



Handelshögskolan  
VID GÖTEBORGS UNIVERSITET  
*Institutionen för informatik*  
2003-05-30

# Prototyping som systemutvecklingsmetod för småföretag

## Abstrakt:

Systemutveckling av mindre system för småföretag är ett område utan större uppmärksamhet från forskningsvärlden. Denna uppsats utvärderade systemutvecklingsmetoden Prototyping genom att med den utvecklade ett bokningssystem samt ett e-handelssystem för ett småföretag i Göteborg. Syftet var att skapa en rekommendation för systemutveckling av mindre system för småföretag. Arbetet var begränsat till att utveckla *två* system i *ett* företag med Prototyping. Genom intervjuer med användarna fick de efter utvecklingen av systemen berätta för oss hur de upplevt sin roll och arbetet enligt Prototyping. Prototyping visade sig fungera väl i vårt fall med undantag av några negativa aspekter. Hela projektets resultat var beroende av engagerade och motiverade användare. Användarnas roll var stor och utvecklingsarbetet kunde bli krävande för dem. Tidsåtgången för projektet var också svåruppskattad. Prototyping fångade emellertid upp utvecklarnas misstag och användarnas roll var mycket betydelsefull för utvecklaren, då användaren fungerade som en riktlinje för utvecklaren. Användarna lärde sig systemet samtidigt som de med sina krav och önskemål formade systemet som de ville ha det.

Författare: Ola Andersson  
Anette Stenborg

Handledare: Mathias Klang  
Magisteruppsats, 20 poäng

<b>1</b>	<b>INLEDNING</b> .....	<b>4</b>
<b>2</b>	<b>TEORETISK REFERENSRAM</b> .....	<b>6</b>
2.1	DEFINITION .....	6
2.2	OLIKA TYPER AV PROTOTYPING .....	7
2.3	FYRA STEG TILL ETT FÄRDIGT SYSTEM .....	8
2.4	FASER I PROTOTYPARBETET .....	9
<b>3</b>	<b>METOD</b> .....	<b>19</b>
3.1	VÅRT VAL AV METOD .....	19
3.2	VAL AV ÄMNE OCH OMRÅDE .....	21
3.3	LITTERATURSTUDIE.....	21
3.4	FALLSTUDIE MED PROTOTYPING .....	22
3.5	UTVÄRDERING AV PROTOTYPING.....	24
3.6	ANALYS OCH TOLKNING .....	25
3.7	VALIDITET OCH RELIABILITET .....	25
<b>4</b>	<b>FALLSTUDIE</b> .....	<b>27</b>
4.1	PROTOTYPEN FÖR WEBBOKNINGEN .....	27
4.2	PROTOTYPEN FÖR E-HANDEL .....	28
4.3	TEKNISK BESKRIVNING .....	30
4.4	ANVÄNDARSCENARIO .....	31
<b>5</b>	<b>RESULTAT</b> .....	<b>33</b>
5.1	LITTERATUR .....	33
5.2	INTERVJU MED FRISÖREN INNAN UTVECKLINGSARBETET .....	33
5.3	OBSERVATIONER .....	35
5.4	MODELLERING .....	36
5.5	PROTOTYP .....	41
5.6	INTERVJU MED FRISÖREN EFTER UTVECKLINGSARBETET .....	42
5.7	INTERVJU MED KUNDER EFTER UTVECKLINGSARBETET .....	43
5.8	SAMMANFATTNING .....	47
<b>6</b>	<b>DISKUSSION</b> .....	<b>49</b>
6.1	INTERVJUN MED FRISÖREN OCH OBSERVATION INNAN UTVECKLINGEN .....	49
6.2	MODELLERING .....	50
6.3	PROTOTYP .....	50
6.4	INTERVJU MED FRISÖREN OCH ANVÄNDARNA.....	51
6.5	PROTOTYPING INOM SMÅFÖRETAG .....	52
6.6	PROTOTYPING SOM SYSTEMUTVECKLINGSMETOD .....	53
6.7	PROTOTYPING I ETT VIDARE PERSPEKTIV.....	54
<b>7</b>	<b>SLUTSATS</b> .....	<b>56</b>
<b>8</b>	<b>REFERENSER</b> .....	<b>58</b>
8.1	LITTERATUR.....	58
8.2	INTERNET .....	59
8.3	ARTIKLAR .....	59

<b>9 BILAGOR.....</b>	<b>60</b>
9.1 BILAGA 1 – INTERVJU MED FRISÖREN INNAN UTVECKLINGSARBETET .....	60
9.2 BILAGA 2 – ENKÄTUNDERSÖKNING .....	62
9.3 BILAGA 3 – INTERVJU MED FRISÖREN EFTER UTVECKLINGSARBETET.....	64
9.4 BILAGA 4 – INTERVJU MED KUNDER EFTER UTVECKLINGSARBETE.....	66

# 1 Inledning

Systemutveckling i småföretag är ett område som är tämligen outforskat och med ytterst lite rekommendationer. Det står helt klart att systemutveckling i liten skala givetvis är lättare att utföra än att utveckla stora komplexa system. Det stora problemet vad gäller systemutveckling i mindre system i småföretag är med andra ord att det inte finns några riktlinjer på hur systemutvecklingen ska gå till. Oskarsson (1994) skriver dock om systemutveckling ”De omfattande handböckerna säger i allmänhet att små projekt får välja ut de delar av handböckerna som de behöver. Det fungerar inte, eftersom ansträngningen och arbetsinsatsen skulle vara för stor”. Med småföretag menar vi företag som har en till nio anställda (Dahmström 2000). Med mindre system menar vi system som inte innefattar fler klasser än tio och som inte har den komplexa och komplicerade strukturen som större system kan ha.

Småföretaget Klippotek Passagen kontaktade oss och önskade två olika system som båda skulle vara Internetbaserade. Det var ett bokningssystem och ett e-handelssystem som vi fick i uppgift att utveckla åt dem. Welling och Thomson (2001) skriver genom att Internet har expanderat kraftigt under de senaste åren har fler möjligheter utvecklats för att bland annat kunna ge ökad service till kunder. Allteftersom datorer blir var mans egendom så kommer också företag kunna använda Internet som hjälpmedel till sin verksamhet, genom att exempelvis sköta bokningar av tjänster eller göra inköp och beställningar av olika slag. Frisersalongen Klippotek Passagen vill utnyttja denna Internetvåg genom sin satsning att försöka konkurrera med liknande företag som redan satsat på bland annat bokning via Internet. Genom ett bokningssystem på Internet kommer kunden härmed att få service dygnet runt genom möjlighet att boka sin tid för behandling när kunden själv vill. Ett bokningssystem på Internet gör också att frisören slipper avbrott i arbetet genom telefonsamtal från kunder som vill boka tid. Dessutom önskar Klippotek Passagen försöka sänka sina lagerkostnader genom ett e-handelssystem på Internet. Kunderna kan där beställa de varor de önskar och på så sätt slipper frisersalongen köpa in varor som sen kunder inte visar sig vara intresserad av. Utbudet av varorna kommer också att öka eftersom alla varor som frisören genom grossister har möjlighet att köpa kommer visas på e-handelssidan.

Efter en tids studerande efter intressant systemutvecklingsmetod hittade vi Prototypingmetoden som verkade kunna fungera bra för systemutveckling av mindre system i småföretag. Welling och Thomson (2001) skriver också att webbaserade system passar bra för metoden Prototyping och med denna uppsats vill vi därmed testa denna teori. Systemutveckling i småföretag är vidare intressant eftersom de oftast inte har datorer som en naturlig del av sin kärnverksamhet. Många småföretag har oftast inte ens en IT avdelning. Prototyping kan för dessa företag fungera som en kvalitativ metod som ger en försäkran om att de kommer att få ett system som passar dem och som de kommer att förstå, eftersom Prototypingprocessen är färdig när användarna är nöjda.

