val av flöden att presentera i applikationsmodellen

Av tidsmässiga skäl koncentrerar vi oss på att ta med och presentera ett mindre urval av de nio företagsflöden vi funnit på Informator, och som tidigare använts för att bygga upp flödesstrukturen. Vid urvalet har vi för avsikt att ta med flöden som använder sig av olika komponenter från flödesstrukturen, så att vi i så stor utsträckning som möjligt visar vad vi kommit fram till under arbetets gång. Eftersom applikationsmodellen ska kunna lära ut olika sorters flöden väljer vi att implementera tre stycken, som alla är olika sett till typ, utseende och komplexitet;

- Skicka fax
- Datorkonstruktion
- Utbildningsadministration

5.2 Analys av befintliga, närbesläktade, utbildningsapplikationer

Innan vi börjar skapa applikationsmodellen, väljer vi att ta en närmare titt på de utbildningsapplikationer som finns idag. Detta får fungera som en inspirationskälla, där vi kan studera vad som tidigare gjorts.

Våra försök att finna beskrivningar på likartade applikationer via de största sökmotorerna på Internet har inte gett några relevanta resultat. Troligen är det

så, utifrån vad vi själva upplever på Informator, att bara en mycket liten del av det som produceras, publiceras på Internet. Det mesta används internt på företaget och är därmed inte sökbar via sökmotorerna.

Vi utesluter inte att någon skulle kunna ha tagit fram någon form av flödessimulator som ett specifikt kunduppdrag. Någon applikation för mer generella företagsfall finns dock inte att hitta i dagsläget, och därmed tyder det mesta på att det inte finns något företag idag som saluför en applikation med samma syfte som vår applikationsmodell.

Av de resultat vi fått fram handlar det i de flesta fall om applikationer som antingen ska analysera eller förbättra ett befintligt företagsflöde. Applikationer som används för att skapa flödesscheman är också vanligt förekommande, och det har varit intressant att studera dessa just för att se hur man kan skapa och rita upp flödesscheman. I övrigt är inget av det vi funnit av större intresse för vårt arbete.

5.3 Olika tänkbara lösningsförslag för flödespresentationen

Det praktiska arbetet måste grundas på att vi har en klar idé om vad det är vi ska skapa. Vi väljer att starta med att se vilka möjliga lösningar vi har att utgå från, och sedan arbeta vidare därifrån. Som vi ser det kan flödespresentationen i applikationsmodellen baseras på ett av två övergripande lösningsförslag:

- **Linjär lösning**

En lösning baserad på ett antal på varandra efterföljande moment, ungefär som en berättelse med olika delar och handlingsförlopp. Man gör ett moment klart innan nästa startas.

Fördelen med denna lösning är att man kan portionera upp det som ska presenteras, och därmed mängden information som användaren behöver ta till sig. Dessutom medför en linjär lösning att man kan använda sig av upprepning för att återkoppla till information som presenterats tidigare och i ett annat sammanhang, vilket underlättar inläringen.

Efterföljande moment ger även en möjlighet att kontrollera om användaren har förstått det han/hon läst, genom att testa detta en stund senare när allt hunnit sätta sig.

Nackdelen är att det kan bli ostrukturerat och hackigt att presentera information allt eftersom.

- **Icke-linjär lösning**

En lösning baserad på moment där användaren går på egen upptäcktsfärd i det virtuella rummet (presentationsdelen). Det finns inget som styr i vilken turordning användaren gör saker och ting.

Den stora fördelen med denna lösning är att användaren själv kan ta sig runt i presentationsdelen och studera det som han/hon då känner för.

Nackdelen är att det kan blir svårt att få till någon bra instuderingsmetodik med återkoppling och repetition när allt presenteras samtidigt och i användarens egen ordningsföljd.

Utifrån dessa två huvudkategorier för det övergripande presentationsupplägget kan man tänka sig ett antal olika möjliga lösningar för ett enskilt moment:

- **Linjärt pussel⁶**

En momentlösning som går ut på att placera bitar i en rak ordningsföljd, exempelvis enbart från vänster till höger. Bitarna representerar delar i flödet. När man byggt ihop pusslet är flödet komplett och man har slutfört momentet.

- Variant 1: Feedback om bitar ligger rätt eller fel återges när alla bitar har placerats ut.
- Variant 2: Feedback om en bit ligger rätt eller fel återges direkt när en bit placeras ut.

- **Kvadratisk pussel**

Ett mer icke-linjärt moment där man ska placera bitar med vissa egenskaper på en kvadratisk pusselyta. Egenskaperna för bitarna styr hur de ska sättas ihop; från vänster till höger eller uppifrån och ned. När man byggt ihop pusslet är flödet komplett och man har slutfört momentet.

- Variant 1: Feedback om bitar ligger rätt eller fel återges när alla bitar har placerats ut.
- Variant 2: Feedback om en bit ligger rätt eller fel återges direkt när en bit placeras ut.

⁶ Med ordet ”pussel” menas inte den traditionella beskrivning av ett pussel med bitar, vars form antyder hur de ska sättas samman. I stället handlar det om fyrkantiga bitar eller klossar som alla är av samma form.

- **Dynamiskt konsekvenspussel**

Ett helt icke-linjärt moment där man på ett fritt sätt kopplar ihop bitarna med varandra. Här finns ingen förutbestämd pusselyta på vilken bitarna ska placeras, utan användaren kan lägga bitarna huller om buller och sedan klicka på två bitar för att koppla ihop dessa. Feedbacken består av ett bildobjekt som ändrar utseende beroende på om man gör rätt eller fel. Ett bra exempel är flödet "Bilmontering". Monteras dörrarna före karossen så får man se en animation med resultatet av det felaktiga byggandet.

- Variant 1: Först lägger man ut och kopplar ihop alla bitarna, och därefter spelas animationen upp och presenterar resultatet.
- Variant 2: När man kopplar ihop en bit med en annan får man direkt feedback, genom att bildobjektet (animationen) förändras. Detta kan liknas med en simulator där man direkt ser konsekvenserna av sitt handlande.

- **Labyrint**

Ett moment som består av att flödet ritas ut i form av en labyrint, med förgreningar och villovägar. Tanken är sedan att man som användare ska ta sig igenom flödet och svara på frågor.

- Variant 1: Användaren får först hitta vägen genom labyrinten och därefter besvara frågor som relaterar till vägen som valdes.
- Variant 2: Frågorna besvaras direkt på givna platser i labyrinten. Svarar man rätt på kommer man vidare, svarar man fel får man börja om från start igen (alternativt få en ny fråga).

- **Flödesläggare**

Ett moment där man ska komplettera flödet genom att placera ut saknade bitar i ett rutnät. Skillnaden mot ovanstående pusselvarianter är att en del av flödet nu är utritat (vägen bitarna emellan), och uppgiften är att fylla i luckorna. Vägsymbolik på bitarna ger en ledtråd om var bitarna ska placeras, och således kan man klara pusslet utan att läsa någon textinformation. När bitarna ligger på plats ska man kunna följa vägen från start till mål. Bildas ingen hel väg från start till mål med de bitar man har till förfogande, så har man gjort fel.

- Variant 1: Feedback om bitar ligger rätt eller fel återges när alla bitar har placerats ut och man ser den fullständiga vägen.

- Variant 2: Feedback om en bit ligger rätt eller fel återges direkt när en bit placeras ut.

Till varje lösningsförslag finns även ett par underställda frågeställningar för varje moment, vilka vi väljer att ta ställning till först efter att vi valt ett lösningsförslag, t.ex:

- Ska flödesvägen vara synlig från start eller inte?
- Ska flödesdelarna läggas eller inte?

5.3.1 Utvärdering och val av lösningsförslag för flödespresentation

Med tanke på mängden information vi vill förmedla, väljer vi en linjär lösning för hela flödespresentationen. Då det ibland handlar om flöden med komplext innehåll anser vi att det är bättre att dela upp informationen på olika moment, som gör att användaren lättare kan ta till sig denna. Med en linjär lösning kan man dessutom styra vilket som är nästa moment och hur användaren ska gå framåt i applikationsmodellen, vilket inte går att göra med en icke-linjär lösning.

Risken att användaren gör misstag minskar också när vi strukturerar det på ovanstående sätt, vilket är viktigt att tänka på eftersom de ska hantera applikationsmodellen på egen hand. Faktainnehållet i applikationsmodellen är ganska komplext, varvid vi alltså anser det nödvändigt att försöka presentera detta på ett lättsmält sätt i en linjär berättarform. Dessutom medför faktakomplexiteten i flödena att vi känner att det är nödvändigt med någon form av repetitions-/återkopplingsmoment, vilket gör en icke-linjär lösning olämplig.

För att uppnå ovanstående väljer vi ett övergripande linjärt upplägg för hela flödespresentationen. Vi väljer sedan att kombinera momentlösningarna "Flödesläggare" med "Labyrint" för att kunna dela upp presentationen i två enskilda och lämpliga moment:

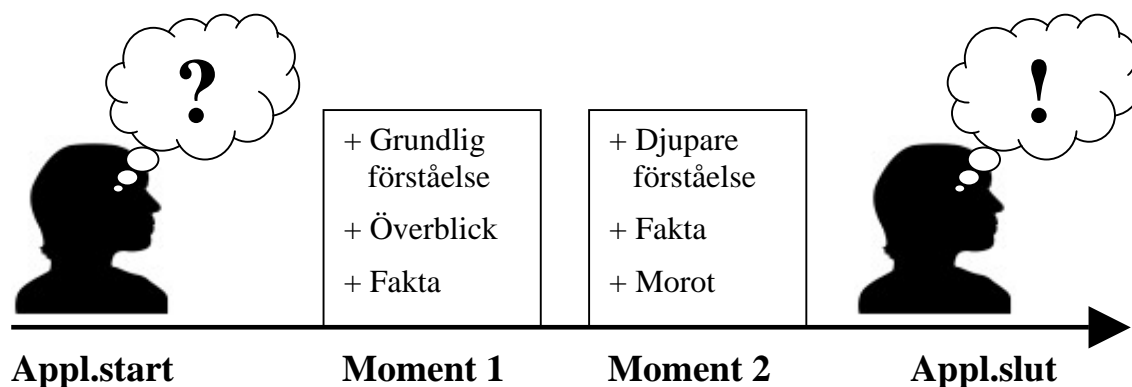
- 1. Visualiseringsmoment** (överblick, grundlig förståelse)
- 2. Förståelsemoment** (repetition, djupare förståelse)

För att kunna ta till sig informationen, d.v.s. innehållet i flödena, måste användaren först och främst förstå konceptet och kunna överblicka vad allt handlar om. Visualiseringsmomentet ska därför ses som en mjukstart där det handlar om att visa användaren hur ett flöde är uppbyggt, och varför en del i flödet finns på en viss plats. Användaren får tänka logiskt och kan skaffa sig en grundlig förståelse om innehållet.

Den stora poängen med visualiseringsmoment är att användaren ska kunna prova sig fram genom att själva placera ut de saknade flödesdelarna, varvid vi utnyttjar pusselinteraktiviteten. Att användarna får testa sig fram och tillåts att misslyckas är en viktig del i inlärningsprocessen [14]. ”Man lär av sina misstag” är mer ett budord än en klyscha inom e-learning, något som vi vill ta fasta på i applikationsmodellen.

E-learning kräver enorm disciplin och någon form av motivation för användaren, och generellt sett är spel- och tävlingsmoment bra medel att nyttja för att höja motivationsfaktorn [3]. Tanken är därför att utforma moment två som ett tävlingsmoment, vilket ska engagera och bidra till bättre inläring.

I förståelsemomentet kontrollerar vi om användaren tar till sig och kan tolka den presenterade flödesinformationen. Detta gör vi genom en frågesport, där det gäller för användaren att svara på frågor av olika karaktär och samla så mycket kunskapspoäng som möjligt. Detta poängsystem är den morot vi tar med för att användaren ska bli mer engagerad, och vilja ta till sig informationen och försöka svara rätt på frågorna.



Figur 5.1 – Huvudmomenten i applikationsmodellen och vad de tillför användaren.

Som figur 5.1 illustrerar så ska alltså de två huvudmomenten komplettera varandra och portionera upp det faktamässiga innehållet för användaren, så att denne har fått en både grundlig och en djupare förståelse för flödet vid applikationsmodellens slut.

5.4 Riktlinjer för applikationsmodellen

För att den nu valda flödespresentationslösningen ska kunna passa ihop med de övriga delarna i applikationsmodellen, behöver vi sätta upp någon form av riktlinjer och avgränsningar inför det fortsatta arbetet. Riktlinjerna ska ses just

som riktlinjer, inte krav, för funktionaliteten som ska implementeras i applikationsmodellen, så att denna knyter an till målgruppens prioriteringar. För att applikationsmodellen ska bli möjlig att utveckla rent praktiskt, och detta på utsatt tid, innefattar riktlinjerna också vissa avgränsningar.

5.4.1 Allmänna riktlinjer

- Applikationsupplägget för hela applikationsmodellen ska vara linjärt och utan krånglig navigering, vilket gör att man snabbt kan komma igång. Detta baseras på målgruppens prioriteringar.
- En inloggningsfunktion ska användas för att simulera en verklig applikation där en användare endast har behörighet att se vissa flöden.
- Applikationsmodellen ska bestå av ett inledande moment med en menysida. Användaren kan där se vilka flöden som finns, läsa mer om dessa och göra sitt val inför kommande moment.
- Applikationsupplägget för flödespresentationslösningen ska vara linjär med tanke på mängden information som ska förmedlas.

5.4.2 Riktlinjer för visualiseringsmomentet

- Rutnätet på pusselytan, där bitarna ska placeras, ska bestå av 36 (6x6) rutor för att kunna täcka in våra utvalda flöden. Är rutnätet mindre får vissa flöden inte plats. Är rutnätet däremot större innebär det att varje bit måste bli betydligt mindre för att kunna rymmas på skärmen, och därmed skulle all information inte få plats.
- Flödets start- och slutpunkter ska vara utritade då de inte är delar i själva flödesstrukturen.
- Begränsa detaljeringsnivån så att max 18 flödesdelar ska kunna användas som bitar. Användaren ska inte utsättas för en situation där det finns så många valmöjligheter att det blir rena gissningsleken att kunna förstå sig på ett flöde. Dessutom ska det gå att representera flödena rent grafisk utan att det blir för smått eller att det blir svårt för användaren att kunna överblicka innehållet.
- I förväg placera ut ett visst antal statiska bitar som en form av vägledning, så att det blir förståeligt var de saknade delarna ska placeras och därmed undvika att allt resulterar i en gissningslek. Mellan två redan givna bitar ska användaren alltså placera ut en bit. Att användarna får denna hjälp med vägen mellan bitarna tror vi snarare bidrar till att ge en överblick och förståelse för hur flödet är uppbyggt, än att det skulle bli för enkelt och banalt.

- Två bitar som ska placeras ut får inte ligga bredvid varandra i rutnätet, eftersom det då blir för stora luckor, vilket kan göra det svårt att se hur bitarna ska placeras ut. Mellan två bitar som ska placeras ut av användaren, ska det alltså finnas minst en vägledande bit.
- Feedback ska ges både när en bit placeras rätt och fel.
- Alla saknade delar i flödet ska placeras ut innan nästa moment tar vid.

5.4.3 Riktlinjer för förståelsemomentet

- En separat fråga ska ställas för varje flödesdel. Frågan ska vara möjlig att besvara korrekt utifrån tidigare presenterad information.
- Vid iterationsval där en och samma bit är återkommande ska olika frågor ställas.
- Vid vägval ska användaren själv få välja vilken väg han/hon vill ta.
- Frågorna ska vara av olika karaktär; radio, check (multiple choice), field, sliders etc.
- En fråga får inte ha för många alternativ, max fem stycken. Detta för att arbetsytan är begränsad och att frågorna inte ska bli omöjliga att besvara.
- Genom att svara på frågorna ska poäng utdelas tillsammans med en kommentar, så att användaren får feedback på om och varför svaret var rätt eller fel.
- Poängresultatet ska presenteras när sista frågan har besvarats och användaren har tagit sig från start till mål.

5.5 Storyboard och detaljskisser

Storyboarden använder vi för att skapa en övergripande bild över applikationens struktur och innehåll. Till detta väljer vi att även ta fram ett antal detaljskisser (se bilaga 6) som mer ingående visar de olika nyckelscenerna i storyboarden. Dessa skisser fungerar som en form av pappersprototyp som presenteras för några testpersoner. Efter eventuella korrigeringar utifrån synpunkter, utgör denna pappersprototyp underlag inför implementeringen.

Följande moment ingår i applikationsmodellen och beskrivs i storyboarden (se bilaga 5):

- 1) **sida1_index.html**
En första HTML-sida som fungerar som en länk på vilken annan webbsida som helst. Sidan innehåller en textlänk för att öppna det externa applikationsfönstret, samt en kortare text där man ombedes ladda ner Flash Player om man inte har denna installerad.