Prototyping skiljer sig från de traditionella systemutvecklingsmetoderna, med vilka vi menar systemutvecklingsmetoder som innebär en lång och ofta komplicerad analys som komplement till programmeringen, på en mängd olika punkter. En av nackdelarna med traditionella systemutvecklingsmetoder är att hela utvecklingen bygger på en kravspecifikation. Blir kravspecifikationen fel kommer hela projektet att falla. Användarna har heller ingen naturlig roll i de traditionella systemutvecklingsmetoderna då hela arbetet lämnas åt systemutvecklarna. Utan användarna har systemutvecklarna ingen riktlinje förutom kravspecifikationen och användarna kommer först att kunna påverka systemet när det har blivit installerat hos beställaren. Detta innebär att användaren riskerar att få ett system som

den är missnöjd med. Prototypingmetoden innebär istället att användarna under hela utvecklingsprocessen är med och testat systemet. Här kan användaren komma med krav på ändringar och genom dessa styra utvecklingen. På detta sätt är det större chans att användaren blir fullt ut nöjd med systemet. (Ahlandsberg och Sandström 1988)

Syftet med vårt magisteruppsatsarbete är att utvärdera systemutvecklingsmetoden Prototyping genom att med den utveckla ett bokningssystem och ett e-handelssystem för småföretaget Klippotek-Passagen. Målet med denna magisteruppsats är, på grund av brist av tidigare forskning, att arbetet ska kunna ses som en rekommendation för systemutveckling med Prototyping av mindre system i småföretag.

Vår frågeställning är därmed: *Hur fungerar systemutvecklingsmetoden Prototyping vid systemutveckling av mindre system för småföretag?*

## 2 Teoretisk referensram

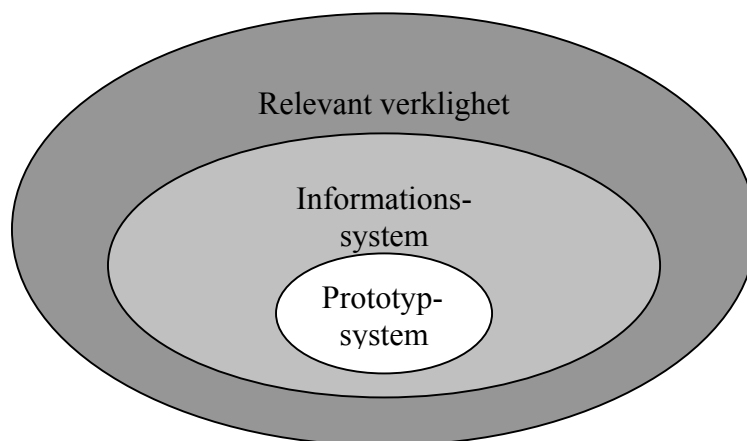
Enligt Ahlandsberg och Sandström (1988) finns ofta ett antal företeelser som kan orsaka problem vid andra systemutvecklingsmetoder. Det första problemet är att ett utvecklingsprojekt ofta pågår under en längre tid, vilket gör att resultatet visas så sent att användaren mister intresset för systemet som utvecklas. Det andra problemet är att det är svårt att ta reda på vad användaren egentligen vill ha och sen efter systemet tagits i bruk märker användaren att det ändå inte tillgodoser alla önskemål och krav. Användaren har då två alternativ, antingen be systemutvecklarna om vidareutveckling av systemet eller helt enkelt använda det utan att alla önskemål är uppfyllda. Ett tredje problem är att under den långa utvecklingsprocessen kan användarnas krav och önskemål förändras och när systemet väl tas i bruk tillgodoser systemet inte användarnas önskemål. Ett fjärde problem är kommunikationen som ofta kan vara dålig mellan användare och systemutvecklare under utvecklingsfasen.

Dessa problem slipper systemutvecklaren vid användning av metoden Prototyping. Användaren är under hela utvecklingsprojektet med i arbetet och kan påverka arbetet så användarens krav och önskemål uppfylls. Ett nära samarbete med användaren är ett krav för metoden, då det är användaren som ska bestämma när systemet är färdigutvecklat. På detta sätt mister inte användaren sitt intresse lika snabbt samt att de direkt kan påverka projektet om det är något de är missnöjda med. (Ahlandsberg och Sandström 1988)

För att förklara metoden Prototyping kommer först ett avsnitt med definition på systemutvecklingsmetoden. Härfter redogörs för huvudtyperna av Prototyping för att sedan komma in på stegen som ingår i dess process. Dessa olika steg ser vi som en förklaring på arbetsgången i metoden. Härfter kommer ett avsnitt som representerar mer ingående hur processen går till och vad som faktiskt ingår i de olika faserna i processen.

### 2.1 Definition

För att utveckla informationssystem av olika slag måste man utgå från verkligheten på något sätt. Verkligheten innehåller objekt i form av företeelser och ting som systemet senare ska kunna tillhandahålla information om. Verkligheten är således en mycket viktig del i systemutvecklandet. Mattsson et al (1988) skriver att ”Informationssystemet, med vilket vi avser ett datoriserat sådant, är en modell av verkligheten. Prototypsystemet är i sin tur en modell över det framtida informationssystemet.” Detta åskådliggörs i figur 1.



Figur 1 Prototypsystem (Källa: Mattsson et al 1988)

Med detta som underlag ges en definition av systemutvecklingsmetoden Prototyping: "En form av systemutveckling där en prototyp, dvs. en modell av det blivande systemet, programmeras och ligger till grund för fortsatt arbete". Ahlandsberg och Sandström (1988) kommenterar också prototypbegreppet som "Ett system som på ytan liknar ett tänkt framtida system".

Mattsson et al (1988) skriver angående traditionell systemutveckling som när "...de blivande användarna av systemet inte medverkar på ett aktivt sätt utan istället blir intervjuade om hur de vill ha systemet innan utvecklingen börjar". Med Prototypingmetoden får användaren alltså inom kortare tid än många andra utvecklingsmetoder se resultat i form av en prototyp av systemet. Systemutvecklarna visar användarna nya versioner av prototypen under hela utvecklingsprojektet och användaren har här tillfälle att komma med nya krav och önskemål eller bara förklara de krav och önskemål de först angivit. Detta är en stor fördel för utvecklaren då användaren under projektet aktivt kan följa att utvecklaren är på rätt spår i utvecklandet. Prototypen som användaren kontinuerligt ger respons på kan snabbt ändras jämfört med att ändra ett helt färdigt system. Kommunikationen är härmed bättre än vid andra utvecklingsmetoder eftersom systemutvecklarna har en kontinuerlig dialog med användaren genom att visa upp prototyper av systemet. Att visa prototyper istället för dokumentation av projektet ger en större förståelse för användaren. Ofta kan inte användaren tolka de olika dokumentationsdelar med olika komplicerade analys och designmodeller som projektdokumentationen innehåller i traditionell systemutveckling. (Ahlandsberg och Sandström 1988)

## 2.2 Olika typer av Prototyping

Enligt Mattsson et al (1988) finns två huvudtyper av Prototyping, vilka är Användarprototyping och Rapid Prototyping. Användarprototyping innebär att användarna av det framtida informationssystemet själva kommer att utveckla en prototyp som kommer att verka som kravspecifikation. Utifrån denna prototyp kommer sedan den anställda systemutvecklaren att utveckla ett färdigt system. Rapid Prototyping innebär istället att utvecklaren utvecklar en prototyp åt användarna som sedan de får bedöma. Användaren i det senaste alternativet har ofta ingen större systemutvecklingskunskap och har därmed ingen möjlighet att utveckla ett system själva.