- 2) **sida2_inlogg.swf**
Inloggningsida innehållandes text, logotyp och inloggningsfält. Sidan har två syften:
 - Förhindra att obehöriga får tillgång till företagshemligt material, vilket flödena och dess innehåll kan vara.
 - Marknadsföra applikationen genom en kort och beskrivande information som riktas till de som inte är bekanta med applikationen, t.ex. förstagångs användare eller intresserade ströbesökare.

- 3) **sida3_loader.swf**
Eftersom applikationsmodellen kan ta en stund att ladda in beroende på användarens uppkoppling, så väljer vi att lägga in en ”preloader”, för att användaren ska se att något händer, och vad som händer, d.v.s. att en sida laddas in. Sidan består av en procenträknare som visar hur mycket av applikationsmodellen som laddats in.

- 4-5) **sida4_huvudmeny.swf**
Huvudmenyn (se bilaga 6 för detaljskiss) där användaren kan se vilka flöden som finns, läsa mer om dessa och göra sitt val inför kommande moment. Sidan animeras fram genom att informationsfältet kommer in från vänster och flödesmenyn tonas fram. Sidan ska innehålla:
 - Ljus arbetsyta/bakgrund där all information presenteras (återkommande i alla påföljande moment).
 - Flödesmeny med tillgängliga flöden i alfabetisk ordning.
 - Informationsfält med faktabaserad text om de flöden som finns i menyn. Kompletteras med relevanta bilder.
 - Snabbfakta (översiktlig information) med flödesstorleken och relativ tid det ska ta att gå igenom de kommande momenten.
 - Användarmeny nere i höger hörn, med ljud- och stängknapp (återkommande i alla påföljande moment).

- 6) **sida5_flode[nr].swf**
Det första av de två moment som finns till varje flödesval (se bilaga 6 för detaljskiss). Sidan består av en 6x6-bitars spelplan för

flödespusslet och utslumpade bitar som användaren ska kunna dra och släppa. Spelplanen och bitarna är kvadratiska och platta i formatet. Start och mål är utritade från början, liksom vägen mellan de saknade delarna, som en hjälp om var man ska börja, var man ska gå och var man ska sluta. Varje pusselbit kompletteras med en hjälpruta innehållandes information om pusselbiten, d.v.s. om ett visst steg i flödet, som visas när musen förs över biten.

7) popup_feedback.swf

Dialogruta med feedback på användarens handlande när denna placerat en bit. Här förklarar en presentatör, illustrerad bild, om en bit ligger rätt eller fel, och varför den gör detta. Texten är kort och informativ och skriven i dialogform. När dialogrutan är framme så skuggas bakgrunds innehållet, och blir klickbart igen först när användaren stänger dialogrutan (återkommande för alla påföljande dialogrutor).

8) sida5_flode[nr].swf

När dialogrutan stängs så körs en funktion som kontrollerar ifall en bit placerats på rätt plats, och ifall alla bitar ligger på rätt platser. Utplacerade bitar markeras genom en färg eller symbol för att visa om de ligger på rätt eller fel plats.

9) sida5_flode[nr].swf

När alla bitar ligger på rätt platser så spelas en animation upp med pilar som rör sig. Animationen spelas upp i ca 2-3 sekunder och illustrerar något som flödar från start till mål.

10) popup_teststart.swf

Dialogruta där presentatören introducerar nästa moment – förståelsetestet. Dialogrutan dyker upp automatiskt efter att pilanimationen spelats upp. Introduktionstexten ska vara kort och informativ och förklara de viktigaste sakerna i momentet. Förståelsemomentet startas genom att användaren klickar på startknappen som finns i dialogrutan.

11) sida6_test[nr].swf

Det andra av två moment som finns till varje flöde. Här ska användaren stega sig igenom flödet från start till mål, t.ex. genom att presentatören går framåt mellan bitarna i rutnätet. Hela flödet är utritat från början och tanken är att ta sig till mål och samla poäng på vägen genom spelplanen. Om ett flöde består av vägval ska användaren själv kunna välja väg.

12) popup_testfraga.swf

Dialogruta där presentatören ställer en fråga i anslutning till en flödesdel. Frågorna är i huvudsak textbaserade, men kompletteras ibland med bild och video för att göra det mer tilltalande och förståeligt. När en fråga har besvarats ges feedback i form av en kommentar och plus-/minuspoäng. Bredvid spelplanen ligger en mätare som visar användarens aktuella poäng.

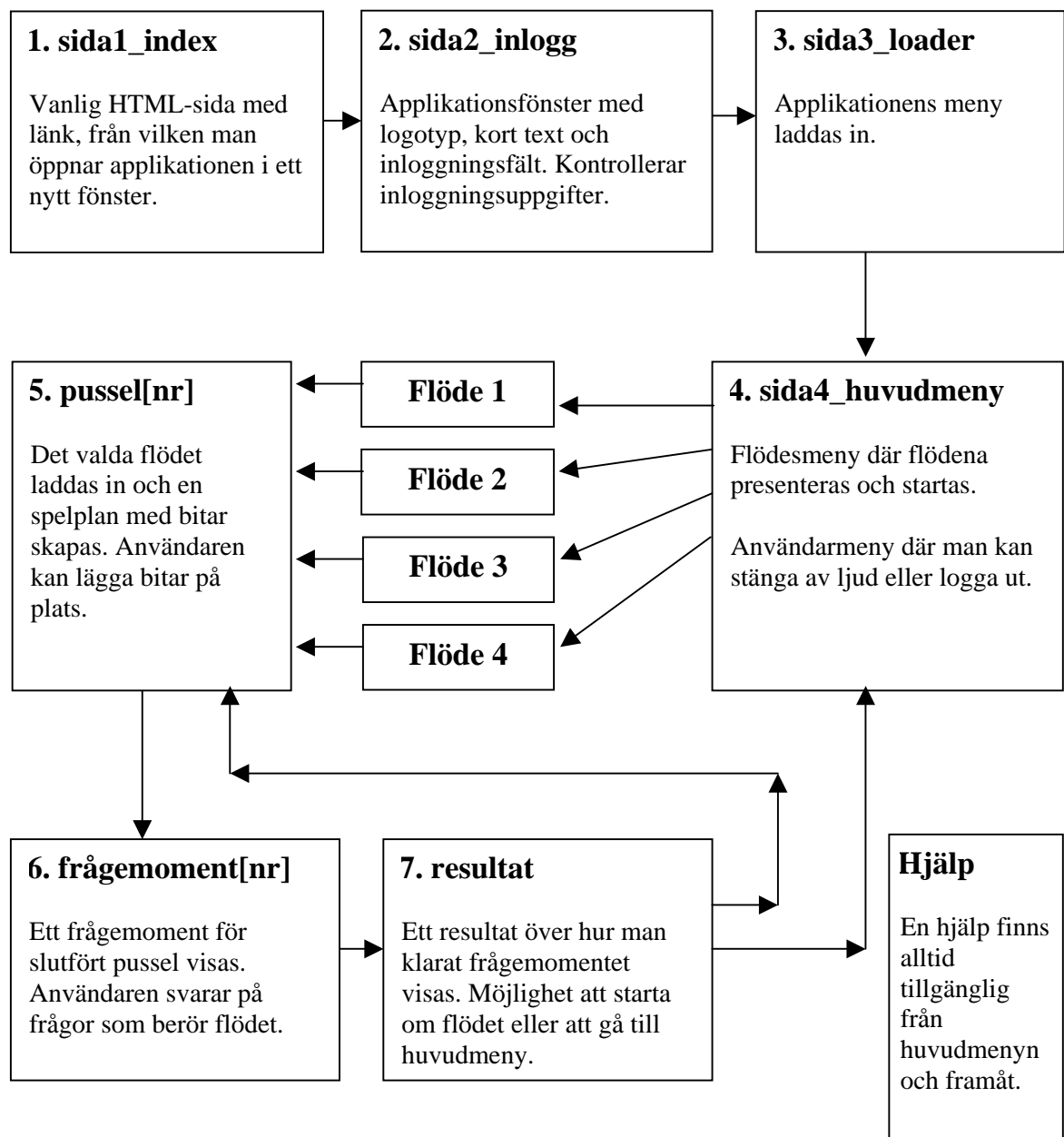
13) sida6_test[nr].swf

Liksom i det första momentet så markeras bitarna genom en färg eller symbol för att visa på rätt eller fel, nu utifrån hur användaren besvarat en fråga. När sista frågan är besvarad och man kommit till målet, så spelas en animation upp med pilar (eller något annat) som rör sig från start till mål.

14-15) popup_testresultat.swf

Dialogruta där resultatet av testet presenteras, tillsammans med snittresultatet från tidigare spelomgångar som hämtas från databasen. Det nya resultatet adderas till det gamla och sparas i databasen. När poängen har presenterats så har användaren två knappar att välja mellan; antingen att trycka på ”Prova igen” och gå tillbaka till pusselmomentet (se steg 6), eller att trycka på ”Huvudmeny” och återgå till huvudmenyn (se steg 5). Vid återgång till huvudmenyn så är det aktuella flödet fortfarande framme.

5.5.1 Walkthrough



Figur 5.2 – Walkthrough med samtliga moment och deras kopplingar i applikationsmodellen.

5.6 Applikationsmodell 0.1 – en första version

Med en klar bild av upplägget för gränssnittet och vilka olika moment som ska ingå, bygger vi upp en första version av applikationsmodellen. I detta första skede är det applikationsmodellens funktionalitet som prioriteras, för att kunna se att våra idéer verkligen fungerar, innan vi ser till att det ser bra ut grafiskt.

Arbetet med applikationsversion 0.1 är inriktat mot att skapa en form av systemskal som innehåller funktionaliteten för de ingående momenten, så att vi direkt kan genomföra testningar av dessa. Innehållet, i form av text och fakta, och den grafiska profilen tillfogas sedan allt eftersom.

Det finns några nyckelmoment som kräver extra arbetsinsatser, både vad gäller implementering och testning. Framförallt handlar det om huvudmenyn, pusselmomentet och frågemomentet, som alla innehåller programmeringskod och funktioner som vi valt ska användas i samtliga pussel- och frågemoment.

Vi arbetar helt utifrån storyboarden och implementera huvudfunktionaliteten moment för moment, för att sedan testa dessa innan vi går vidare på nästa.

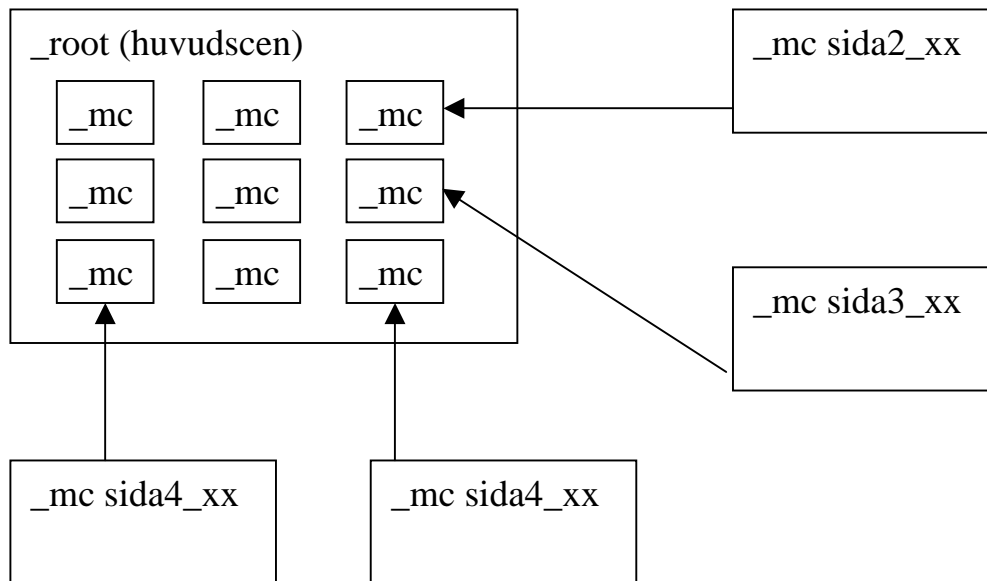
5.6.1 Applikationens uppbyggnad

När det gäller applikationsmodellens uppbyggnad (se figur 5.3) väljer vi att arbeta utifrån en metod där alla moment är fristående, men samtidigt hänger ihop på ett eller annat sätt. Vi skapar de olika sidorna som separata, externa, swf-filer som vi länkar in som movieclips på en och samma huvudscen.

Huvudscenen består enbart av ett skal med containers (behållare för movieclips), en för varje sida som applikationsmodellen består av. Till dessa containers länkar vi sedan in våra olika swf-filer. När en fil laddats in i en container på huvudscenen ser vi samtidigt till att innehållet i de andra containrarna tas bort, och på så sett existerar enbart det fönster vi jobbar med för tillfället. Det är då bara det enskilda utplacerade movieclipet som får sitt innehåll utbytt, medan själva huvudscenen förblir oförändrad. Detta är inte märkbart för användaren, och förkortar samtidigt laddningstiden eftersom bara en del av applikationsmodellen behöver laddas in.

Mycket av koden flyttas ut från själva swf-filerna och läggs i externa ActionScript-filer⁷. Detta gör vi med anledning av att koden blir ganska omfattande och svårhanterlig på några av sidorna. Dessutom har vi en hel del funktioner och metoder i koden som är gemensamma för flera swf-filer, t.ex. de som skapar spelplanen och flödesbitarna som ska läggas, vilket gör att det är onödigt att skriva koden för dessa flera gånger.

⁷ ActionScript – skriptspråk baserat på JavaScript som används i Adobe Flash. Kod skriven i ActionScript kan placeras i externa filer av typen .as.



Figur 5.3 – Applikationsstrukturen med externa, separata swf-filer som länkas in i tomma movieclips på huvudscenen.

Det finns en rad anledningar till att vi väljer att konstruera applikationsmodellen enligt ovanstående arbetsmetod;

- Vi får en bättre struktur och överblick, både vad gällande de olika sidorna och själva koden.
- Det är lättare att fördela arbetet eftersom vi i stort sett skapar de olika sidorna fristående från varandra. Förutom en del funktioner så finns ingen egentlig koppling mellan sidorna, och var och en länkas oberoende in på huvudscenen när det behövs. Funktioner som delas mellan olika sidor utgör egna externa ActionScript-filer.
- Då applikationsmodellen ska vara webbaserad måste vi tänka på prestanda och laddningstid. Sammantaget blir applikationsmodellen ganska stor, men då vi väljer att dela upp den i mindre fristående delar som laddas in allt eftersom, blir laddningstiden inte så påtaglig.
- Genom att vi inte laddar om hela huvudscenen, utan bara ett movieclip som ligger utplacerat, så ger detta oss en möjlighet att ha flera separata fönster öppna samtidigt. Genom att ladda in ett nytt movieclip och samtidigt låta det gamla ligga kvar i bakgrunden, så kan vi vid behov växla mellan två olika fönster.

En nackdel ligger i att behöva förlita sig på webbservern. Bryts anslutningen till denna så har man inte tillgång till de olika swf- eller ActionScript-filerna och en sida eller funktion kan laddas in ofullständigt. Att användaren har en fast och fungerande internetuppkoppling är dock något vi utgår från.

5.6.2 Gränssnittet – färg och form

När vi utformar gränssnittet för applikationsmodellen fokuserar vi på själva innehållet och att ge ett allmänt lättanvänt intryck, så att användaren inte ska känna att det tar emot att komma igång. Eftersom vi inte vill att det visuella ska stjäla fokus från innehållet, är resultatet ett ganska stramt och stilrent gränssnitt där färg och form används sparsamt. För att inte riskera att applikationsmodellen blir för tråkig, lägger vi till några mer lättsamma element (t.ex. presentatören).

Huvudmenyn utformas så att både dess innehåll och struktur är klar och tydlig och enkel att använda. I flödesmenyn används tydliga markeringar för att visa den valda kategorin och det valda flödet, vilket sedan kan återknytas till flödesrubriken som visas i informationsfältet. Avdelningen för snabbfakta använder vi för att användaren ska få en bra överblick över flödesinnehållet.

Generellt är all text skriven i kort men koncentrerad form, främst p.g.a. applikationens begränsade utrymme. Det krävs en hel del koncentration från användaren för att kunna ta till sig faktamaterialet, och för att underlätta läsbarheten så använder vi oss av textindrag, punktlister, styckeuppdelning etc. Typsnitten är anpassade för skärmläsning eftersom vi inte bygger in någon utskriftsfunktion (säkerhetsåtgärd p.g.a. företagshemligheter).

I övrigt använder vi oss av saker som de flesta datoranvändare känner igen; ikonerna i användarmenyn har vedertagna symboler som känns igen från en vanlig webbläsare, knappar ändrar färg när man för muspekaren över dem, muspekaren ändrar form för att indikera klickbara områden.

5.6.2.1 *Det multimediala innehållet*

Eftersom en kombination av olika medietyper medför att användaren lättare kan ta till sig informationen, är det en sak som vi försöker ta hänsyn till i gränssnittsutformningen. Applikationsmodellen utformas så att allt från huvudmenyn, till själva pusselbitarna som ska placeras ut, kan innehålla video, bild, text, eller en kombination av dessa.