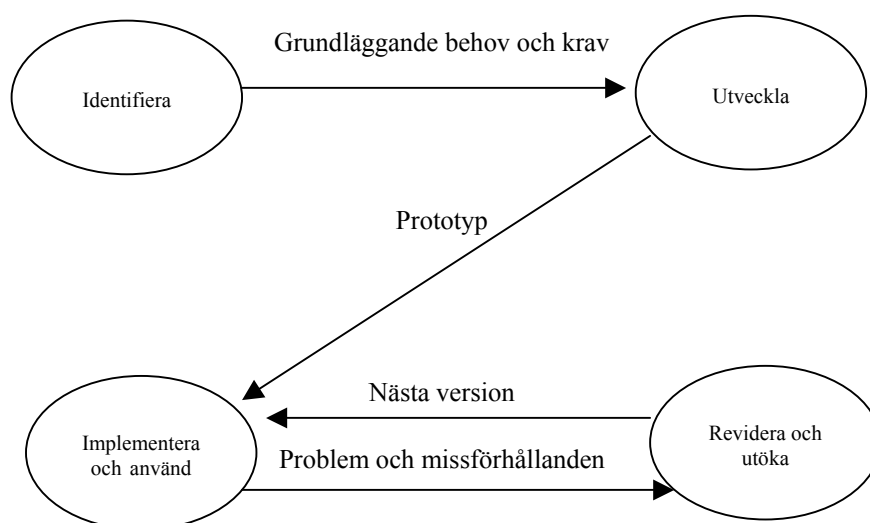
Rapid Prototyping är arbetsredskapet i denna uppsats. Arbetsgången är att utveckla en prototyp som frisersalongens anställda och kunder får bedöma.

## 2.3 Fyra steg till ett färdigt system

Prototypingmetoden bygger på en tät kommunikation mellan användaren och utvecklaren. Metoden innehåller enligt Ahlandsberg och Sandström (1988) fyra steg som visas i figur 2.

1. Steg ett innebär att utvecklaren och användaren identifierar vad systemet ska utföra. Krav och önskemål som användaren har bestämt ligger till grund för systemet. Det sker också en uppskattning av kostnaden för utvecklandet av systemet. I Prototypingmetoden förutsätts inte att alla krav och önskemål kommer fram i detta steg utan användaren kan komma senare i utvecklandet av systemet med andra krav. Det är en del i metoden att kontinuerligt ändra och förbättra systemet genom att bygga på med ytterligare funktioner.
2. Efter kommit fram till de grundläggande kraven som användaren vill ha utvecklas en första prototyp.
3. Härfter implementeras systemet för att användaren ska börja använda systemet och därmed utvärdera det. Vad användaren vill ändra på kommer fram och detta blir underlaget till nästa steg.
4. Prototypen ändras för att tillgodose användarens krav och önskemål. Efter ändringen och utökandet av systemet implementeras det på nytt för att användaren ska kunna utvärdera det igen.

Steg tre och fyra är en del av en iterativ process där användaren tycker till om hur prototypen tillgodoser användarens krav och gör den inte det utökas och förändras prototypen. Denna process sker tills användaren är nöjd.



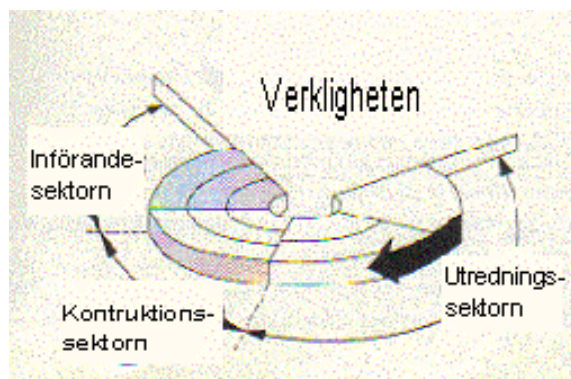
Figur 2 Prototypingmetoden (Källa: Ahlandsberg och Sandström 1988)



Det är användaren som bestämmer när utvecklingsarbetet är klart och har därmed en stor roll i processen. Detta kräver dock användare som är villiga att medverka aktivt i utvecklingsarbetet. Denna modell är passande när det krävs att snabbt få fram ett någorlunda fungerande system för att visa användarna. Här får utvecklaren snabbt reda på om han är på rätt väg och en utvärdering av systemet sker snabbare än vid många andra systemutvecklingsmetoder. Utvecklaren har också råd, både tidsmässigt och kostnadmässigt, att testa olika lösningar då det går relativt snabbt att skapa en prototyp. (Ahlandsberg och Sandström 1988)

## 2.4 Faser i prototyparbetet

För att tydligare klargöra vad som ska göras i Prototypingprocessen förklaras nedan ytterligare en modell. Denna modell kan ses som en ingående förklaring av de olika stegen, där vi beskriver hur utvecklingsarbetet går till snarare än att det görs. Modellen ovan av Ahlandsberg och Sandström (1988) kan istället ses som en modell av den faktiska arbetsgången vid utvecklingsarbetet, då den inte är lika detaljerad om vad de olika stegen innebär.



Figur 3 Cirkelmodell (Källa: Burman och Bäckman, 1992)

Figur 3 visar en cirkelmodell av Prototypingprocessen vilken innehåller olika sektorer. Modellen startar i verkligheten för att sedan inkorporera i de olika sektorerna, utredningssektorn, konstruktionssektorn och införandesektorn. Efter införandesektorn kommer vi in i verkligheten igen. Precis som Ahlandsberg och Sandströms (1988) modell av Prototyping är utvecklingsarbetet tänkt som en iterativ process. Burman och Bäckman (1992) skriver "Enligt många gamla systemeringsmetoder skall detta varv endast genomlöpas **en gång**, men idag kan man välja att gå igenom cirkelmodellen flera varv innan man är nöjd med resultatet". De skriver också att efter varje varv som utvecklingsarbetet genomgår fångas en allt större del av verkligheten upp och prototypen utökas.

Resultaten från en sektor ska användas som en bro över till nästa sektor. Dokumentationen som görs i utredningssektorn är en viktig del i denna metod som ska fungera som en övergång och ge underlag till nästa sektor. Dokumentationsdelen tas inte upp i Ahlandsberg och Sandströms (1988) version av Prototyping, varför vi istället betraktar den som en övergripande arbetsgång vid utvecklingsarbetet som görs i utredningssektorn.

## 2.4.1 Utredningssektor

I utredningssektorn utreds verkligheten och verksamheten för att skapa en grund inför konstruktionen av systemet. Enligt Burman och Bäckman (1992) görs detta genom att leta efter så kallade verksamhetsaktiviteter och analysera vilka mål och problem verksamheten har. Detta för att utreda och analysera vad systemet ska hantera. För att modellera och utreda verksamheten krävs att utvecklarna undersöker verksamheten och verkligheten kring den genom exempelvis intervjuer och observationer. Vi kommer att utöka denna modell genom att använda utvalda delar av metoden Objektorienterad analys och design (OOAD). Detta kan i andra utvecklingsprojekt som bygger på andra metoder än Prototyping ta en mycket lång tid. Eftersom OOAD metoden är tämligen utförlig kommer vi att endast använda vissa delar, då vi hela tiden har en tät kontakt med användarna som leder oss i rätt riktning i utvecklingsarbetet. Resultatet av denna sektor är alltså analys och design dokumentation som ligger till grund för konstruktionen av prototypen.