Bild och, framförallt, text är de klart dominerande medietyperna eftersom det stundtals komplexa och abstrakta innehållet i flödena är svårt att illustrera genom video eller ljud. Risker är att vi försvårar det ytterligare genom att tillföra ett moment där användarna måste tolka innehållet i en videosekvens, för att därefter kunna tolka informationen som ska förmedlas.

Ljud används främst i feedbacksyfte och för att göra användaren uppmärksam på att något viktigt händer, exempelvis att en fråga dyker upp som användaren ska svara på.

Ett par exempel på videosekvenser implementeras i frågedelen, där en fråga kopplas till det som utspelar sig på filmen. Av tidsmässiga skäl kommer dock inga egna videosekvenser med flödesrelaterat innehåll att användas.

5.7 Informationspresentation och pedagogik

En viktig del i utformandet av applikationsmodellen är hur man presenterar den faktabaserade informationen på ett bra sätt, och det är där pedagogiken kommer in i bilden.

Vi väljer att införa en karaktär, en tecknad presentatör, som löper som en röd tråd genom applikationsmodellen. Denna presentatör förmedlar faktabudskap och ger feedback genom att härma en dialog mellan användaren och applikationsmodellen, vilket skapar en mer personlig prägel och gör användaren mer mottaglig för information [2, 6, 7 m.fl.]. Detta medför då också att det som presenteras i hjälpväg inte behöver ses som en manual, utan snarare som ett personligt hjälpmeddelande.

Ett annat viktigt grepp som vi använder är det klassiska uttrycket ”learning by doing”, d.v.s. att användarens lär genom handling. Genom att användaren är delaktig och interagerar med applikationsmodellen, och erhåller feedback, kan han/hon dra lärdom av sitt handlande. Detta utnyttjar vi rent konkret i vår applikationsmodell, då användaren ska placera ut bitar i ett rutnät.

Feedbacken som ges är konstruktiv och förklarande så att användaren ska förstå innebörden och konsekvensen av sitt handlande. Genom att vi inte använder feedbacken som en pekpinne eller en uppmaning, minskar risken att skapa distans till användaren.

Slutligen är det också viktigt att användaren får testa sin kunskap i någon form av tävlingsmoment, vilket görs i förståelsemomentet. Genom att svara på ett antal frågor, erhålla poäng och få feedback, ser användaren om han/hon har förstått det tidigare momentet.

5.8 Metafor för flödespresentationen

I sin renaste form kan man tänka sig att enbart visa användaren ett flödesschema över ett specifikt flödes uppbyggnad, men troligen är detta både för abstrakt och, framförallt, ointressant för att användaren ska vilja ta till sig informationen. Genom en metafor omvandlar man denna abstrakta information och gör det intressant och lättbegripligt, vilket förkortar inlärningsprocessen och användarens vilja att ta till sig det okända systemet

[1]. Om applikationsmodellen saknar en metafor ökar risken för att budskapet, att förmedla information hur ett flöde ser ut och varför det ser ut som det gör, inte tas emot av användaren.

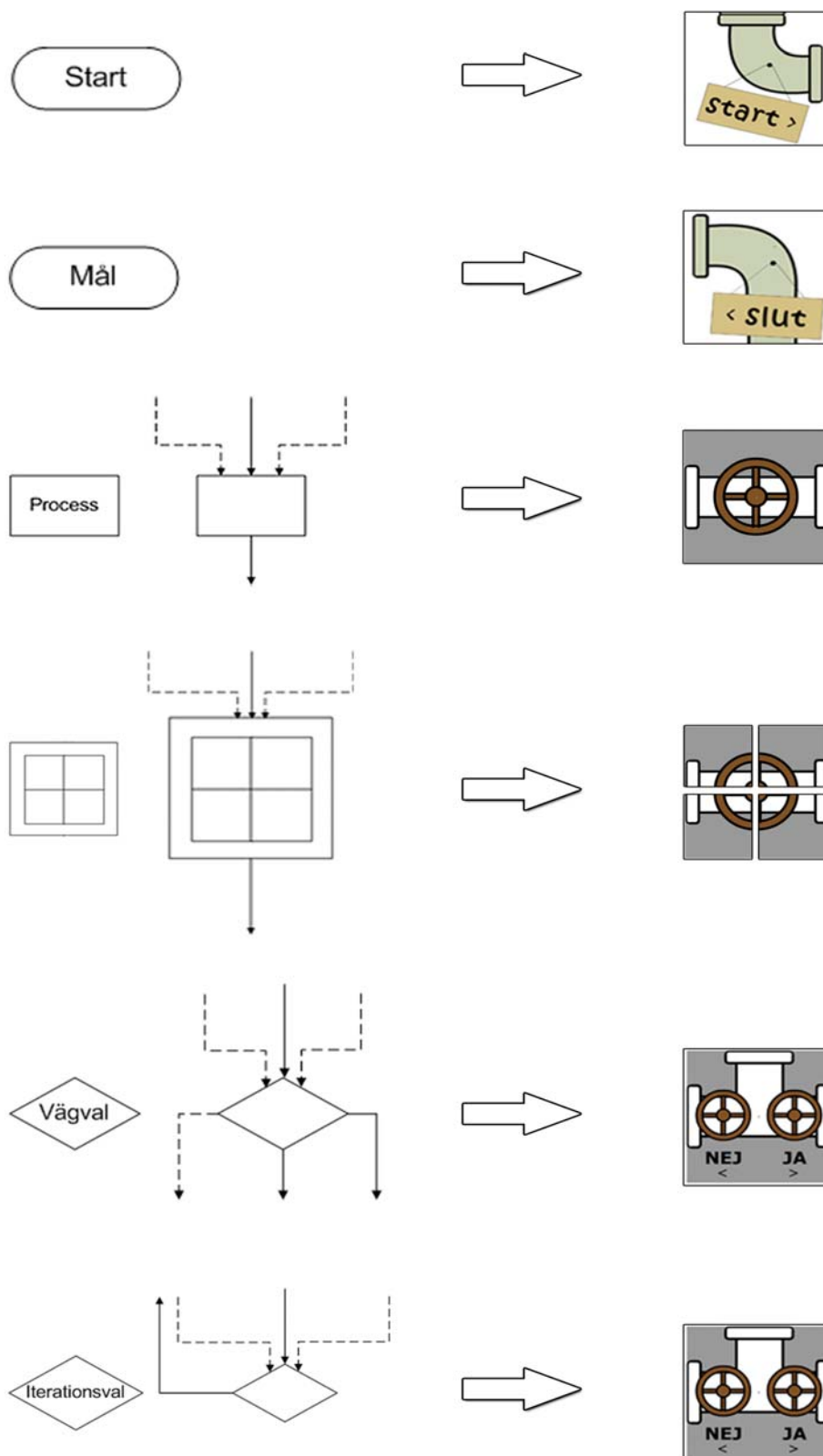
Man kan säga att metaforen används för att beskriva något som användaren inte har en aning om, med något som användaren redan känner till, d.v.s. ett känt system används för att beskriva ett okänt system. Det är alltså viktigt att tänka på att målgruppen kan relatera till metaforen och att den är allmänt vedertagen. Om så inte är fallet, kan man inte beskriva det okända systemet, eftersom metaforen också är okänd.

Vi har kommit fram till ett antal olika tänkbara förslag på metaforer, allt från grottexpeditioner och skattjakter till versioner av Rubriks kub. För att kunna göra vårt val så väljer vi att väga respektive för- och nackdelar, och diskutera om metaforen är lämplig eller ej. Det är nämligen inte så passande att beskriva hur man t.ex. konstruerar en bil, med någon form av flygplansmetafor, eftersom detta kan leda till vissa moraliska dilemman och frågor som ”vad f-n har detta med ett flöde att göra?”. Till detta måste vi även tänka på att metaforen ska passa in på målgruppen och användningsmiljön.

Vårt val faller på en metafor som vi känner att många användare på något sätt kan relatera till, och som samtidigt inte är specifikt för en viss typ av företag. Metaforen beskrivs lättast som en kombination av ett pussel och spelet Pipes⁸, där man ska lägga rörbitar för att bygga färdigt en rörledning från start till mål. Bitarna representerar de olika momenten i flödet och bildar, när dessa ligger på sina rätta platser, delar i rörledningen (flödesstrukturen).

”Få flödet att flöda” är en metaforisk beskrivning av vår metafor. Poängen är att se till att röret blir komplett så att vattnet har möjlighet att flöda från start till mål, vilket visualiseringsmomentet handlar om. I det påföljande förståelsemomentet ska man sedan se till så att vattnet rinner genom röret, vilket görs genom att svara på frågor och vrida på kranar som sitter längs med vägen. Figur 5.4 visar hur de olika komponenterna i vår flödesstruktur presenteras för användaren genom metaforen.

⁸ Pipes – ett klasiskt arkadspel från 80-talet där man ska hinna bygga ihop ett system med rör som har olika former (svängda, raka, korsade o.s.v.). Man bygger rörsystemet på tid för att hindra forcerande vatten från att läcka ut.



Figur 5.4 – Flödesstrukturens komponenter och hur de presenteras i applikationsmodellen.

5.9 Löpande test och regressionstest

Efterhand som nya moment implementeras vill vi kontrollera så att de tidigare fortfarande fungerar, vilket är vad regressionstesterna syftar till. Det kan t.ex. handla om att stämma av så att nya (eller befintliga) variabler får önskade värde när en ny funktionalitet tillfogas. Vi väljer att påbörja regressionstestningen i ett tidigt skede, redan när själva applikationsskalet skapas, för att lättare kunna spåra problem och sätta in åtgärder.

Utöver regressionstesterna utför vi även löpande tester, framförallt gällandes nya funktioner och enstaka moment. Även dessa test påbörjas i ett tidigt skede så att vi snabbt får feedback på om användare förstår och kan använda sig av de nya funktionerna, samt om dessa fungerar rent tekniskt.

I storyboarden finns en hel del moment som vi anser kritiska och som vi väljer ut som lämpliga att testa löpande:

- **Inloggningen**
Fungerar databaskopplingen så att inmatade värde i inloggningsfälten kontrolleras mot värdena i databasen? Kommer man till huvudmenyn när man loggat in? Kan användarens namn hämtas från databasen och användas vid senare tillfälle?
- **Huvudmenyn**
Går det att välja ett flöde i flödesmenyn? Presenteras rätt information i informationsfältet? Förstår man vilket flöde man valt? Förstår man kategorierna och ordningen i menyn? Går det att starta ett flöde?
- **Användarmenyn**
Hittar man menyn och förstår vad de olika menyvalen ska användas till? Kan man stänga av (och sätta på) ljudet? Får man fram hjälpfunktionen? Går det att logga ut?
- **Visualiseringsmomentet**
Går bitarna att dra och släppa? Snappar de till rutnätet? Ges det någon feedback när en bit placeras? Kommer man vidare till nästa moment när bitarna ligger på rätt platser?
- **Förståelsemomentet**
Förstår man vilken fråga man är på? Är det rätt fråga som kommer fram? Får man poäng när man besvarar en fråga? Går det att komma vidare när man svarat på en fråga?

- **Textinnehåll**
Förstår man texten som presenteras? Är texten läsbar på skärmen? Är texten relevant?
- **Ljud och video**
Spelas ljud och video upp som det ska? Förstår man innehållet i videosekvenserna?

De löpande testerna utförs i första hand av tre oberoende personer, som har tillgång till applikationsmodellen online. Dessa personer testar ovanstående moment och ger skriftlig feedback på vad som är bra, respektive mindre bra. Ett antal testningar, där vi observerar testpersoners handlande för att se deras agerande gentemot applikationsmodellen, genomförs också löpande.

5.9.1 Testutvärdering och testernas inflytande på arbetet

Mest positiva är testpersonerna till huvudmenyn, då de tycker att dennas struktur och informationsinnehåll är lättförståelig. En liten korrigerings är dock gjord då flödesmenyn är flyttad från höger till vänster sida, där användarna ansåg att man borde utgå från.

En presentatör har införts tidigt i utvecklingen av applikationsmodellen, i syfte att kunna presentera information på ett mer personligt sätt. Helt enligt storyboarden så dyker denna presentatör upp i en egen dialogruta så fort användaren får någon form av feedback. Detta ogillas av testpersonerna då de behöver stänga ner rutan hela tiden, vilket ter sig irriterande och onödigt. Då det dessutom inte är något tvång att läsa texten, så känns det ännu mer omotiverat att den dyker upp i ansiktet på användaren. Därför har presentatören blivit mer integrerad i gränssnittet, och dyker således inte upp i en dialogruta utan direkt på arbetsytan.

Användarmenyn är ett moment som testpersonerna har många åsikter om. Vid observationstestning visar det sig att de flesta inte upptäcker denna, om man bara ber dem att logga ut. Ber man dem däremot att stänga av ljudet eller att öppna hjälpen, så hittar de menyn nästan omedelbart. Detta med anledning av att testpersonerna förutsätter att denna funktionalitet finns i kanten eller hörnan av en applikation. Vi åtgärdar problemet genom att göra användarmenyn tydligare med större knappar, samt tillfoga den inloggades namn bredvid knapparna, vilket vi tror hjälper till att skapa fokus.

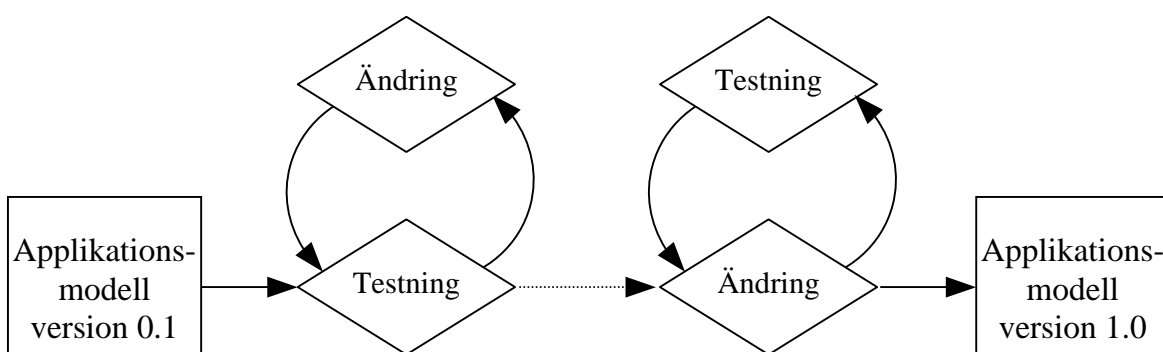
Genom observationstest av visualiseringsmomentet inser vi att upplägget för pusslet inte är riktigt så självklart som vi föreställde oss vid implementeringen. För några av testpersonerna har det tagit en bra stund innan de förstått vad

som skulle göras. Andra har behövt hjälp för att överhuvudtaget kunna komma igång. Här gör vi en rad visuella förbättringar såsom att ha olika färger på bitar som ska placeras kontra spelplanen i övrigt, och förändra en bits form, genom att lägga till en kran när en bit lagts på rätt plats.

Förståelsemomentet uppfattas som klart och tydligt i sin utformning, men testpersonerna har påpekat att frågorna består av mycket text och vaga formuleringar. Detta tar vi fasta på och väljer att skära ner mängden text och koncentrera frågeställningarna.

Ett par av testpersonerna har starka invändningar mot att man bara kan komma tillbaka till huvudmenyn genom att antingen stänga av och logga in igen, eller att göra klart förståelsemomentet och därefter klicka på knappen ”Huvudmeny”. Detta är något som vi väljer att åtgärda genom att addera en knapp till användarmenyn, där man nu kan avbryta ett moment och återgå till huvudmenyn.

Testningen har utförts iterativt enligt figur 5.5, och de förändringsåtgärder som gjorts visar sig vara tillfredsställande. De oberoende testpersonerna upplever förändringarna som positiva efter att de återigen provat de problematiska momenten. Dessutom har vi valt att, efter de större redigeringarna, testa funktionaliteten på nya testpersoner som inte tidigare sett applikationsmodellen. Även dessa ser ut att hantera den tidigare problematiken på ett mycket bättre sätt.



Figur 5.5 – Vårt löpande testförfarande enligt en iterativ metod.

När det gäller regressionstestningen så har denna redan från start kunnat genomföras på förväntat sätt, med det arbetsupplägg och den applikationsstruktur vi har valt. Det som vi upplever kräver störst arbetsinsats i regressionstestningen är att kontrollera de variabler, funktioner och komponenter (t.ex. användarmenyn) som delas mellan applikationsmodellens olika moment.

5.10 Applikationsmodell 1.0

Genom en omfattande utvecklingsprocess med många löpande tester och förändringar, består nu applikationsmodellen av en fullt fungerande inloggningssida, som leder till en betydligt mer logisk och strukturerad huvudmeny än tidigare. På huvudmenyn presenteras flödena med tillhörande texter och bilder, samt en faktaruta som ger information om hur lång tid och hur svårt ett valt flöde är. Detta kompletteras sedan med en kunskapsmätare som återspeglar användarens tidigare resultat, vilket uppdateras efter varje spelomgång.