Det finns många olika sätt att modellera verkligheten och verksamheten och ett sätt är enligt Mathiassen et al (2001) att följa metoden OOAD. Metoden använder sig av objekt och klasser. Under analysen gäller det att försöka förstå systemets omgivning och då används objekt för att organisera omgivningen. Dessa objekt beskriver vanligen människor och ting i omgivningen. Under designarbetet används sen objekt för att beskriva och förstå systemet. Objekten beskriver då vanligen fenomen i systemet som vi kan styra. Klasser används i sin tur till att beskriva objekten. Många objekt kan likna varandra och då används klasser för att beskriva ett antal objekt på samma gång. (Mathiassen et al 2001)

Metodens inriktning på objekt anges ge många fördelar gentemot andra metoder. Att använda objekt för att beskriva fenomen ger tankemässig struktur, men den ger också andra fördelar. Mathiassen et al (2001) skriver ”Objekt, tillstånd och beteende å andra sidan är mer generella begrepp och är lämpliga för att beskriva de flesta fenomen som kan uttryckas i ett naturligt språk”. Vidare skrivs att en av objektens främsta fördelar är att den ger tydlig information om systemets omgivning.

Syftet med metoden är att bestämma systemkraven, att förstå en systemdesign utan osäkerheter och att förstå ett system, dess omgivning och villkoren för dess implementering. (Mathiassen et al 2001)

Metoden består av fyra stora delar: analys av problemområdet (se 2.4.1.1), analys av användningsområdet (se 2.4.1.2), arkitekturdesign (se 2.4.1.3) och komponentdesign (se 2.4.1.4). Syftet med att beskriva metoden OOAD är att visa på vad metoden går ut på och *vad* som görs snarare än *hur* det görs. Att beskriva metoden känns viktigt eftersom vi utökar den ursprungliga dokumentationsdelen i Prototyping med denna objektorienterade metod. Vi vill genom denna beskrivning försöka ge en bild av vad OOAD går ut på.

### 2.4.1.1 Analys av problemområdet

Syftet med denna första del av metoden är att avgränsa och modellera ett problemområde. Problemområdet är enligt Mathiassen et al (2001) ”den del av omgivningen som administreras, övervakas eller styrs av ett system”. Vidare menas att omgivningen ska modelleras som användaren kommer att se den. Här ska fokus vara på omgivningen, inte på existerande eller framtida system. Systemet ska hanteras senare med modelleringen av

omgivningen som underlag. Mathiassen et al (2001) skriver också ”När man väl har en god modell kan man använda den för att designa och implementera ett system som kan behandla, överföra och presentera information om problemområdet på ett ändamålsenligt och användbart sätt”.

Aktiviteterna som ingår i denna fas är att man väljer de objekt, klasser och händelser som kommer att ingå i modellen av problemområdet (2.4.1.1.1). Därefter reds ut vilka klasser och objekt som är förknippade med varandra och därmed har en relation, en struktur (2.4.1.1.2). Som sista aktivitet i denna fas gäller det att reda ut vilka egenskaper objekten ska ha, alltså dess beteende (2.4.1.1.3). (Mathiassen et al 2001)

#### **2.4.1.1.1 Klasser**

I denna aktivitet ställer man sig grundfrågan: ”Vilka objekt, klasser och händelser bör vi inkludera i modellen och vilka bör vi utelämnas?” Fenomen ska genom att se dem som objekt och händelser abstraheras från problemområdet. Objekten ska sedan klassificeras och därefter sker en systematisk värdering och val av vilka som systemet ska ge information om. Det är endast de objekt, klasser och händelser som vi vill att systemet ska kunna ge information om som ska vara med. Resultatet blir en händelstabell med de klasser och händelser som ska vara med i analysen av problemområdet. (Mathiassen et al 2001)

#### **2.4.1.1.2 Struktur**

Förra aktiviteten gav oss objekt, klasser och händelser. Detta resultat utgår vi ifrån när vi i denna aktivitet beskriver de strukturella relationerna mellan klasser och objekt i ett problemområde. Endast de nödvändigaste relationerna ska finnas med. Resultatet av denna del kommer att vara ett klassdiagram med klasser och dess relationer.

I modelleringen förenklar vi objekten genom att enbart fånga de utvalda egenskaperna som är relaterade till problemområdet och därmed intressanta för oss i vårt byggande av systemet. När det gäller relationer mellan klasser är vi intresserade av relationen mellan två eller flera klasser. Detta görs enligt Mathiassen et al (2001) genom att använda två olika sorters klasstrukturer: ”generaliseringsstrukturer, som beskriver ett antal klasser som specialiseringar av en mer generell klass, och klusterstruktur, som grupperar samlingar av relaterade klasser”. Här finns också två typer av objektstrukturer, vilka är aggregat och association. Aggregatstruktur visar en relation mellan två eller flera objekt som ”uttrycker att ett objekt är en fundamental och definierande del av ett annat objekt”. Exempelvis så består en bil av olika delar som kaross, motor och hjul. En associationsstruktur beskriver en relation mellan två eller flera objekt där objekten helt enkelt har en meningsfull relation mellan varandra. Exempelvis så har en bil och en person en relation i och med att en bil ägs av en person. (Mathiassen et al 2001)

#### **2.4.1.1.3 Beteende**

I denna aktivitet utgår man ifrån den händelstabell och det klassdiagram som var resultatet av de två förra aktiviteterna. Det gäller här att beskriva beteendemönster och attribut för att som resultat få ett tillståndsdigram. Här ska redas ut vilka olika händelser ett objekt kan vara

inblandat i. Dessa händelser blir tillsammans ett visst beteendemönster för ett objekt vilket kartläggs. Beteendemönstret ska till slut beskriva ett beteende som är gemensamt för alla objekt i klassen i ett så kallat tillståndsdigram. (Mathiassen et al 2001)

Mängden av händelser som objekt är inblandade i utgås ifrån när beteendemönstret beskrivs. En händelsetabell har redan gjorts i tidigare aktivitet, men dessa händelser är oordnade. I denna aktivitet börjar man med att finna den första och sista händelsen. Detta görs enligt Mathiassen et al (2001) genom specifika frågor: "Vilka händelser leder till att det skapas ett objekt i problemområdet? Vilka händelser orsakar att ett objekt i problemområdet försvinner?" När detta är utrett tas resten av händelserna där emellan omhand och ordnas.

### **2.4.1.2 Analys av användningsområdet**

Mathiassen et al (2001) skriver "Om man börjar med att analysera problemområdet kommer man att fokusera på vad det hela egentligen handlar om, snarare än på gränssnitt och funktioner. Senare kan man ge sig in på användning av systemet och definiera användarkrav på grundval av en fundamental förståelse av begreppen i området".

Detta är den andra fasen av metoden OOAD och här ska systemets övergripande användningskrav bestämmas. Med ett användningsområde menas "en organisation som administrerar, övervakar eller styr ett problemområde". Användningsområdet ska i denna fas bestämmas och detta görs genom så kallade användarfall. Användarfall är i sin tur ett mönster för interaktion mellan systemet och aktörer i användningsområdet. Här är det alltså nödvändigt att samarbeta med användarna. Huvudfrågan för denna fas är "Hur kommer målsystemet att användas?". Utifrån frågan ska systemets funktioner och gränssnitt definieras. (Mathiassen et al 2001)

De aktiviteter som ingår i denna fas är att först kartlägga hur systemet samverkar med människor och andra system i omgivningen (se 2.4.1.2.1), sen måste kartläggas vad systemet ska kunna delge information (se 2.4.1.2.2) och slutligen vilka kraven på gränssnittet är (se 2.4.1.2.3). (Mathiassen et al 2001)

#### **2.4.1.2.1 Användning**

Här bestäms hur aktörer kan interagera med systemet. Detta görs genom att bestämma användningsområdet med hjälp av användarfall. Dessa användarfall ska utvärderas i samarbete med användarna och resultatet blir en beskrivning av alla användarfall och aktörer. Resultatets användarfall bestämmer sedan all användning av systemet. (Mathiassen et al 2001)