Genom att starta ett flöde börjar visualiseringsmomentet, där man presenteras för en spelplan med ett rörsystem som ska byggas ihop av de bitar som flödet består av. Bitarna ligger utslumpade, och håller man muspekaren över någon av dem, dyker det upp information som man har nytta av för att slutföra byggandet. Under tiden som bitarna placeras ut så ger presentatören feedback. I en pratbubbla bredvid spelplanen förklarar han om man gjort rätt eller fel och varför det är så.

När man klarat av att bygga ihop flödet startar förståelsemomentet, där man ska leda vatten genom det nyss hopbyggda rörsystemet. Detta görs genom att presentatören återigen dyker upp, denna gång för att ställa frågor som användaren ska svara på. Genom att besvara dessa frågor samlar användaren kunskapspoäng och kan låsa upp kranar längs med röret, varvid vattnet kan flöda vidare. När man fått vattnet att ta sig igenom systemet och fram till mål, redovisar presentatören resultatet tillsammans med en kort kommentar. Slutligen får man valet att starta om flödet eller att gå tillbaka till huvudmenyn.

Resultaten från de löpande testerna har visat på att applikationsmodellen har blivit tydligare och mer lättanvänd allt eftersom. Den stora skillnaden mot den tidigare versionen är att applikationsmodellen nu innehåller alla tänkta funktioner, grafiska element och relevanta texter. Huvudmenyn har strukturerats upp ytterligare, användarmenyn har förtydligats och kompletterats, faktatexterna har beskurits och komprimerats o.s.v.

Att vi väljer att ge applikationsmodellen versionsnummer 1.0 beror på att vi anser den vara funktionsduglig i en skarp miljö, och därmed är redo för sluttestning. I sluttestningen ska det visa sig om målgruppen kan hantera applikationsmodellen på förväntat sätt och vilka sista justeringar som ska göras innan en slutversion uppnås.

5.11 Sluttestning

Syftet med sluttestet är att testa samtliga funktioner när dessa nu anses färdigimplementerade, för att därefter kunna utvärdera i vilken grad applikationsmodellen uppfyller företagets krav och målgruppens prioriteringar.

Sluttestningen genomförs i två steg; först ett systemtest där vi som utvecklare själva kontrollerar applikationsmodellen mot uppställda företagskrav, därefter scenariotestningar mot målgruppen som ska visa på om applikationsmodellen matchar målgruppens behov och prioriteringar. Scenariotestningen mot målgrupp består i sin tur av tre moment; observation, frågestund med återkoppling till observationen och enkätsvar.

Inför scenariotestningen mot målgrupp väljer vi ut tre respondenter med en ledande befattning inom försäljning eller projektledning. Samtliga alltså inom ramen för den tilltänkta målgruppen, men ändå med lite olika arbetsuppgifter, kunskaper och erfarenheter.

- **Respondent ett (R1)**
R1 är en kvinna i 40-årsåldern som arbetar som projektledare och ansvarar för mediakurser. Hon har arbetat på företaget i ca fem år och tycker sig ha mycket god kännedom om verksamheten.
- **Respondent två (R2)**
R2 är en 26-årig manlig projektledare med ansvar för teknikkurser. R2 är ingenjör och har mycket datorvana, men däremot ingen större egen erfarenhet av e-learning.
- **Respondent tre (R3)**
R3 är en 34-årig kvinnlig säljchef. Hennes främsta arbetsuppgifter är att knyta nya kundkontakter och vara med i uppstarten av nya projekt. Hon är relativt ny på företaget men har erfarenhet inom projektering sedan tidigare. Hon är väl insatt i konceptframtagningen av e-learningapplikationer.

Respondenterna har inte någon specifik erfarenhet av utveckling av e-learningapplikationer, vilket är ett medvetet val från vår sida. Vi vill i största möjliga mån simulera verklighetstroga situationer med icke-insatt personal, som använder applikationsmodellen utifrån ett mer oberoende synsätt. De med erfarenhet av e-learning har i stället rådfrågats löpande under arbetet, men som sagt, väljer vi att inte involvera dessa under sluttestningen.

Observationerna genomförs i verklighetstrogna situationer, på användarnas egna kontor och på deras egna datorer, dels för att användaren ska känna sig bekväma i situationen, men även för att vi vill se hur applikationsmodellen fungerar i skarpt läge. Varje testperson genomför fyra olika scenarion, vilka stegvis blir mer och mer komplexa. En av oss läser upp scenariona för testpersonen, medan den andra observerar och noggrant antecknar hur användaren agerar. I god testanda så;

- förklarar vi syftet med testet för testpersonen
- förklarar vi för testpersonen att denna kan avbryta närhelst denna vill
- förklarar vi att testresultatet är anonymt
- ber vi testpersonen tänka högt
- ger vi inte testpersonen någon hjälp (om det inte är absolut nödvändigt)

Direkt efter observationen sätter vi oss ned för att diskutera delar av våra anteckningar tillsammans med testpersonen. Vi fokuserar då på de konkreta saker som nedtecknats, och då särskilt på de problem som testpersonen påträffade. Vi diskuterar alltså inte vad testpersonen tycker, utan fokuserar istället på att diskutera dennes beteende. Forskning inom området visar nämligen att användarnas åsikter vid testning är tämligen värdelösa – det är testpersonens beteende i konkreta, realistiska situationer som är av intresse för utvecklingsprocessen [8]. Frågestunden är inte standardiserad utan anpassas helt utifrån vad som observerats.

Enkäten delas ut efter observationen och testpersonerna ombeds lämna in denna någon dag senare, för att hinna få perspektiv och reflektera över behållningen av applikationsmodellen. Genom enkäten vill vi ta del av testpersonernas allmänna åsikter om applikationsmodellen, både sett till funktion och innehåll. Fokuseringen ligger alltså inte på detaljnivå, då detta redan diskuterats i anslutning till observationen.

5.11.1 Testscenarion

Testscenariona tas fram med utgångspunkt i målgruppens krav och prioriteringar, och dels utifrån den feedback vi fått från de löpande testerna där vi kunnat se vilka moment som bör testas en extra gång. Testscenariona ska visa på ifall punkterna i kapitel 3.5.4 är uppfyllda, d.v.s. om applikationsmodellen kan göra de saker som användaren har behov av.

Nedanstående scenarion utgår från HTML-sidan och är skrivna oberoende av varandra, om ifall att någon testperson inte har tid eller lust att genomföra hela testningen.

- **Testscenario 1**

- 1) Öppna applikationsfönstret och logga in med användarnamn *test* och lösenord *test*.
- 2) Navigera dig fram och ta reda på information om flödet för datorkonstruktion. Får du någon uppfattning om svårighetsgraden för flödet redan här, så att du vet detta innan du fortsätter?
- 3) Logga ut på valfritt sätt.

Detta scenario syftar till att se om användaren kan logga in, navigera sig fram och ta reda på information om specifika flöden på egen hand. Scenariot ska även visa på om användaren lokaliserar användarmenyn och kan tolka dess symbolik för att hitta logga-ut-knappen.

- **Testscenario 2**

- 1) Öppna applikationsfönstret och logga in med användarnamn *test* och lösenord *test*.
- 2) Öppna hjälpfunktionen.
- 3) Tag reda på information om användarmenyn.
- 4) Stäng hjälpfunktionen.
- 5) Logga ut på valfritt sätt.

Detta scenario syftar till att se om användaren kan hitta hjälpfunktionen och navigera i denna för att hitta efterfrågad information.

- **Testscenario 3**

- 1) Öppna applikationsfönstret och logga in med användarnamn *test* och lösenord *test*.
- 2) Navigera dig fram och starta flödet ”Skicka fax”.
- 3) Följ instruktionerna och slutför första momentet (pussel).
- 4) När pusslet är slutfört, återgå till huvudmenyn snarast möjligt.
- 5) Logga ut på valfritt sätt.

Detta scenario syftar till att se om användaren kan hantera funktionaliteten i pusselmomentet, samt att se om de förstår flödets uppbyggnad och den feedback som ges. Scenariot ska även visa på om användaren använder användarmenyn för att återgå till huvudmenyn.

- **Testscenario 4**

- 1) Öppna applikationsfönstret och logga in med användarnamn *test* och lösenord *test*.
- 2) Navigera dig fram och starta flödet ”Utbildningsadministration”.
- 3) Följ instruktionerna och slutför första momentet (pussel).
- 4) Följ instruktionerna och starta andra momentet (test).

- 5) Svara på två-tre frågor.
- 6) Stäng av ljudet.
- 7) Svara på en-två frågor.
- 8) Sätt igång ljudet.
- 9) Slutför momentet genom att svara på resterande frågor och följa angivna instruktioner.
- 10) När resultatet är presenterat, återgå till huvudmenyn snarast möjligt och skumma igenom informationen om flödet. Vad har ändrats?
- 11) Logga ut på valfritt sätt.

Detta scenario syftar till att se om användaren kan ta sig igenom hela applikationsmodellen på egen hand. Vi vill genom scenariot undersöka hur testpersonen navigerar, hanterar pusselmomentet och frågemomentet, samt ifall testpersonen kan notera sitt resultat.

5.11.2 Resultat, utvärderingar och förändringar

5.11.2.1 Systemtest

Systemtestet visar att krav 1001-3003 i kravspecifikationen är uppfyllda, med följande noteringar och tillägg:

- **Krav 1001 – Tidsplanen**
Tidsplanen har uppdaterats löpande, men inte i den takt som skulle behövas.
- **Krav 2012 – Databas**
Vi har använt oss av en webbversion av MySQL i stället för en SQL-server. Detta eftersom vi startade med att skapa databaskopplingen i applikationsmodellen under tiden som de anställda på Informator hade semester, vilket innebar att vi inte kunde få tillgång till någon intern SQL-server. Detta har inte inneburit några förändringar i arbetet.
- **Krav 2014 – Upplösning**
Applikationsmodellen har testats på flera olika skärmar med olika upplösningar. 1024*768 är absolut minimum för att applikationsfönstret ska rymmas på skärmen. En skärmupplösning mellan 1024*768 och 1280*1024 är att rekommendera.
- **Krav 2026 – Videoformat**
Vi använder oss av AVI som grundformat för video. Eftersom vi väljer

att ha ett strömmande videoformat så har dock filerna fått omvandlas från AVI till FLV⁹ under arbetets gång.

- **Krav 2030 – Obligatoriska ljudmoment**

Även om enkätundersökningen visade att nästan samtliga i målgruppen hade möjligheten att spela upp ljud, så förekommer inte några obligatoriska ljudmoment i applikationsmodellen med tanke på dess användningsmiljö. Applikationsmodellen innehåller ljud, men detta främst i syfte att påvisa viktiga händelser för användaren. Att ha ljudet påslaget är att rekommendera, men alltså inte nödvändigt.

Förutom databaskravet så uppfylls alltså samtliga krav enligt specifikationen. Med tanke på att denna avvikelse inte förändrar funktionaliteten på något sätt, och inte heller är märkbar för användaren, så anser vi att applikationsmodellen får ses som godkänd utifrån kravperspektivet.

5.11.2.2 Scenariotest

Genom denna sista del i slutttestningen har en hel del nya, både större och mindre, problemområden har uppdagats. Dessa har vi valt att analysera och, om det känns befogat, åtgärda inför varje nytt testfall. En sammanfattning av vad som framkommit presenteras i bilaga 7.

Många intressanta fakta har framkommit och observationerna av våra tre respondenters agerande visar på flera saker som vi upplever som positiva. T.ex. så tycker samtliga respondenter att strukturen på huvudmenyn varit bra. De anser sig få en bra överblick över vad som finns och kan i stort sett omedelbart hitta vad som efterfrågas, och därigenom snabbt komma vidare. Själva flödesmenyn och dess kategorier uppskattas, sett både till placering och till funktion.

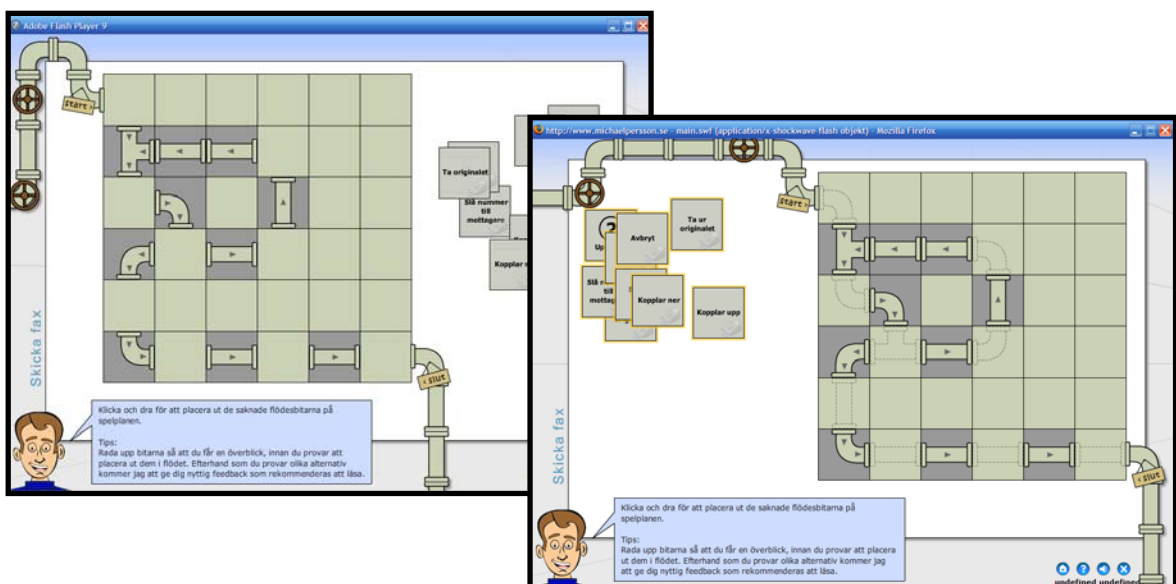
Användarmenyn är fortsatt något av ett problem. De förändringar vi gjort utifrån de löpande testerna har gett viss effekt, men inte tillräcklig. Menyn tar fortfarande en stund att upptäcka, vilket observerats i två testfall. En respondent påpekar också att hjälptexten som dyker upp ovanför menyknapparna är alldeles för liten. Vi gör därför ytterligare ett försök att göra denna meny tydligare, genom att förstora och ändra färg på både knappar och text.

Vi har blivit positivt överraskade över hjälpfunktionen och respondenterna visar inga tecken på att ha problem med att finna var man öppnar denna, trots

⁹ FLV – Flash Video. Ett populärt filformat för att baka in och strömma filmer på webbsidor.

att de annars har svårt att upptäcka själva användarmenyn. De gillar hjälpens utseende och enkla navigering, men har synpunkter på att innehållet inte är särskilt givande. Detta väljer vi att åtgärda så att hjälptexterna blir mer informativa.

Ett problem är att respondenterna inte vet vad de ska göra efter att de startat ett flöde och kommit till visualiseringsmomentet. De förstår inte vilka som är flyttbara pusselbitar, eller var dessa ska placeras ut, vilket leder till att de förvirrat klickar sig omkring i applikationsmodellen efter hjälp. En starkt bidragande orsak till detta visar sig vara att bitarna ligger till höger, och spelplanen till vänster. Bitarna, där interaktionen finns, hamnar inte i blickfånget, utan istället fokuserar respondenterna på den vänstra halvan av applikationsfönstret där spelplanen ligger. För att åtgärda detta börjar vi med att spegelvända hela momentet, så att bitarna hamnar till vänster, och spelplanen till höger. Vi ser också till att de bitar som ska placeras ut är svagt blinkade. Detta för att göra användaren uppmärksam på att det finns något att interagera med. En sista åtgärd görs också för att man lättare ska förstå var bitarna ska placeras. Vi markerar rutorna där bitar ska placeras genom att där lägga skuggade/streckade rör. Det blir då enklare att följa vägen i flödet, samtidigt som man får en mer logisk bild över hur bitarna ska placeras. Dessa till synes ganska små (se figur 5.6), men drastiska förändringar, gör uppenbar nytta och respondent tre uppvisar inga som helst problem med funktionaliteten i detta moment.



Figur 5.6 – Förståelsemomentet i applikationsmodellen, före och efter ändring av spelplanen och bitarna.

I samband med att respondenterna inte vet hur de ska börja, märker vi att de av irritation inte läser den information som finns. När vi efter testet frågar varför de inte läste instruktionen som hade medfört att de kunnat komma vidare, får vi reda på att texterna inte känns inbjudande. Det är de stora textmängderna, i kombination med ett ibland svårtolkat innehåll, som gör att respondenterna gärna undviker att läsa informationen. Vi tittar därför över samtliga texter och försöker göra dem mer inbjudande. Vi tar bort en hel del överflödigt text, och försöker skriva på ett enklare språk.