Först och främst gäller det att hitta aktörer och användarfall till systemet. Utvecklarna måste ta reda på *vilka* som ska använda systemet och *hur* de ska använda systemet. Detta görs genom att identifiera vilken arbetsfördelning som finns och försöka hitta roller som finns i systemets omgivning. Denna undersökning besvarar frågan om vem som använder systemet och på vilket sätt. För att bestämma användarfall fordras även ett bra samarbete mellan användare och utvecklare. Användarna formulerar vilka behov de har på systemet och utvecklaren formulerar därefter användarfall med användarnas behov som grund. (Mathiassen et al 2001)

När alla aktörer identifieras ska de beskrivas i en så kallad aktörsspecifikation, där bland annat mål, kännetecken finns för varje aktör. När alla användarfall är identifierade gäller samma sak. En användarfallsspecifikation upprättas för varje användarfall, där själva interaktionen mellan aktören och systemet beskrivs och vilka objekt samt funktioner som berörs. (Mathiassen et al 2001)

#### **2.4.1.2.2 Funktioner**

Här ska bestämmas vad systemet ska utföra åt aktörerna. Detta görs genom att först hitta alla de funktioner som systemet ska utföra. Därefter specificeras de komplicerade funktionerna. Som sista steg gäller det att utvärdera bland de funktioner man hittat innan resultatet fastslås. Resultatet blir en lista av funktioner för systemet. (Mathiassen et al 2001)

För att hitta funktioner finns det olika sätt. Det gäller att hitta de källor som ger funktioner. ”Varifrån kommer de systemkrav som ger funktioner?” I en tidigare fas gick vi igenom problemområdet och bland de klasser och händelser som fastställdes kan funktioner hittas. Sen kan utvecklaren också gå igenom beskrivningen av användningsområdet för att hitta funktioner. Detaljnivån vid beskrivningen av funktionerna beror på vilka som är med i projektet. Funktionerna måste vara så pass detaljerat beskrivna så alla förstår exakt vad funktionen innebär. Mer komplicerade funktioner kräver en mer detaljerad specifikation än enklare funktioner. (Mathiassen et al 2001)

Slutligen ska användaren kontrollera funktionslistan för att se att de funktioner är med som önskas. Denna lista ska också jämföras med systemdefinitionen och modeller som tidigare skapats så allt stämmer överens. (Mathiassen et al 2001)

#### **2.4.1.2.3 Gränssnitt**

Syftet med denna aktivitet är att bestämma gränssnittet för systemet, både användargränssnitt och systemgränssnitt. Resultatet är för användargränssnittet dialogstilar<sup>1</sup>, presentationsformer, en fullständig lista över komponenter i användargränssnittet, utvalda fönsterdiagram<sup>2</sup> och ett navigeringsdiagram<sup>3</sup>. Med en komponent innebär ”en samling programdelar som utgör en helhet och har ett väldefinierat ansvarsområde”. Resultatet för systemgränssnittet är klassdiagram för externa apparater och protokoll för interaktion med andra system. Detta görs genom att först bestämma, beskriva och sedan utvärdera komponenterna. Det är lättare att nu fysiskt se problemen som eventuellt återstår att ta omhand. (Mathiassen et al 2001)

För att bestämma användargränssnittets komponenter gäller det att ta hänsyn till en del saker. Bland annat så skriver Mathiassen et al (2001) ”Som helhet måste dialogen vara enkel, naturlig och konsekvent, kraven på användarens minne måste vara minimala, återkopplingen måste vara informativ och konstruktiv och fel måste förebyggas”. Nielsen (2001) skriver också om vikten om ett enkelt gränssnitt: ”Enkelheten vinner alltid över komplexiteten, särskilt på webben där var femte byte du sparar in ger en förkortning av hämtningstiden på en

---

<sup>1</sup> Hur kommunikationen sker mellan användare och system.

<sup>2</sup> Beskriver hur ett fönster är uppbyggt och vilka element som ingår.

<sup>3</sup> Förminskad bild av varje fönster med pilar emellan som anger funktionalitet och vad som kan utföras i systemet.

millisekund”. När man väl bestämt sig för hur gränssnittet ska se ut och vad man ska använda är det dags för att beskriva de olika delarna. Liksom vid funktioner ska gränssnittet beskrivas och specificeras, men onödiga detaljer ska inte vara med och endast de mest komplicerade komponenterna specificeras detaljerat. Vid utvärderingen kontrolleras uppdelning av gränssnittet i de komponenter som skapats och utformning av de enskilda komponenterna i gränssnittet. Det är endast de väsentliga komponenter som ska vara med som verkligen används och komponenterna ska interagera med varandra. (Mathiassen et al 2001)

### **2.4.1.3 Arkitekturdesign**

Syftet med den tredje fasen är att strukturera systemet. Resultatet blir relationer för komponenter och processer i ett system.

Arkitekturen baseras på tre grundläggande principer. Den första principen är ”definiera och prioritera kriterier”. Man kan inte få alla kriterier uppfylla. I många fall utgör de olika kraven och kriterierna motsatsförhållanden vilket kräver prioritering. Exempelvis kan ibland kriteriet att systemet ska vara mycket säkert göra att det kan vara svårt att uppfylla kriteriet att systemet ska vara lätt att underhålla. Den andra principen är ”bygg en bro mellan kriterier och den tekniska plattformen”. Enligt Mathiassen et al (2001) kan arkitekturdesignen innebära komplicerade beslut som har oförutsägbara konsekvenser. Med detta som underlag är det bra att inte basera designen på endast spekulationer. Härav kommer den tredje principen, ”utvärdera designen tidigt”.

De tre aktiviteterna i denna fas är att ta reda på villkoren och kriterierna för designen (se 2.4.1.3.1), systemets struktur i komponenter (se 2.4.1.3.2) och hur distribuering och samordning av systemets processer ska ske (se 2.4.1.3.3). (Mathiassen et al 2001)

#### **2.4.1.3.1 Kriterier**

Syftet är att bestämma designprioriteringar och resultatet innebär en samling prioriterade kriterier i ett skriftligt dokument.

Ett system kan aldrig bli helt perfekt. Som tidigare nämnts så gäller det att prioritera bland de kriterier som ställs på systemet. Mathiassen et al (2001) skriver ”Kvalitet är väsentligen frånvaro av brister”. Systemet behöver alltså inte vara helt perfekt och fri från brister. Är den perfekt är det snarare så att komplexiteten av systemet kan ha gått förbi och utvecklarna missat något väsentligt. Att ta med brister i designen visar att man upptäckt dem och menar att behandla dem. (Mathiassen et al 2001)

Det är alltså viktigt att gå igenom vilka kriterier som ska gälla och i vilken prioritetsordning. Mathiassen et al (2001) nämner en princip som lyder ”En god design gör en avvägning mellan flera kriterier”. Det gäller först att analysera vilka villkor som gäller, därefter ska kriterierna övervägas och till sist gäller det att prioritera. När kriterierna övervägs betonar metoden tre generella designkriterier. ”En god design är användbar, flexibel och begriplig” och behöver inte vara perfekt och fri från brister. Med användbarhet innebär att designen ska tillgodose användarnas behov och designen ska passa den tekniska plattformen. Flexibilitet syftar till att systemet ska kunna fungera även om förutsättningar ändras. Begriplighet specificerar att bland annat dokumentationen ska vara lätta att förstå för användaren. (Mathiassen et al 2001)

#### **2.4.1.3.2 Komponentarkitektur**

Syftet med att skapa komponenter är att skapa en bra arkitektur för systemet. Med en komponent innebär alltså ”en samling programdelar som utgör en helhet och har ett väldefinierat ansvarsområde”. En komponent kan vara alltifrån en klass till hela systemet. Komponentarkitektur är ”en systemstruktur med inbördes relaterade komponenter”. Resultatet av denna aktivitet är ett klassdiagram med komponenter och en komponentspecifikation. (Mathiassen et al 2001)

Utgångspunkten är resultatet från förra fasen, de fastslagna kriterierna. Utifrån dessa ska delsystem, komponenter, definieras i det system som byggs. Därefter ska komponenter identifieras. Dessa komponenter och delsystemen ska sedan bilda ett klassdiagram. Utifrån klassdiagrammet specificeras sedan komplicerade komponenter och tillsammans bildar de en komponentspecifikation. Systemet ska delas in i delar för att skapa ansvarsområden och få en bättre överblick. (Mathiassen et al 2001)

#### **2.4.1.3.3 Processarkitektur**

Syftet med denna aktivitet är att definiera struktureringen av ett system. Här handlar det om processer och det innebär ”en stycke utrustning som kan exekvera ett program”. En processarkitektur är i sin tur ”en systemexekveringsstruktur bestående av ömsesidigt beroende processer”. Resultatet av aktiviteten är ett fördelningsdiagram. Detta diagram visar de processer systemet har och de programkomponenter och aktiva objekt som processen har blivit tilldelad. En programkomponent är ”en fysisk programmodul”. Ett aktivt objekt är ”ett objekt som har tilldelats en process”. (Mathiassen et al 2001)

Vid processarkitekturen fokuseras på exekvering där processer och objekt är aktuellt istället för komponenter och klasser. Det blir ett steg mot den fysiska implementeringen av systemet och syftet är att strukturera själva exekveringen av programmet. De komponenter som beskrivs i förra aktiviteten exekveras av denna aktivitets processer. Det gäller alltså att fördela komponenterna man definierat på processer. (Mathiassen et al 2001)

Det första som ska göras i denna aktivitet är utifrån klassdiagrammet och komponentspecifikationen som gjordes i förra aktiviteten fördela programkomponenterna på de processer som finns i systemet. Därefter gäller det att hitta de delade resurserna i systemet för att hitta möjliga flaskhalsar i systemet. Delade resurser kan vara processer, programkomponenter eller extern apparat av något slag. Sist ska samordningsmekanismer väljas. Dessa mekanismer kan vara synkronisering av data eller utbyte av data. (Mathiassen et al 2001)

#### **2.4.1.4 Komponentdesign**

Den fjärde och sista fasen i metoden OOAD är komponentdesign. Syftet är ”att bestämma en implementering av krav inom ramen för en arkitektur”. Resultatet av fasen är en beskrivning av de komponenter som ska finnas i systemet. (Mathiassen et al 2001)

De tre aktiviteterna är att ta reda på hur modellen representeras som klasser i systemet (se 2.4.1.4.1), hur funktionerna implementeras (se 2.4.1.4.2) och hur komponenterna är förbundna med varandra (se 2.4.1.4.3).

#### **2.4.1.4.1 Modellkomponent**

Syftet med modellkomponentaktiviteten är att designa en representation i form av en modell av problemområdet. Modellkomponenten ska lämna data till funktioner, gränssnitt och användaren och denna lagrade informationen hämtas från problemområdet.

Modellkomponenten är alltså den del av systemet som implementerar modellen av problemområdet. Resultatet är ett reviderat klassdiagram från analysen där nu även modellkomponenten är med. (Mathiassen et al 2001)

För att skapa en modellkomponent granskas först klassdiagrammet, beteendemönster och komponentspecifikation. Därefter ska gemensamma och privata händelser representeras. Sist ska klasserna omstruktureras. Detta innebär att det gamla klassdiagrammet revideras med nya klasser, attribut och relationer. (Mathiassen et al 2001)

Mathiassen et al (2001) skriver att privata händelser är händelser där endast ett objekt från problemområdet är inblandat. Dessa händelser ska representeras beroende på hur många gånger händelsen inträffar. ”En händelse som inträffar högst en gång kan ihågkommas som klassattribut. En händelse som inträffar godtyckligt många gånger fordrar en ny klass.”

Om en händelse är gemensam påverkar händelsen flera objekt och då bör händelsen representeras i förhållande till ett av objekten. Händelsen representeras beroende på hur händelsen är involverad i tillstånddiagrammet. Mathiassen et al (2001) skriver att ”Om händelsen är involverad i tillstånddiagrammen på olika sätt, representera den i förhållande till den klass som ger den enklaste representationen. Om händelsen är involverad i tillstånddiagrammen på samma sätt, måste man väga olika möjliga representationer mot varandra.”

Till sist ska klasserna omstruktureras och förenklas och då används bland annat generalisering, association eller iterationer.

#### **2.4.1.4.2 Funktionskomponent**

Syftet med denna aktivitet är att bestämma implementeringen av funktioner. Funktioner används till att ge användargränssnittet och andra systemkomponenter tillgång till modellen där all data finns som efterfrågas. Resultatet är enligt Mathiassen et al (2001) ”ett klassdiagram med operationer och specifikationer av komplicerade funktioner”.

Funktioner ska designas som operationer. Detta innebär bland annat att funktionerna delas in efter typ. Det finns bland annat uppdateringsfunktion, avläsningsfunktion, beräkningsfunktion och signaleringsfunktion. Efter denna del ska de komplicerade funktionerna specificeras och förenklas. Enkla funktioner behöver endast tillsättas namn, men för de mer komplicerade funktionerna krävs mer detaljerade beskrivningar. Inga osäkerheter får finnas i designen och funktionerna kan bland annat förklaras genom operationsspecifikation, sekvensdiagram och tillstånddiagram. Vid operationsspecifikationen förklaras exempelvis funktionens syfte,



kategori, villkor, indata, effekt, algoritm, placering, inblandade objekt och utlösande händelser. Sekvensdiagrammet visar interaktionen mellan objekt för just den funktionen. Tillståndsdigrammet visar relationen mellan tillstånd och tillståndsförändringar hos ett objekt. All denna dokumentation blir sen resultatet i form av funktionsspecifikation. (Mathiassen et al 2001)

#### **2.4.1.4.3 Förbindelser mellan komponenter**

Syftet med aktiviteten är enligt Mathiassen et al (2001) ”att förbinda systemkomponenter med varandra”. Resultatet är vidare ”ett klassdiagram över de inblandade komponenterna”.

I tidigare aktivitet kom vi fram till att flexibilitet och begriplighet är viktiga kriterier för design. Detta kan mätas genom koppling och kohesion. Klassdiagrammet som vi har för de inblandade komponenterna utvärderas. Hög kohesion inom klasser och löst kopplade komponenter är det mått som strävas efter. (Mathiassen et al 2001)

Mathiassen et al (2001) skriver att koppling är ”ett mått på hur nära två klasser eller komponenter är förbundna med varandra”. Klasser eller komponenter ska inte vara hårt förbundna och detta ska strävas efter att minimeras.

Kohesion är däremot ”ett mått på hur väl en klass eller komponent hänger samman”. Detta är bra för systemet och ska eftersträvas.

Aktiviteten innebär att förbinda klasser och sedan utvärdera dessa förbindelser. För att få klasser att hänga samman finns olika former. Det kan ske genom aggregering av klasser mellan lika komponenter. Det kan även ske genom specialisering av klasser. En publik klass tas då från en annan komponent. Det går också att göra operationsanrop till operationer i en annan komponent. Dessa olika sätt gör att kohesionen ökar mellan olika komponenter. Efter denna del ska förbindelserna utvärderas. (Mathiassen et al 2001)

### **2.4.2 Konstruktionssektor**

Burman och Bäckman (1992) skriver att ”Ut ur konstruktionssektorn får vi systembeskrivning, programkod, strukturer och databasbeskrivningar.” Vid konstruktionssektorn programmerar vi bokningssystemet och e-handelssystemet för Klippotek Passagen. De databaser som används av de båda systemen skapas också här.