Förståelsemomentet har respondenterna inga direkta svårigheter med, och upplägget visar sig vara uppskattat. Problemet här ligger i att vårt budskap inte går fram som vi hade tänkt. Respondenterna svarar på frågorna, och läser feedbacken som visar ifall de svarat rätt eller fel. De läser också instruktionen för hur man ska vrida på kranarna för att komma vidare. Däremot bryr de sig inte så mycket om feedbacken som förklarar varför deras svar var rätt eller fel, vilket är applikationsmodellens bärande budskap. De märker heller inte att deras svar resulterar i några kunskapspoäng. Kontentan är att testpersonerna blir lite förvånade när alla frågor är besvarade och resultatet presenteras för dem, och att de då har en helt annan uppfattning om hur det har gått jämfört med vad som presenteras. Därför inför vi en rad viktiga justeringar i frågemomentet. Vi strukturerar om frågorna och dess feedback, så att informationen på om svaret är rätt eller fel kommer först, tillsammans med instruktionen för hur man ska gå vidare. Därefter kommer feedbacken på varför det är rätt eller fel. De plus- eller minuspoäng som erhålls efter att en fråga besvarats kommer, väljer vi att visa tydligt genom att de tonas fram ur frågan för att sedan ”åka” ner i kunskapsmätaren. På så sett ser användaren att något händer och att deras handlande har någon effekt, vilket förhoppningsvis ger dem en anledning att svara rätt på frågorna.

Två av våra respondenter efterfrågar länkar, eller motsvarande, som leder till en extern källa där man kan inhämta mer information om ett ämne. Detta eftersom den information som finns i applikationsmodellen är ytterst koncentrerad, och att det därför kan vara intressant att få läsa mer på egen hand för dem som är intresserade. För detta ändamål lägger vi till ett par länkar både till webbsidor och till dokumentfiler. Vi väljer att göra detta för att visa hur det kan se ut och fungera med länkar, vart de leder och vilket innehåll som visas prioriterar vi dock inte.

6 Resultat

Målet med examensarbetet var att ta fram en applikationsmodell för att synliggöra företagsflöden och varför dessa ser ut som de gör. Innehållet skulle vara företagsspecifikt, medan utseende och funktion var tänkt att vara generell. Arbetet med att ta fram denna applikationsmodell har angränsat till teoretiska områden för pedagogik och systemarkitektur, och vi har under processen hunnit reflektera över många olika saker, som var för sig är hela vetenskaper och examensarbeten. E-learning är ett enormt stort område, likaså när det gäller företagsflöden och arbetsprocesser. Vi är långt ifrån fullärda inom detta, men känner samtidigt att vi fått en allt bättre inblick eftersom. Hade vi haft mer kunskap om e-learning från början är sannolikheten stor att applikationsmodellen hade fått en helt annan utformning.

Arbetet har dragit ut på tiden, ungefär sex månader har nu förflutit sedan vi var på Informator och träffade Henrik Thorsell för första gången. Mycket tid har gått åt till det omfattande förarbetet, men också till att sätta sig in i hur utvecklingsverktyget Adobe Flash fungerar. I takt med att vi lärt oss mer, har vi fått en klarare bild för hur vi skulle gå till väga, och arbetstakten och modet att våga göra större ändringar i applikationsmodellen har ökat.

Tidsplanen har fått omarbetas flera gånger, detta främst med anledning av vår iterativa arbetsmetod. I flera omgångar har vi implementerat och testat, för att sedan i stort sett förkasta en idé och börja om på nytt. Vi kunde troligen ha tjänat en hel del tid på att använda en vattenfallsmodell och då satt upp klara datum mål för de olika arbetsmomenten, men med tanke på vår något ringa ämneskunskap vid projektets start så kändes detta aldrig som något alternativ.

Allt startade med en grundlig förstudie, där vi nu med facit i hand lyckats göra bra analyser av målgruppen och de flöden vi hittat på Informator. Vi har insett att det finns flöden som är starkt knutna till ett företags verksamhet, men också att de flesta vardagliga händelser kan definieras som flöden. Genom att studera dessa olika typer av flöden har vi insett att de, trots sina principiella olikheter, ändå har en hel del gemensamma nämnare. Vi har då kunnat definiera dessa nämnare och ta fram en flödesstruktur, där i princip vilket flöde som helst kan ritas upp så länge det följer våra definitioner. Flödesstrukturen har visat sig användbar och har inte behövt förändras efter att vi ansåg den som godkänd för sitt ändamål. Flödesstrukturen har sedan utgjort underlaget för skapandet av applikationsmodellen, vilken är det utbildningskoncept för e-learning som Informator efterfrågade.

Applikationsmodellen, som kan ses som den grafiska representationen av de flöden som är uppbyggda utifrån flödesstrukturen, innehåller enligt målgruppen en kombination av teori och spelmoment i lagom balans. Samtidigt är den snabb att komma igång med och kräver inte någon längre tid för att man ska kunna lära sig något, vilket efterfrågats. Ett bra tecken är även att hjälpfunktionen inte har behövt användas i någon större utsträckning. Efter de inledande redigeringarna har testpersonerna ansett applikationsmodellen vara självförklarande.

För att kunna presentera flödena och dess innehåll på ett intressant och lättförståeligt sätt så har vi arbetat med två saker; en metafor och en presentatör. Vi har tillämpat en metafor som består av en kombination av ett pussel och spelet Pipes. Denna metafor har sedan kompletterats av en presentatör, som kunnat förmedla feedback och kortare instruktioner på ett mer personligt sätt i dialogform. Både metaforen och presentatören har visat sig fungera bra och fått löpa som röda trådar genom hela applikationsmodellen. Innehållet har över lag upplevts som intressant.

Applikationsmodellen uppfyller huvudsyftet att kunna lära ut varför flöden ser ut som de gör. Detta under förutsättning att användarna verkligen engagerar sig och tar till sig den information som ges. Mycket av innehållet är textbaserat, och man måste ha ett engagemang och vara fokuserad på det man läser för att lära sig något. De som inte tar sig tid att sätta sig in i materialet, förstår troligen inte innebörden av innehållet, och lär sig därmed inte heller det väsentliga. Applikationsmodellen är inte på något sätt perfekt och det finns många förbättringar som skulle kunna göras, framförallt när det gäller att underlätta för användarna att ta till sig information. Frågan är då hur man kan förbättra detta, utan att behöva skära ner på omfattningen av det som finns att förmedla. Kanske är svaret att använda ljud och video, varvid informationen i större utsträckning kunde ha förmedlats på ett lättare sätt.

Det som vi tillför med detta arbete kan sammanfattas som; en struktur och komponenter för att kunna presentera olika flöden på ett enhetligt sätt, en lämplig metafor för att kunna presentera flödesinformationen på ett intressant sätt, en modell och ett koncept för att visa hur man kan använda kombinationen av strukturen och metaforen i utbildningssyfte. Vad vi har kommit fram till är att applikationsmodellen och framförallt den bakomliggande strukturen mycket väl skulle kunna användas för det tilltänkta syftet. Mer tid skulle dock behöva läggas på själva applikationsupplägget och tänkandet kring e-learning och pedagogiska aspekter. I stort sett vilket flöde det än gäller, så kan vår applikationsmodell rent teoretiskt användas för att både visa och förklara dess innebörd. Det handlar bara om att förändra hur information ska presenteras användaren.

7 Slutsatser och diskussion

Tanken med e-learning i företagssyfte är att de anställda i princip ska kunna läsa in en kurs på sin egen kontorsdator, samtidigt som de sitter med sitt vanliga arbete. Trots att e-learning som utbildningsområde har utvecklats enormt de senaste åren och att variationen på applikationer nu är stor, så kvarstår kärnproblemet; att få människor som initialt är omotiverade att genomföra en kurs, att faktiskt bli intresserade och kunna lära sig något nytt.

Det finns många slutsatser att dra av vårt arbete, och då kanske framförallt den senare delen som har med applikationsmodellen att göra. Vi har tagit fram vår applikationsmodell i tron på att användarna generellt sett är intresserade och engagerade, vilket förmodligen inte är fallet i verkligheten. Användarna ute på företag är troligen inte villiga att sätta sig ner och lära sig något nytt, parallellt med deras andra arbetsuppgifter, i den omfattningen som vi utgått från. Det egna engagemanget, kombinerat med ett upplägg som så lätt som möjligt presenterar information, är vad som krävs, och det är själva upplägget som vi kanske inte lyckats så bra med. Vi märker exempelvis att testpersonerna först upplever att applikationsmodellen känns lätt och enkel, för att sedan bli lite smått chockerade när de upptäcker att det finns mycket text och fakta som det är tänkt att de ska ta till sig. Det roliga och mer lekfulla utforskandet övergår till något mer allvarligt då de måste sätta sig in i information och verkligen tänka till. Slutsatsen är att den stora mängden information som användaren ska ta till sig, i kombination med ett från början bristande engagemang, medför att engagemanget blir ännu mindre och därmed också användarens vilja att ta till sig information. Allt resulterar i en ond cirkel där användarens engagemang och mottagningsförmåga ständigt minskar, varvid risken ökar för att användaren missar viktiga instruktioner och information som gör att man stegvis ska kunna uppnå en djupare förståelse.

Vi har också märkt att det är högst personligt hur ett flöde ser ut. Under förstudien analyserade vi flöden utifrån de intervjuer vi gjorde med de anställda. Vi skisserade flödena och bad Henrik Thorsell att titta på dem, för att på så vis se om vi gjort korrekta tolkningar. När vi sedan gjorde sluttestningen visade det sig att flera av testpersonerna hade synpunkter på hur de implementerade flödena var uppbyggda. Även om detta visar på att de med hjälp av applikationsmodellen kunde visualisera flödet, så visar det samtidigt på att de inte delar företagets uppställda riktlinjer för hur arbetet är tänkt att utföras. Motståndet att ta till sig informationen ökar då de anser att deras eget arbetssätt är bättre än det som presenteras genom de olika flödena. En slutsats som vi drar av detta är att e-learning som inlärningsforum, vilket i hög grad innefattar vår applikationsmodell, brister när det kommer till

diskussionsstadiet där information ska tolkas av användaren. Begränsningen ligger i att användarens motpart är ett system/applikation och inte någon fysisk person som kan argumentera och diskutera. Det vi vill uppnå med applikationsmodellen är att användarna ska få en djupare förståelse och börja reflektera över vad de gör. I samma stund som de gör detta kommer de dock också att börja ifrågasätta den information som de presenteras för, och i den stunden har han/hon ingen att diskutera med.

En intressant reflektion är också att användare som har en hög datorvana förmodligen har lättare för att lära sig något av en e-learningkurs, främst för att de redan är vana vid att ta till sig information på elektronisk väg. De som är mindre datorvana kan ha svårare att lära sig något, eftersom de i första hand måste fokusera på funktioner, snarare än innehåll. De måste dels lära sig att hantera datorn och applikationsmodellen, dels lära sig innehållet som förmedlas. Det blir alltså ett extra moment som kan göra en e-learningapplikation som vår mindre effektiv för dem med låg datorvana.

För att uppnå effektmålet, att kunna förbättra verksamheten, så bör man nog fördela inlärningsstegen i olika nivåer: en grundnivå och en avancerad nivå. En nivå för de som ska lära sig grunderna och få en ytligare förståelse, och en för de som är mycket insatta i en viss process sedan tidigare. En nivå med syfte att lära och en nivå för att tänka djupare och kunna förbättra och effektivisera. Vi anser att bara de som är insatta i grunderna och har viljan att ta till sig informationen kan uppnå en djupare förståelse för ett flöde och dess innehåll. Ibland måste information få presenteras på ett svårt sätt, och passar det då inte alla användare så får man utveckla applikationen därefter. Exempelvis genom att bygga in olika nivåer, så att var och en kan ta till sig information på det sätt som passar dem.

Som ett råd inför framtiden skulle vi vilja att man tänker sig för en extra gång när man skapar en e-learningapplikation. En varningsklocka bör ringa i huvudet när man vill utveckla en applikation där användaren är tänkt att uppnå en djupare förståelse, genom eget tänkande och tolkande av den information som presenteras. Vi anser att e-learning som medietyp lämpar sig bättre när det gäller fakta som kan presenteras svart på vitt, och som därmed inte lämnar så mycket utrymme för användaren att tolka information. För när de väl börjar göra detta uppstår en diskussionssituation, och vem ska de då diskutera med? Det som finns att tillgå är systemet, som inte kan kommunicera med användaren på det sätt som krävs. Användarens misstroende mot informationen som presenteras leder till ett bristande engagemang, vilket sedan leder till ytterligare misstroende och bristande engagemang. Till slut kommer man till en punkt där hela applikationskonceptet tappat sin poäng, att

användaren ska bli intresserad och själv tolka information för att uppnå en djupare förståelse.

7.1 Vidareutveckling och rekommendationer

Eftersom användarna, på grund av den stora textmängden, har haft vissa problem med att ta till sig informationen, bör man åtgärda detta på något sätt. Att använda ljud och video mer ingående i applikationsmodellen medför att textmängden troligen skulle kunna minskas ner, vilket i sin tur gör att användarna lättare tar till sig informationen. Detta kan sedan kombineras med att den tecknade presentatören görs mer levande, genom animationer och en egen röst. Då skulle dialogen med användaren förstärkas ytterligare, och användaren skulle därmed inte behöva anstränga sig på samma sätt för att få den feedback som den behöver.

Det poängsystem som finns implementerat i applikationsmodellen skulle kunna utvecklas och få en mer genomgående roll i applikationsmodellen, förslagsvis genom någon mer övergripande poängpanel än den kunskapsmätare som finns i nuläget. Om användarna redan från början skulle se att det finns poäng att inhämta och möjlighet att nå nya nivåer, skulle de kanske engagera sig mer.

Om man väljer att bibehålla mängden information i en framtida applikation, skulle man kunna tänka sig att implementera någon form av anteckningsfunktion för att underlätta för användaren. Med denna funktion skulle användaren själv kunna skriva in egna kommentarer, eller lägga till egna bilder till ett flöde och dess olika delar, för att skapa ytterligare engagemang och underlätta inläringen.

Det grafiska gränssnittet skulle kunna vara dynamiskt utbytbart så att man på ett enkelt sätt skulle kunna anpassa hela applikationen efter en kunds egen grafiska profil (färgschema, typsnitt etc.).

Det finns en komponent som är definierad i vår flödesstruktur, men som inte finns implementerad i vår applikationsmodell – delflödeskomponenten. I en vidareutveckling av applikationsmodellen kan man tänka sig att använda denna komponent för att skapa olika nivåer av ett flöde. Eftersom flertalet av beståndsdelarna i ett flöde i sin tur består av egna flöden, kommer denna komponent väl till pass. Man skulle exempelvis kunna tänka sig att när man placerar ut denna bit i flödet, får upp en dialogruta med ett ”miniflöde” som ska läggas för sig. När detta är gjort återgår man automatiskt till huvudflödet.

Viktigt att tänka på om man vill lägga in nya flöden är användningen av processblock. Man bör begränsa antalet för att ett flöde inte ska bestå av för få fasta punkter. Detta för att användarna ska kunna förstå hur flödet är uppbyggt. Skulle man kunna placera ut alla bitar i vilken ordning som helst, så hade alla lösningar varit riktiga, varvid det inte finns något att lära ut.

Att utveckla en databas, där flödets utseende, delar, antal komponenter, frågor o.s.v. kan sparas, hade inneburit att man fått en betydligt mer flexibel applikation. Man skulle kunna fylla databasen med information, som sedan används i ett användargränssnitt för att rita ut flöden och fylla dessa med innehåll.

En sista sak är att effektivisera applikationsmodellen programmeringsmässigt genom att använda objektorienterad programmering med ActionScript 3.0 och XML¹⁰. Exempel på användningsområde kunde vara att skapa en mer dynamisk meny, där man endast behöver ändra i XML-filen för att ändra innehållet. På så sätt kan man underlätta administrationen då man lägger till fler flöden.

7.1.1 Administrationsgränssnitt

Under arbetet gång har det funnits tankar på ett administrationsgränssnitt i någon form. I administrationsgränssnittet skulle man på ett enkelt och smidigt sätt kunna lägga in fler flöden och användare, vilket kopplas till den befintliga databasen, och på så vis förenkla hela applikationsuppdateringen.

Upplägget för ett administrationsgränssnitt kunde vara att man vid uppdatering presenteras för en tom spelyta, på vilken man sedan placerar ut komponenter från en lista och kopplar samman dessa med vägdelar från en annan lista. Därefter får man lägga till informationen som ska finnas för bitarna både vad gällande hjälprutor, feedback och frågor. Slutligen anpassas spelplanens storlek automatiskt efter antalet bitar som man valt att använda. Flödet namnges och sparas i en databas för att sedan bli tillgängligt i flödesmenyn.

Det finns dock en viktig frågeställning som dyker upp i samband med detta; ska applikationen vara en gratistjänst på Internet, eller en applikation som det utvecklande företaget säljer till sina kunder? En intressant tanke med en gratistjänst är att man skulle kunna skapa egna flöden, som man sedan skickar till sina kollegor och som de kan utforska. En framtida applikation som vi ser det måste dock, liksom vår nuvarande modell, innehålla mycket information. Förmodligen är det dock så att en gratistjänst då tyvärr inte kan innehålla de moment som vår applikationsmodell utgör i nuläget. Det är nämligen knappast

¹⁰ XML – eXtensible Markup Language. Ett regelverk utvecklat av W3C för att skapa metaspråk.

troligt att någon skulle ha tid eller lust att på eget bevåg lägga in den information som krävs. Med andra ord skulle en gratisversion behöva bestå av andra moment än de som vi implementerat.