### **2.4.3 Införandesektor**

Införandesektorn är sista delen i cirkelmodellen. Burman och Bäckman (1992) skriver att i denna sektor ska datorer installeras, utbildning ska ske av användarna och grunddata i systemet ska matas in. ”Vad vi nu tillfört verkligheten är en databas, en datorutrustning samt den programvaran som behövs för vårt system i det skick den befinner sig efter ett varv.” Dessa tekniska delar att installera datorerna sker oftast endast på första varvet. Samma dator kan självklart användas även om programvaran har ändrats efter andra varvet. Resultatet av denna sektor är installationen av prototypen hos användarna och de är nu färdiga att utvärdera och testa prototypen.

Efter denna sektor kommer vi till utredningssektorn igen för att utöka prototypen om användarna inte är nöjd med implementerad prototyp. Då kommer ändring av dokumentationen och modelleringen att ske om användarna ville ha ändringar i prototypen eller utökad dokumentation och modellering om användaren ville ha ytterligare funktioner. (Burman och Bäckman 1992)

### 3 Metod

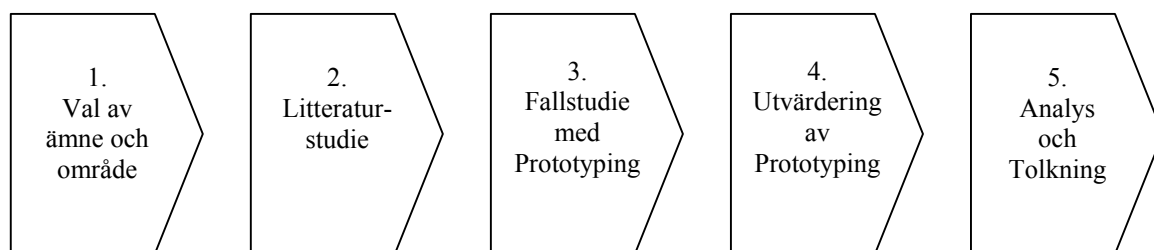
Systemvetarprogrammet har gett oss möjlighet att prova på en mängd olika metoder och teorier. Magisteruppsatsen gav oss möjlighet att sammanfoga många av våra erhållna kunskaper i ett och samma arbete. Efter en lång tids funderande kom vi på att vi ville utveckla ett eget system för att visa våra kunskaper inom ämnet systemutveckling. Vi hade turen att bli kontaktade av ett företag som ville etablera sig på Internet och vi fick möjlighet att styra projektet som vi ville. Efter en tids studerande efter intressant systemutvecklingsmetod hittade vi Prototypingmetoden som verkade kunna fungera bra för systemutveckling av mindre system i småföretag.

Vår metod består i huvudsak av två delar. Den ena delen är en teoretisk studie och den andra delen är en fallstudie. Den teoretiska studien består av en litteraturstudie där vi bland annat studerade processen vid Prototyping och tog del av vad som var känt om metoden. Fallstudien har genomförts vid frisersalongen Klippotek Passagen på Drottninggatan i Göteborg och vi har där själva utvecklat två system med metoden Prototyping för att själva se hur den fungerar.

Backman (1998) skriver om den traditionella forskningsprocessen som innehåller de delar i ett uppsatsarbete som bör på något sätt vara med. Vi utgår från denna process för att beskriva vår metod.

#### 3.1 Vårt val av metod

För att på ett överskådligt sätt förklara hur vi genomfört vårt magisteruppsatsarbete har vi delat upp arbetet i olika faser.



För att beskriva vårt val av metod följer nedan en förklaring av vårt ställningstagande till olika synsätt och forskningsmetoder. I forskningssammanhang talas om bland annat två traditioner; positivistisk och fenomenologisk tradition. Easterby-Smith et al (1991) redogör för dessa olika traditioner och menar att de skiljer sig markant åt genom att se världen och forskningen på olika sätt. Efter detta avsnitt förklarar vi vår egen metod. Vi använder oss av intervjuer och observationer vilket har gett kvalitativ data. Vi anser att intervjuer och observationer har varit det bästa alternativet för att undersöka hur användarna upplevt sin roll i utvecklingsarbetet. Observationer har varit nödvändiga för att ta reda på omständigheter vid Klippotek-Passagen.

### 3.1.1 Positivistiskt synsätt

Det positivistiska synsättet utgår från att världen kan studeras objektivt utan att åsikter och värderingar påverkar resultatet. Objektiv forskning menar att oavsett vem som utför forskningen så blir resultatet likadant om bara forskningen utförts på objektivt och därmed rätt sätt. Världen som studeras är oberoende av forskarens värderingar. (Easterby-Smith 1991)

Positivistisk tradition har ofta ett deduktivt angreppssätt av forskningen. Vid ett sådant angreppssätt utgår forskaren från en teori. Forskaren skapar hypoteser om orsakssamband som det insamlade materialet i studien analyseras med. Dessa hypoteser bekräftas eller falsifieras. Materialet samlas ofta in med kvantitativa metoder. (Easterby-Smith 1991)

Detta fungerar inte i vårt fall eftersom vi anser oss behöva använda intervjuer och verkligen komma i kontakt med frisören och kunderna för att få deras åsikter om Prototypingprocessen. Vidare behöver vi observationer för att klargöra rådande omständigheter vid klippotek Passagen.

### 3.1.2 Fenomenologiskt synsätt

Fenomenologin utgår däremot från att forskningen inte är objektiv och att forskaren alltid har värderingar som påverkar studien. Här beror det alltså på vem som gör forskningen eftersom man har olika bakgrund och erfarenheter och kommer härigenom att genomföra forskningen med olika "glasögon". Vi ser världen på olika sätt. Fenomenologin menar att omvärlden är socialt konstruerad och relationen mellan forskaren och de personer som intervjuas påverkar forskningens resultat. (Easterby-Smith 1991)

Fenomenologisk tradition jobbar enligt induktivt angreppssätt. Här utgås inte från någon hypotes eller teori. Endast det insamlade materialet ligger till grund för resultatet. Materialet samlas ofta in med kvalitativa metoder. (Easterby-Smith 1991)

Utifrån våra behov använder vi metoder som bygger på fenomenologiskt synsätt och ger kvalitativ data. Metoden ger oss en rik datamängd genom en konstant interaktion med användarna.

### 3.1.3 Kvantitativa metoder

I kvantitativ studie har man redan från början bestämt sig för vilka slutsatser studien kan leda till. Metoden studerar det som kan iakttas objektivt och detta innebär att forskaren under materialinsamlandet håller sig objektiv. I kvantitativa metoder sker ett förutbestämt urval och av detta dras resultatet. Kvantitativa studier använder sig ofta av mätningar av olika slag. Det kan vara mätningar av vissa värden eller åsikter och uppfattningar. Intervjuer kan förekomma vid kvantitativ metod, men enkäter är vanligast förekommande. Intervjuer och enkäter är vid kvantitativa metoder strukturerade och svaren begärs ofta i fasta svarsalternativ. Observationer förekommer också, men inriktar sig då på ett specifikt skeende. (Easterby-Smith 1991)

Easterby-Smith et al (1991) skriver att kvantitativa metoder och det positivistiska synsättet har styrkan att den är snabb och ekonomisk. Vidare skriver han "On the debit side, these methods tend to be rather inflexible and artificial; they are not very effective in understanding

processes or the significance that people attach to actions; they are not very helpful in generating theories; and because they focus on what is, or what has been recently, they make it hard for the policy-maker to infer what changes and actions should take place in the future.”