7.1.2 Andra användningsområden

Eftersom vi upptäckt att de flesta vardagliga situationer kan åskådliggöras som flöden, ställer man sig frågan om dessa skulle kunna följa vår flödesstruktur och representeras med applikationsmodellen. Detta anser vi vara fullt möjligt och man skulle exempelvis kunna använda applikationsmodellen till att visa hur man monterar en bokhylla, hur pengaflödet på en myndighet fungerar eller hur man använder en funktion i ett datorprogram. När det gäller möbelmontering så kräver detta kanske lite mer grafik och mindre text, men det skulle helt klart vara fullt möjligt att presentera detta med nuvarande applikationsmodell.

Detta föranleder också en sista diskussion om man kan tänka sig en helt annan målgrupp för en framtida applikation, än den vi har riktat oss till. Kanske är det så att konsumenter som köper bokhyllor har lika mycket nytta av applikationen som de anställda på Informator, som ska lära sig hur affärsadministrationen fungerar. Vi anser att så är fallet.

Vi har hela tiden arbetet utifrån målsättningen att inte begränsa applikationsmodellen när det gäller utvecklingsmöjligheter. Detta anser vi oss nu heller inte ha gjort, utan förhoppningsvis har vi, genom våra lösningar och idéer, istället öppnat alla dörrar för en framtida skarp applikation.

8 Referenslista

8.1 Tryckta källor

- [1] Barfield, Lon (2003). Metaphors. Ur: *Design for new media (1. ed.)*. Addison Wesley. s. 302-310.
- [2] Bergström, Bo m.fl. (2003). Information och utbildning. Ur: *Webbdesign (2. ed.)*. Malmö: Liber Ekonomi. s. 116-117.
- [3] Boyle, Tom. (1997). Simulation and vicarious experience. Ur: *Design for Multimedia Learning*. London: Prentice Hall. s. 40.
- [4] Boyle, Tom. (1997). Strategic views of evaluation. Ur: *Design for Multimedia Learning*. London: Prentice Hall. s. 200.
- [5] Gärdenfors, Peter (2001). Att förstå är att se ett mönster, *IT i skolan - mirakelmedicin eller sockerpiller*. Rapport 45/2001. Stockholm: IT-kommissionen.
- [6] Molich, Rolf. (2002). Var hjälpsam när det uppstår problem. Ur: *Webbdesign – med fokus användbarhet (2. ed.)*. Lund: Studentlitteratur. s. 81.
- [7] Molich, Rolf. (2002). Härma mänsklig dialog. Ur: *Webbdesign – med fokus användbarhet (2. ed.)*. Lund: Studentlitteratur. s. 87.
- [8] Molich, Rolf. (2002). Tänka högt-tester. Ur: *Webbdesign – med fokus användbarhet (2. ed.)*. Lund: Studentlitteratur. s. 149.
- [9] Mårdsjö, Karin & Carlshamre, Pär (2000). Skapa kontakt med användarna. Ur: *Retoriken kring tekniken*. Lund: Studentlitteratur. s. 81-82.
- [10] Patel, Runa & Davidson, Bo (2003). Undersökningens uppläggning. Ur: *Forskningsmetodikens grunder – att planera, genomföra och rapportera en undersökning (3. ed.)*. Lund: Studentlitteratur. s. 54-55.
- [11] Patel, Runa & Davidson, Bo (2003). Frågornas grad av standardisering och strukturering. Ur: *Forskningsmetodikens grunder – att planera, genomföra och rapportera en undersökning (3. ed.)*. Lund: Studentlitteratur. s. 72.

[12] Patel, Runa & Davidson, Bo (2003). Observation. Ur: *Forskningsmetodikens grunder – att planera, genomföra och rapportera en undersökning* (3. ed.). Lund: Studentlitteratur. s. 87-88.

[13] Patel, Runa & Davidson, Bo (2003). Kvalitativ bearbetning. Ur: *Forskningsmetodikens grunder – att planera, genomföra och rapportera en undersökning* (3. ed.). Lund: Studentlitteratur. s. 118.

[14] Schank, Roger (1997). *Virtual Learning, a Revolutionary Approach to Building a Highly Skilled Workforce*. New York: McGraw-Hill Companies.

[15] Watkins, Ryan (2005). Developing Interactive E-learning Activities. *Performance Improvement*, 44(5), s. 5-7

8.2 Elektroniska källor

[16] Energy Northwest

URL: <http://www.energy-northwest.com/images/gen/igcc/CleaningProcess.gif>
(2007-07-12)

[17] Informator Utbildning Svenska AB

URL: http://www.informator.se/om_informator.aspx (2007-06-18)

[18] Institutet för kvalitetsutveckling

URL: <http://www.siq.se/page.asp?id=17> (2007-06-13)

[19] Svenska datatermgruppen

URL: <http://www.nada.kth.se/dataterm/fkt.html#d42> (2007-07-12)

Bilaga 1 – Enkät till anställda vid Informator

Sida 1(5)

Examensenkät - Informator AB

Vi är två studenter från Lunds Tekniska Högskola som gör vårt examensarbete på Informator i Malmö. Som en del i detta behöver vi få in underlag inför skapandet av en e-learningapplikation.

Vi är därför väldigt tacksamma om ni hade velat ta 4-5 minuter av er tid och fylla i formuläret på kommande sida. Formuläret handlar om er arbetssituation och erfarenheter av e-learning.

Om det är en fråga ni inte vill eller kan svara på, går det bra att lämna detta fält blankt.

OBS! Svaren är helt anonyma.

Ev. frågor kan besvaras av [Fredrik Jacobsson](#) eller [Michael Persson](#).

[Gå vidare >>](#)

Examensenkät Informator AB | © [Fredrik Jacobsson](#), [Michael Persson](#)

Sida 2(5)

Allmänt

Kön:

Ålder:

Arbetsort:

Befattning:

Antal år på denna befattning:

Exempel på arbetsuppgifter:

[Gå vidare ->](#)

Examensenkät Informator AB | © [Fredrik Jacobsson](#), [Michael Persson](#)

Datorrelaterat

Hur gammal är din arbetsdator?

Vilket operativsystem använder du (oftast) på din arbetsdator?

Vilken webbläsare använder du (oftast) på din arbetsdator?

Vilka program använder du (oftast) på din arbetsdator?

Vilken upplösning använder du på din arbetsdator? x

På en skala från ett till fem, där fem är mycket bra, hur skulle du gradera din allmänna datorvana?

- 1
- 2
- 3
- 4
- 5

Har du möjlighet att spela upp ljud på din arbetsdator?

- Ja
- Nej

[Gå vidare ->](#)

E-learningerfarenheter

Har du någon tidigare erfarenhet av e-learning?

- Ja
Nej

Om du inte tidigare har erfarenheter, beskriv då vad du tror dig veta om e-learning.

Om du har tidigare erfarenhet, i vilket sammanhang?

- Skapande/utveckling
Egen inläring

Om du har tidigare erfarenhet, i vilken miljö?

- På arbetsplatsen
I hemmet eller annan plats

Om du har tidigare erfarenheter, berätta gärna lite mer om dessa. Exempelvis vad det gick ut på, vad som var bra eller dåligt, osv.

Skicka formulär

Tack!

Svaren från enkäten har nu skickats och vi är tacksamma för att du tog dig tid att svara på frågorna.

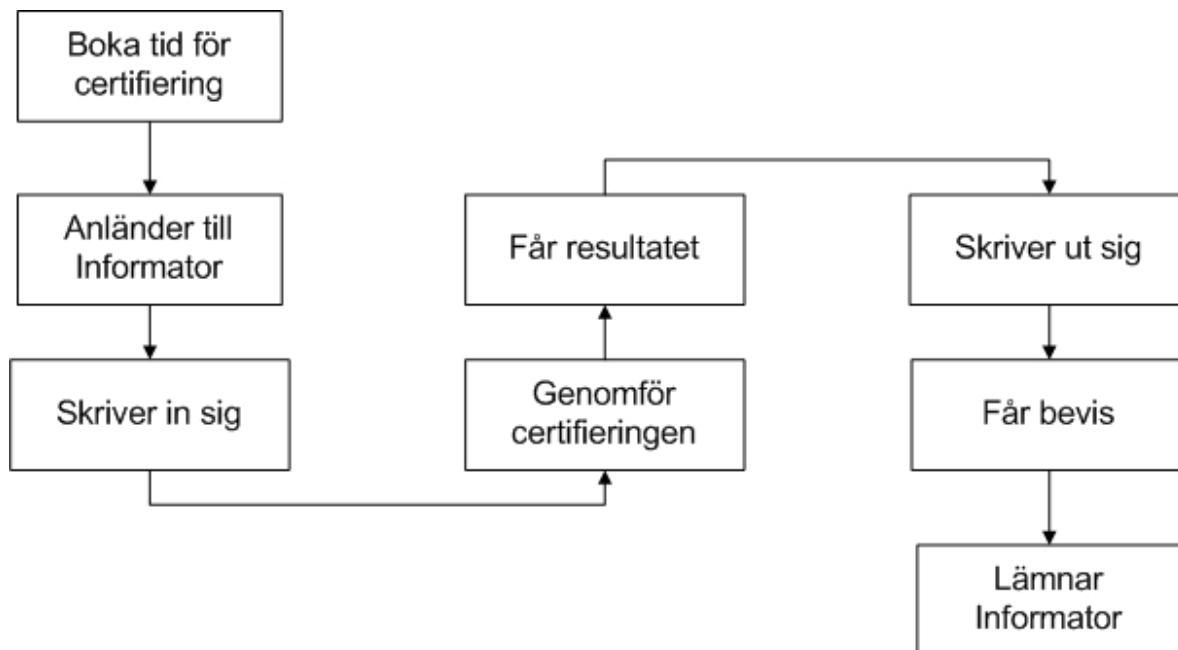
Bilaga 2 – Enkätssammanställning

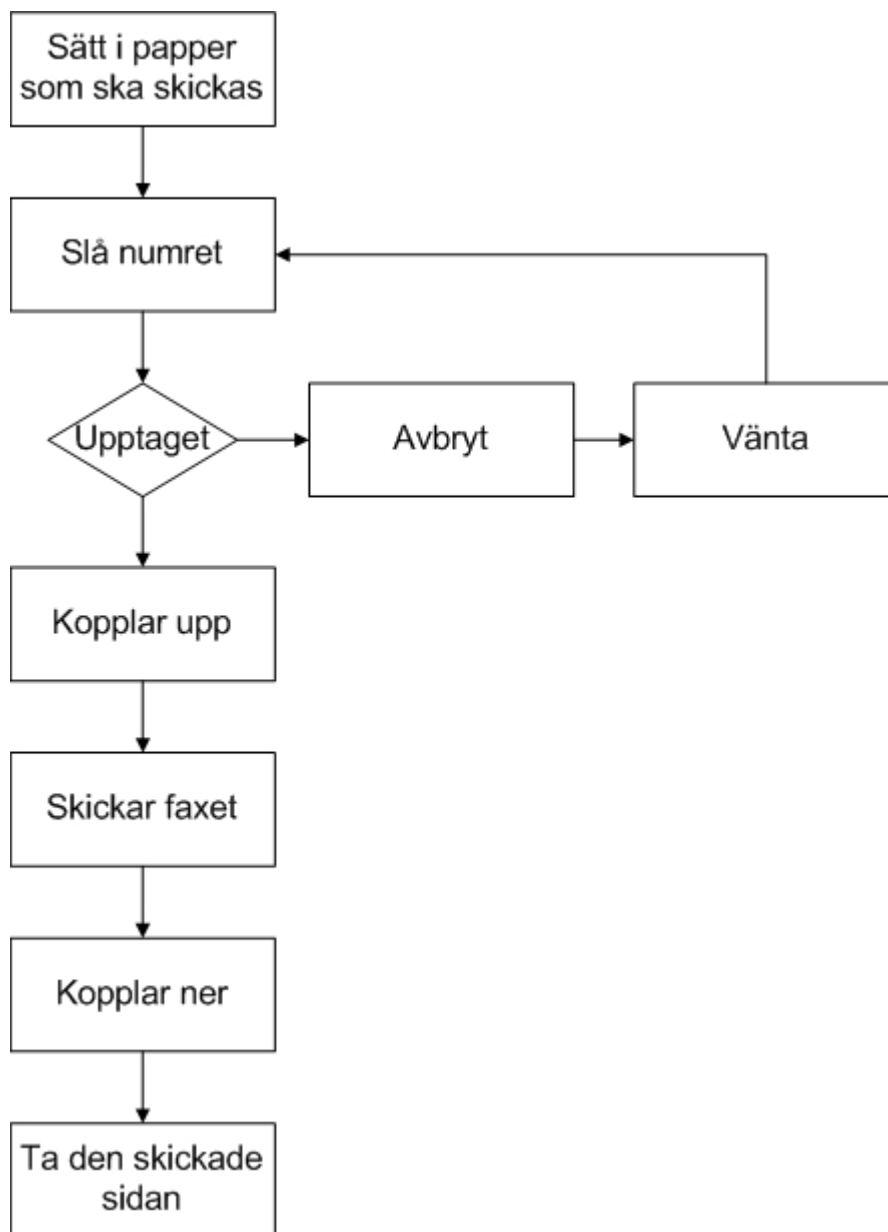
Kön	Ålder	Arbetsort	Befattning	Antal år på befattningen	Arbetsuppgifter
M	36	Göteborg	Key Account manager	3	Projektleddning, kundintervjuer inför utb.projekt
M	39	Stockholm	Key Account manager	10	Försäljning, projekt, analys och marknadsföring
M	35	Stockholm	Produktutvecklare	2	Skapa kurser
K	40	Stockholm	Säljadministratör	0	Stödjer utesäljare med admin och fortsatt praktiskt arbete + fortsatt kundkontakt
M	39	Göteborg	Säljare	3,5	Försäljning mot IT-företag vad gäller avancerad IT
K	29	Stockholm	Marknadsassistent	3	Webbinnehållsuppdateringar, event, seminariearrangemang, trycksaksprodukt
K	29	Stockholm	Administratör	7	Beställa böcker
M	37	Stockholm	Tekniker	5	Ansvar för telefoni och drift av växel. Installation på datorer inför kurs. Support
K	39	Stockholm	Reception/kursmateriel	0,5	Boka resa och logi åt lärare/personal, svara i telefon, beställa kursmaterial...
M	25	Stockholm	Controller	0	Fakturering, diverse lönsamhets/kostnadsutredningar, bonusberäkningar etc.
K	33	Stockholm	Administratör	10	Utdrag av deltagarlistor, beställning av mat, sammanställare av kursutvärdering
K	33	Stockholm	Assistent	0	
M	53	Stockholm	Säljare	7	Säljadmin och kundbesök
M	35	Malmö	E-learning producent	5	Produktion, programmering, projektledning av e-learning.
M	41	Malmö	Systemutvecklare	7	Webbutv. Flash, asp, databasdesign
M	37	Stockholm	Tekniker	5	Kursinstallationer

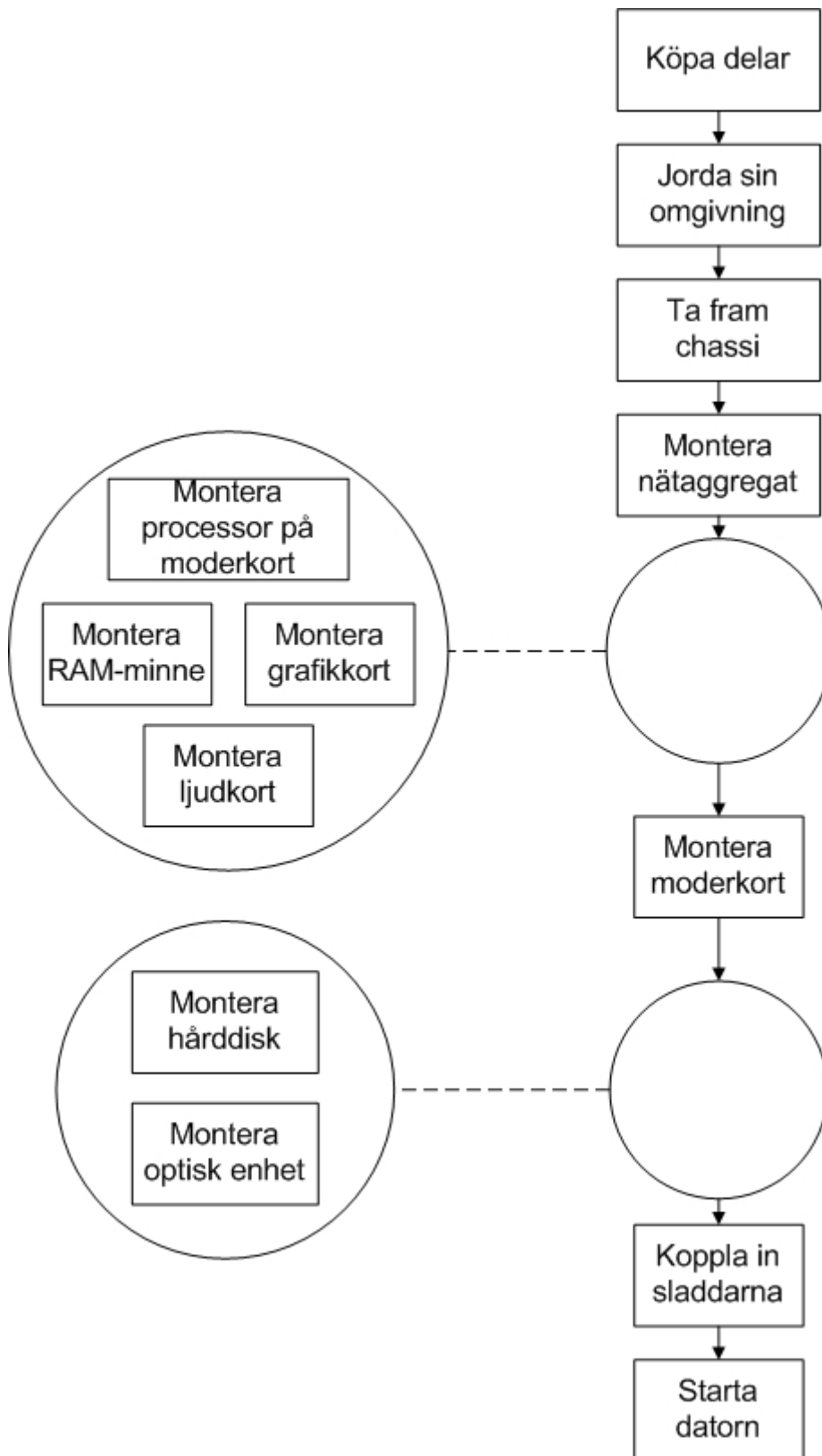
Datorns ålder	Operativsystem	Webbläsare	Program som ofta används	Upplösning	Datorvan	Tillgång till Jjud?
1	Windows XP	IE7	Outlook, IE, CRN, MSN	1280x1024	5	ja
0,4		IE	Excel, Powerpoint		3	ja
1	Windows XP	IE7	Excel, Word, IE	1024x768	5	ja
	Windows XP	IE	Abalon, Office-paketet		4	ja
0,5	Windows 2003 Ser		Excel, Word	1280x1024	4	ja
1	Windows 2003 Ser	IE	Office-paketet, Adobe-produkter, CMC	1440x900	4	ja
1	Windows XP	IE7	Abalon, Outlook, Word, Excel, MSN, m.fl	1680x1050	5	ja
1	Windows XP	IE6	Excel, Outlook, Word, Acrobat, Abalon	1680x1050	4	ja
	39	IE	Excel, Word, Abalon	1280x1024	2	ja
3	Windows XP	IE	Abalon, Outlook, Excel, Maconomy	1680x1050	4	nej?
1		IE	Excel, Outlook, Word	1680x1050	4	ja
0	Windows XP	IE	Outlook, Abalon, IE, Word, Excel	1680x1050	5	ja
2	Windows XP	IE7	Flash, Photoshop	1280x1024	5	ja
1	Windows 2003 Ser	IE7	Flash 8, Homesite	1280x1024	5	ja
1	Windows XP	IE7	Excel	1400x900	3	ja

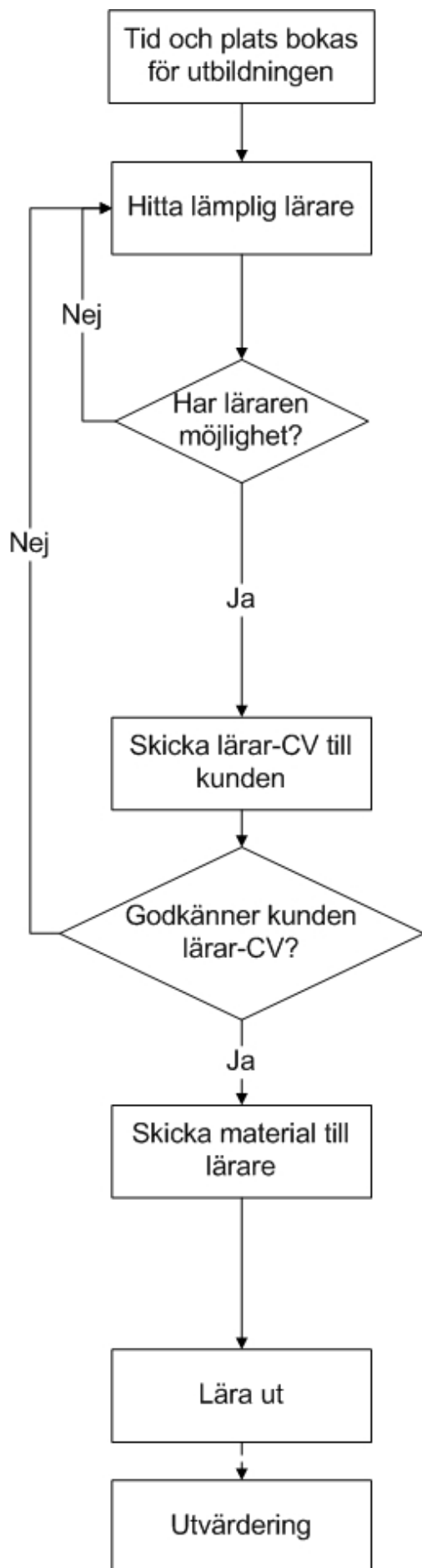
Tidigare erfarenhet	Om inte, vad tror du det är?	Vilket sammanhang?	Vilken miljö?	Utvärdering av erfarenheter
ja		utbildning	Hemma	ECDL-körkort - fungerade bra
nej	Enkelt att planera in utb. när tiden tillåter			
ja		Skapande/utv/utb	Arbete, hemma	Lära sig en applikation. Var primitivt, ser inte ut som ic
nej	Att gå en form av kurs som är internetbaserad			
nej				
nej				
nej	Utbildning helt digitalt. Ingen fysisk personkontakt			
nej	Man får uppg. och information via webben för att klara en utb.			
ja		Utbildning	På arbetsplatsen	Utb. I office 2003, var bra för kunde plugga i eget temp
nej	Interaktivt sätt att inhämta kunskap som sker på användarens villkor			
ja		Utbildning	På arbetsplatsen	Gick ut på att lära sig Excel bättre. Bortprioriterades tyv
ja		Utv/utb	På arbetsplatsen	Sålt och demonstrerat elearning, varit prog.expert i Ex
ja		Skapande/utv	På arbetsplatsen	
ja		Skapande/utv	På arbetsplatsen	Utveckling och testning av diverse webb/cd-applikation
nej	Skärmläsning och email			

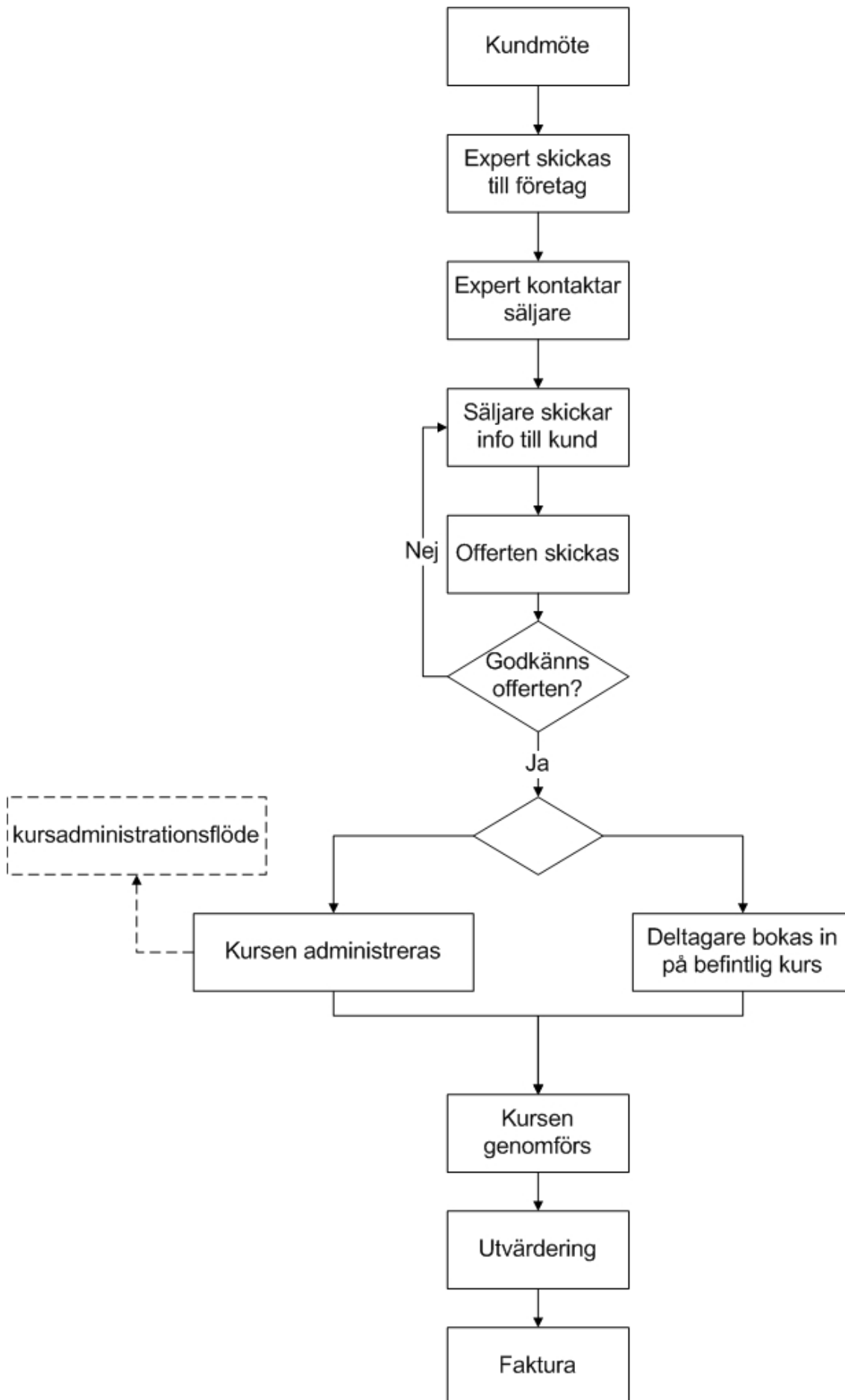
Bilaga 3 – Representativa flöden











Bilaga 4 – Presentationsmaterial för flödesstrukturen

Examensarbete LTH/Informator HT07 - Flödesapplikation

Flödesstrukturen

Start/Mål	Process	Processblock	Iterationsval	Vägval	Exempel

Processblock

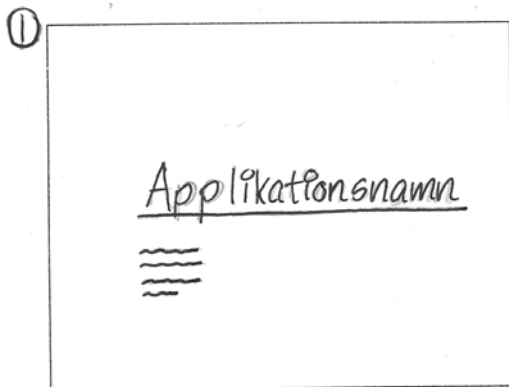
Minst en väg in, endast en väg ut. ▶■

Många olika delprocesser som ska sammanfogas och bilda en helhet.

Komponentens symbol har formen av en kvadrat, bestående av mindre kvadrater.

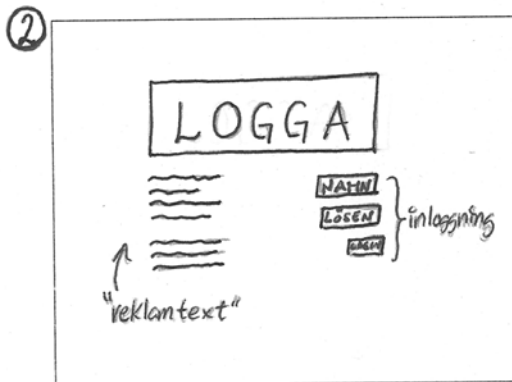
```
graph LR; Start([Start]) --> P1[Process]; P1 --> PB[Processblock]; PB --> I[Iterationsval]; I --> V[Vägval]; V --> P2[Process]; V --> P3[Process]; P2 --> P4[Process]; P3 --> P4; P4 --> P5[Process]; P5 --> Mål([Mål]);
```

Bilaga 5 – Storyboard



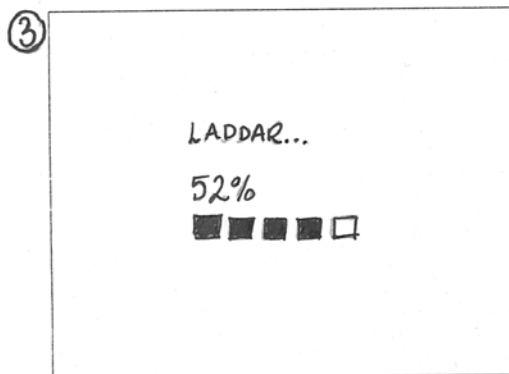
sida1_index.html

- Vanlig HTML-sida från vilken man öppnar applikationsfönstret (nytt fönster med fixerad storlek)



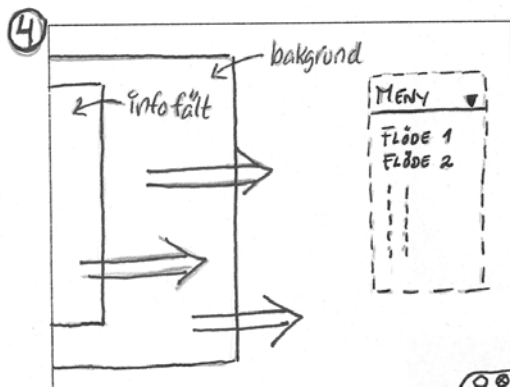
sida2_inlogg.swf

- Applikationsfönster innehållandes logga, kort inledande text om applikationen, inloggningsfält (namn+lösenord) med knapp.
- Textfeedback vid felaktig inmatning



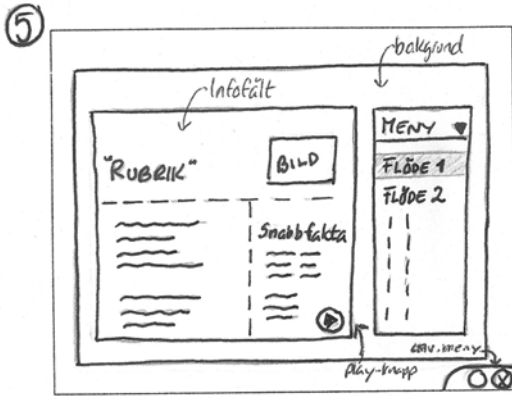
sida3_loader.swf

- Laddningssekvens som körs efter lyckad inloggning.
- Procenträknare och någon form av statusbar som uppdateras medan att huvudmenyn laddas in.



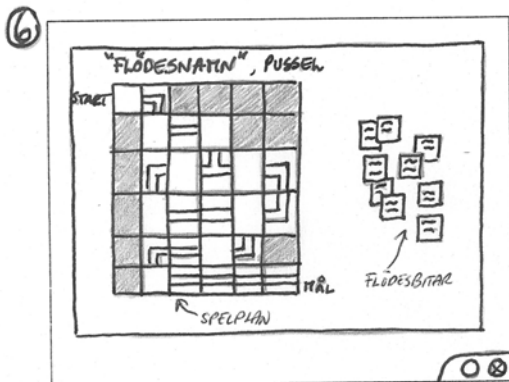
sida4_huvudmeny.swf

- Huvudmenyn animeras fram i en snabb sekvens (3-4 sekunder).
- Menyn animeras med "fade"
- Info fält och bakgrund (vit) för arbetsytan animeras in från vänster.



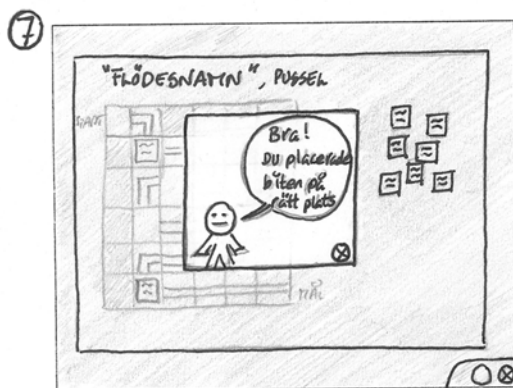
sida4_huvudmeny.swf

- Meny med samtliga flöden i alfabetisk ordning. Valt flöde markeras (färgändring)
- Infofält med rubrik och fakta om flödet. Kompletteras med relevant bild.
- Snabbfakta med "storlek/tid" för flödet.
- Anv.meny med stängknapp & ljud av/på.



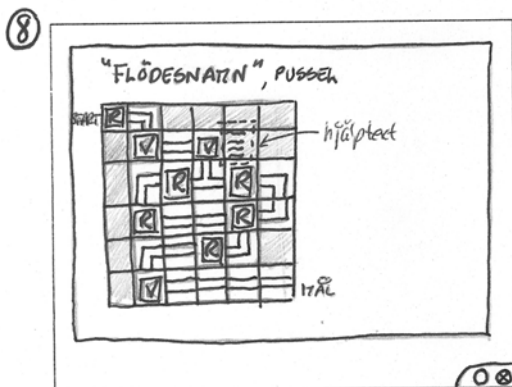
sida5_flode[nr].swf

- Spelplanen för pusselmomentet där det valda flödet ska färdigställas, genom att placera de utslumpade bitarna på rätt plats.
- Start, mål och "flödesvägen" utritad.



popup-feedback.swf

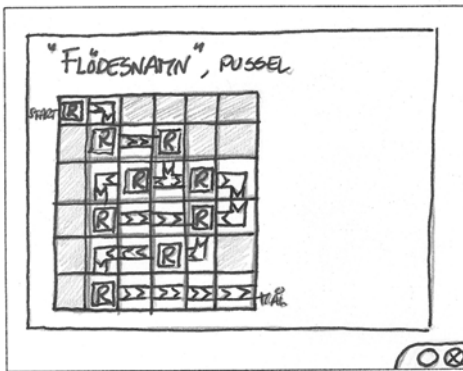
- Dialogruta som dyker upp när en flödesbit lagts på rätt/fel plats.
- Kort, konsis och informativ text.
- Dialogruta stängs via egen stängknapp
- Bakgrunds innehåll skuggat, ej klickbar.



sida5_flode[nr].swf

- Actionscript som lopar och kontrollerar om bitarna mellan start och mål är rätt placerade.
- Varje bit kompletteras med en hjälpruta (placerad bredvid) med information.

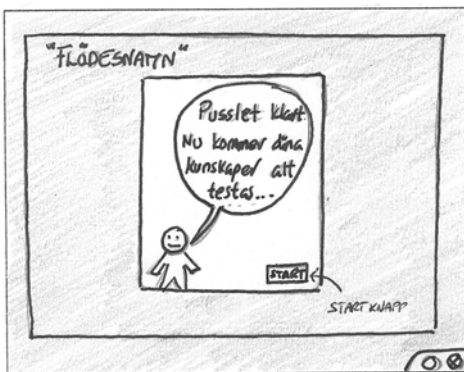
9



sida5_flode[nr].swf

- Lagda bitar färgmarkeras för att visa om de ligger rätt eller fel
- När alla bitar ligger på rätt plats körs en animering med pilar som rör sig från start till mål

10



popup-teststart.swf

- Dialogruta som introducerar nästa moment - förståelsetestet.
- Dialogruta stängs via egen stängknapp.
- Bakgrunds innehåll skuggat och ej klickbart

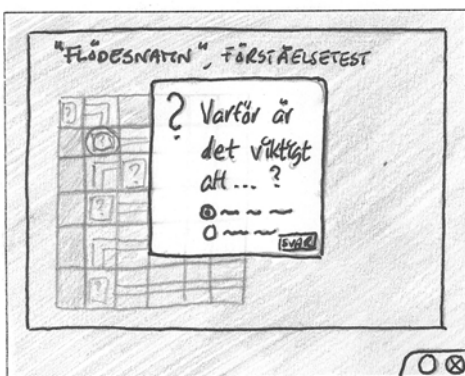
11



sida6_test [nr].swf

- Självttest där ett vattenflöde går från fråga till fråga, från start till mål.
- Vattenflödet är stoppat vid en flödesbit tills dennes fråga är besvarad
- Tydlig markering om aktuell frågebit.

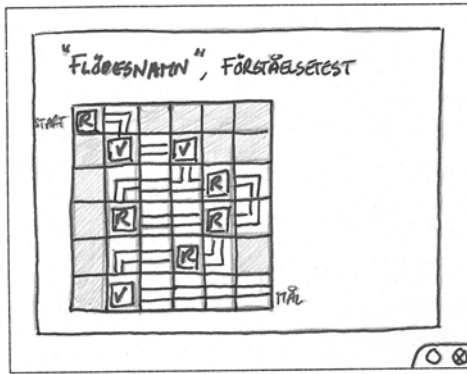
12



popup_testfraga.swf

- Dialogruta med fråga för den aktuella flödesbiten. Frågetyp; radio, check, textfield.
- Dialogruta stängs via egen stängknapp.
- Bakgrunds innehåll skuggat och ej klickbart.

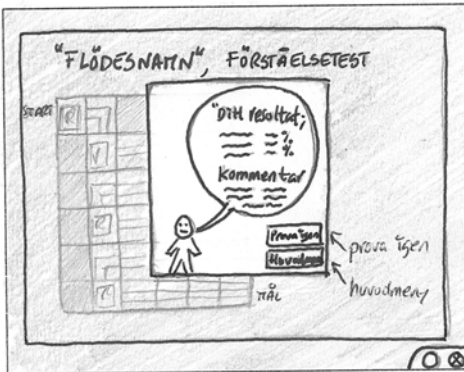
13



sid6_test [nr].swf

- Flödesbitarna färgmarkeras efter att en fråga besvarats, rätt eller fel (eller halvrikt)
- Vattenfud och pilar som animeras när man gått från start till mål.

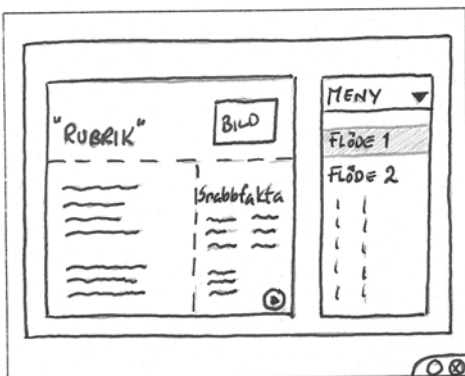
14



popup_testresultat.swf

- Dialogruta där resultatet av testet presenteras. Jämförelse med tidigare resultat, vilket hämtas från databas.
- Knappar för omstart och huvudmeny.
- Bakgrund skuggad och ej klickbar.

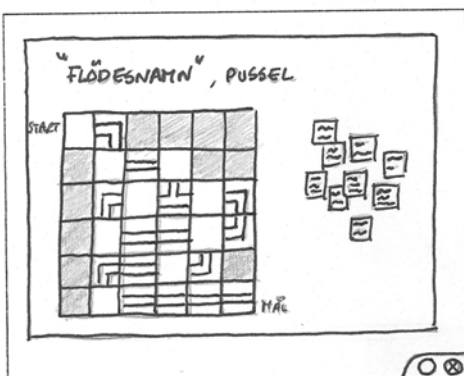
15:1



sid4_huvudmeny.swf

- Om man återgår till huvudmenyn; återgå till bildruta fem.
- Det valda flödet presenteras och snabbfaktan uppdateras med resultatet som erhålls.

15:2



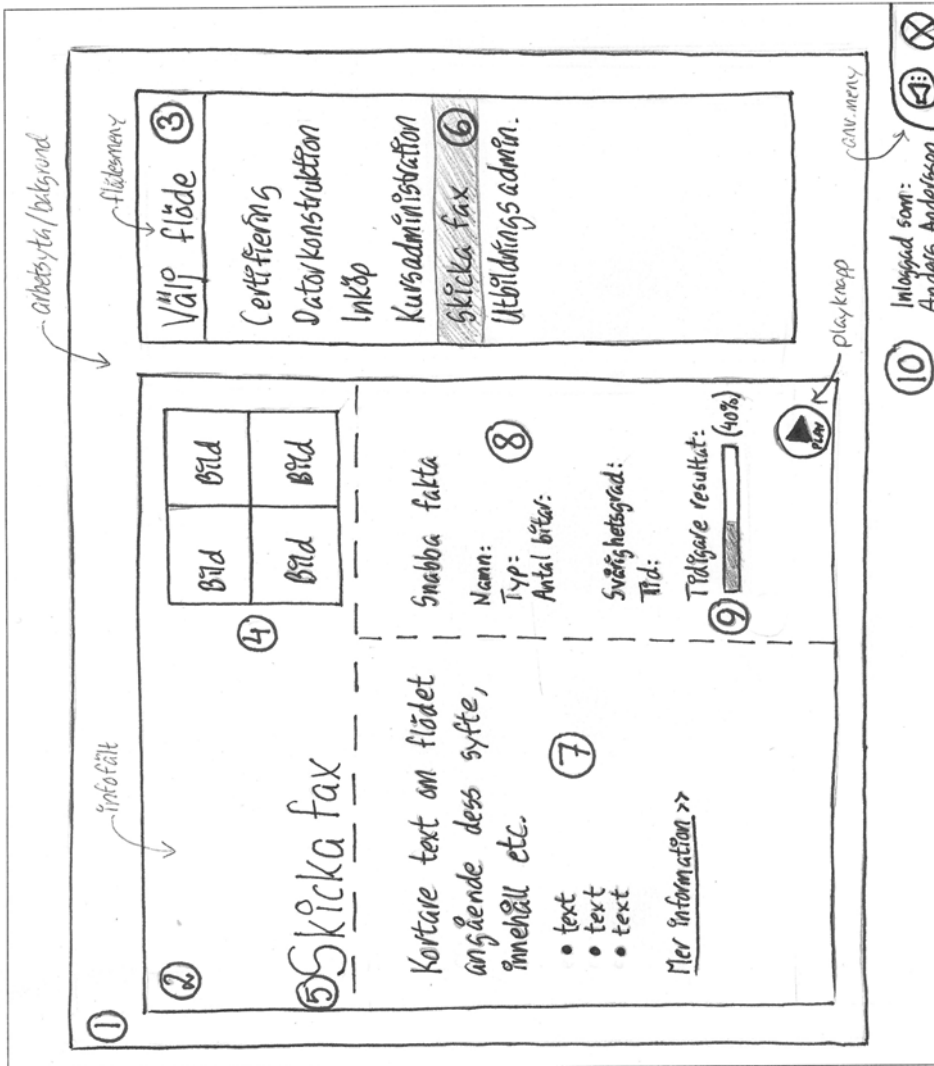
sid5_flode [nr].swf

- Om man väljer att pröva samma flöde igen; återgå till bildruta sex.
- Starta om pusslandet för att testa nya varianter och få ny feedback.

Bilaga 6 – Detaljskisser

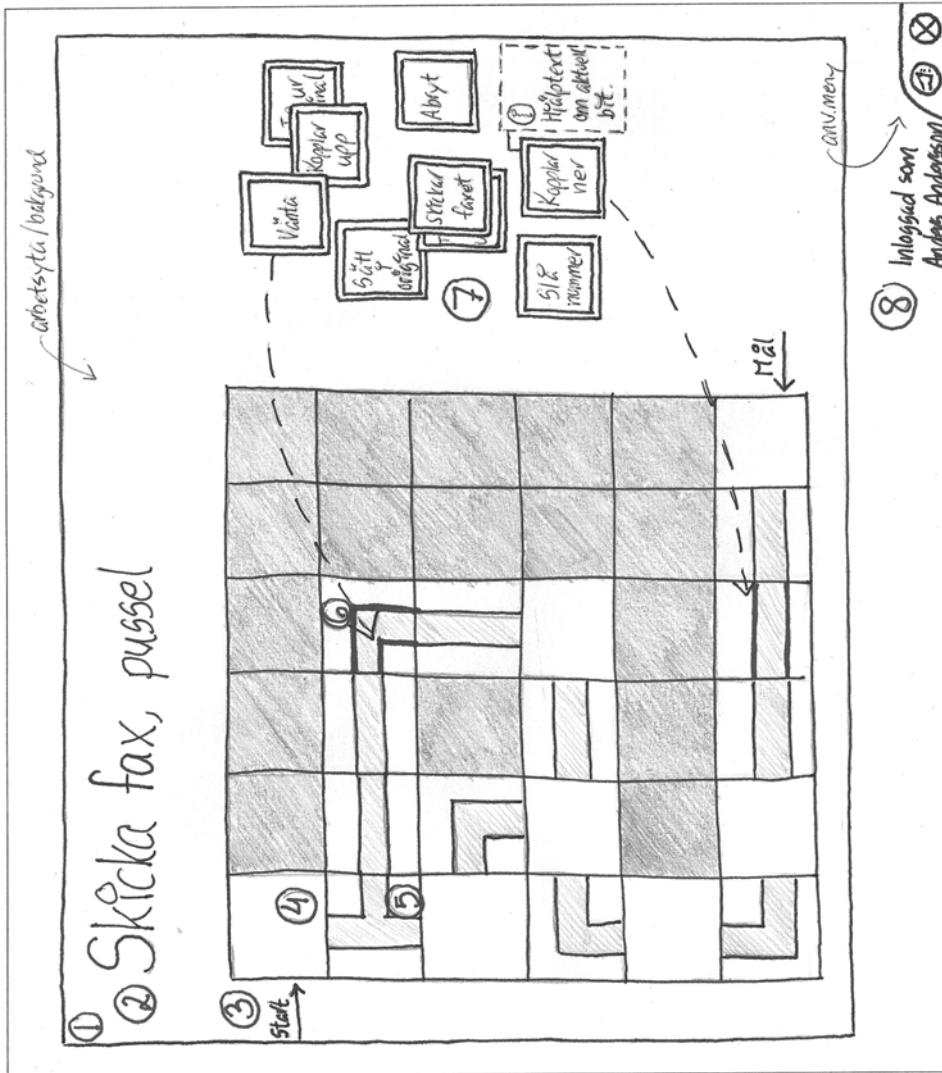
Sida 4 - huvudmeny.saf

- ① Vt/vis arbetsyta som följer med i applikationens olika moment.
- ② Informationsfält, 176 mörkare än alb-ytan
- ③ Meny med flöden, sorterade alfabetiskt
- ④ Det aktuella flödesvalet markeras med bakgrundsfärg, även någon hover-effekt.
- ⑤ Rubrik som kan mappas till menyvalet
- ⑥ Kort information om flödet, lättläst och luftigt i bra webbtypsnitt (Verdana)
- ⑦ Fakta för "genläsning" inför starten.
- ⑧ Animerad mätare för användarens resultat.
- ⑩ Anv.meny med stäng- och ljudknapp



Sida 5 flode [nr] swf

- ① Vit/ljus arbetsyta som följer med i applikationens olika moment.
- ② Rubrik för det aktuella flödet, följt av namnet på momentet (pussel).
- ③ Spelplanen för flödet, 6x6 bitar som upptar ca 60% av arbetsytan. Rutor utan någon väg/förändel är gråskuggade.
- ④ Tomma och upplysta rutor där bitar ska läggas
- ⑤ Fördittad väg/förändel mellan de skuggade delarna.
- ⑥ Väg som bildas om en bit placerats rätt
- ⑦ Flyttbara bitar med tillhörande hjälprutor
- ⑧ Anv.meny med stäng- och ljudknapp



Bilaga 7 – Resultat av scenariotestning

Respondent 1

- Läste all text noggrant på huvudmenyn. Tyckte det var bra med ”snabba fakta”.
- Hittade hjälpfunktionen och kunde enkelt hitta vad som efterfrågades.
- Loggade ut via Windows-krysset.
- Placerade ut flödesbitarna på rad för att skapa sig en överblick. Önskade att det funnits en knapp för att göra detta automatiskt.
- Insåg först inte att bitarna skulle placeras i de tomma rutorna. Läste därefter inte feedbacken och förstod inte om en bit låg på rätt plats. Blev frustrerad när bitarna inte hamnade rätt, och övergick till att testa sig fram på måfå.
- Tycker det är bra med feedback, men att det är för mycket text, framförallt i förståelsemomentet, att läsa och att denna inte presenterades på något tilltalande sätt.
- Ville ändra sitt svar då hon insåg att detta var felaktigt.

Respondent 2

- Loggade ut via Windows-krysset
- Provade hjälpfunktionen för att få reda på vilka bitar det var som skulle placeras ut. Tyckte sig inte få någon hjälp via hjälpfunktionen.
- Återgick till huvudmenyn genom att klicka på Windows-krysset (och loggade in på nytt).
- Föreslog en tydligare introducering över hur pusselmomentet fungerade då man inte förstår vilka som är flyttbara bitar.

Respondent 3

- För mycket informationstext om flödena på menysidan.
- Hittade direkt hjälpknappen, mappade till frågetecken-symbolen.
- Inga problem med att stänga hjälpen. Tyckte det var perfekt att den låg nere till höger då hon läste texten nedåt.
- Hittade direkt de flyttbara bitarna i förståelsemomentet.
- Såg direkt var de flyttbara bitarna skulle placeras på spelplanen.
- För mycket text i feedback/hjälprutan.
- Förstod hur frågemomentet gick till och tyckte upplägget var ”klockrent”.
- Ville besvara en fråga igen när det blev fel svar. Förstod heller inte när 0 poäng gavs, d.v.s. då svaret var nästan rätt.
- Såg inte feedbacken om varför det blev fel, bara att det blev så.
- Tyckte det var mycket bra med rörlig bild/video.