### **3.1.4 Kvalitativa metoder**

Det som kännetecknas som kvalitativa metoder är att forskningens resultat inte från början är förutbestämt vilket det kan bli. Här utgår man ifrån det insamlade materialet, bildar sig en uppfattning och drar sedan slutsatser. Denna metod kräver att materialet verkligen analyseras och bearbetas. Kvalitativa metoders resultat baseras på upplevda erfarenheter som gjorts under studien och resultatet dras när tillräckligt mycket information samlats ihop. De tekniker som vanligtvis används vid kvalitativa metoder är intervjuer och observationer. Intervjuerna kan antingen vara muntliga och skriftliga. Frågorna kan vara halvstrukturerade eller helt öppna. Halvstrukturerade frågor är frågor som är något begränsade och kan rikta sig mot ett visst område. Under intervjuerna skapas en förtroendefull relation mellan intervjuaren och informanten. (Easterby-Smith 1991)

Vad gäller det fenomenologiska synsättet och kvalitativa metoder så är deras styrka att de kan titta över förändringsprocesser över tiden och som Easterby-Smith et al (1991) skriver ”...to understand peoples meanings, to adjust to new issues and ideas as they emerge, and to contribute to the evolution of new theories”. Svagheten för kvalitativa metoder är att det kan ta mycket tid och resurser att analysera materialet samt analysen av data kan ibland vara svår.

## **3.2 Val av ämne och område**

Vi fick kontakt med frisersalongen Klippotek Passagen som ville etablera sig på Internet. De förklarade vid ett möte tidigt i januari vad de kräver av vår insats som systemutvecklare och därefter kontaktades kursansvarige för magisteruppsatsen för diskussion av ämnet. En avgränsning gjordes och ämnet blev inringat. Formuleringen av problemet blev: *Hur fungerar systemutvecklingsmetoden Prototyping vid systemutveckling av mindre system för småföretag?*

## **3.3 Litteraturstudie**

Vi började omgående med en massiv litteraturstudie för att till en början få en så bred kunskap inom ämnet systemutveckling som möjligt och för att därefter specialisera oss på vald systemutvecklingsmetod. Vi lånade en mängd böcker om systemutveckling för att läsa in oss på ämnet. Denna litteraturstudie gav oss underlag att tidigt börja programmeringsfasen. Backman (1998) skriver om vikten av att ha läst litteratur inom forskningsområdet och att denna del måste genomgås ”innan det egentliga forskningsarbetet inleds”. Litteraturgranskning ger oss information om tidigare brister i forskning inom ämnet och ”hjälp oss att formulera en meningsfull ’forskningsbar’ vetenskaplig problemställning”. Samtidigt som litteraturstudien påbörjades den teoretiska referensramen som ligger till grund för studien om Prototyping och som är resultatet av litteraturstudien. Litteraturstudien gav upphov till att vi omformulerade vår frågeställning som härefter var mer precis och konkret. I Backmans (1998) forskningsprocess kommer problemformuleringsfasen efter en

litteraturgranskning och han menar att efter litteraturfasen måste problemformuleringen ”hyfsas för att det ska vara möjligt att ge den ett empiriskt svar”.

### **3.4 Fallstudie med Prototyping**

Figur 2 representerar som sagt arbetsgången för fasen av fallstudien. Först gjordes i samarbete med frisersalongen en liten kravlista som identifierade grundläggande behov och krav för bokningssystemet och e-handelssystemet genom en intervju med frisören och därefter observationer. Här efter användes Objektorienterad analys och design för att dokumentera förarbetet vid systemutvecklingen. Programmeringen av en första prototyp kom snabbt igång och bara några veckor efter kravlistan skapades var prototypen klar för både bokningssystemet och e-handelssystemet. Systemen sattes i bruk och enkäter skickades ut till kunder. Enkäterna innehöll frågor om hur kunderna upplevde prototypen samt frågor om vilka förändringar som krävdes för att tillgodose kundernas behov och krav. Med dessa svar från Klippotek Passagens stamkunder ändrades prototypen och implementerades efter någon vecka igen för att genom fler enkäter få ytterligare respons från kunderna.

Denna fas kan också benämnas som observationsfasen. Det är enligt Backman (1998) ”den fas under vilken forskaren skaffar sig ’belägg’ för sina hypoteser”. I denna fas undersöks verkligheten och detta kan ske med bland annat experiment, enkätintervjuer och observationer. Det är i denna fas som data samlas in som ligger till grund för resultatet. I vårt fall är det i denna fas vi utför vårt experiment med att använda Prototyping för att med egna ögon se hur den fungerar.

#### **3.4.1 Intervju med frisören**

Easterby-Smith et al (1991) skriver att en intervju är lämpligt när ”one aim of the interview is to develop an understanding of the respondent’s ’world’ so that the researcher might influence it, either independently or collaboratively as might be the case with action research”.

En intervju gjordes tidigt vid kursens början för att ta reda på vad frisörerna förväntade sig för system och vilka omständigheter som råder vid frisersalongen. Intervjufrågorna finns som bilaga 1 till uppsatsen. Genom denna intervju fick vi underlag till en första prototyp genom de krav som frisörerna ställer. En av frisörerna blev vår kontaktperson vid Klippotek Passagen och det var med honom vi utförde vår intervju. Vi frågade frisören vad för system de ville ha i öppna frågor för att han skulle kunna svara helt fritt och så alla krav de kunde komma på kom fram. Sedan följde frågor som var relativt styrda för att ta reda på speciella omständigheter kring verksamheten som kan ha betydelse för oss.

#### **3.4.2 Observation**

Vi har använt Easterby-Smith’s (1991) roll ” Interrupted involvement” som beskrivs som ”..kind of role for the observer is for her to be present sporadically over a period of time, moving for example in and out of the organisation to deal with other work or to conduct interviews with, or observations of, different people across a number of different organisations”.

Tre observationer gjordes också för att ge oss systemutvecklare en bild av vad frisersalongen var ute efter för system. Detta för att bekräfta de omständigheter frisören uppgav samt för att ge oss själva en någorlunda objektiv bild av arbetet på frisersalongen. Vi beslutade att besöka salongen vid 3 tillfällen, vilka var 27/1, 14/2 och 24/2. Vi valde dessa olika tillfällen eftersom vi tror att det kan ha betydelse när i månaden man gör observationen, då löneutbetalningar i viss mån kan styra vad kunderna har råd med. Vid alla tillfällena var vi på salongen i fyra timmar. Under denna tid observerades hur mycket telefonen ringde, vilka produkter som såldes och även om det fanns produkter kunderna önskade men som inte fanns i lager. Efter varje observation gjordes en sammanställning av våra intryck.

### **3.4.3 Prototyp**

Efter en intervju med vår kontaktperson på Klippotek Passagen och en observation hade vi nu en någorlunda bild på vad det var frisörerna förväntade sig. Med detta som underlag började vi modellera med metoden Objektorienterad analys och design (OOAD). Vi beslöt oss för att utöka den ursprungliga dokumentationsdelen i metoden Prototyping till OOAD. Detta för att våra kunskaper inom OOAD är bättre än kunskaperna av den ursprungliga dokumentationsdelen. En ytterligare anledning är att vi ville göra en mer detaljerad och noggrannare modellering av vad som fanns som ursprungligt krav i Prototyping. Vi valde dock bort många delar av OOAD eftersom det annars skulle bli alltför detaljerat. Vi skulle få göra om prototypen en mängd gånger och har användare som riktlinje så att använda hela metoden OOAD kändes alltför omfattande. Vi använde de delar av OOAD som vi ansåg skulle vara till mest nytta för oss i vår programmering och följande delar användes: objektmodell, klassspecifikation, användarfallsdiagram och händelsetabell.

Efter modelleringen började programmeringen med de båda systemen. Programmeringen pågick först i två veckor för att få ihop en första prototyp. Prototyping innehåller då endast de mest nödvändigaste funktionerna. Detta för att kunderna skulle få påverka systemet så mycket som möjligt och därför ville vi inte utveckla systemet alltför mycket. Med systemet syftar vi på hela Internethemsidan för Klippotek Passagen som innehåller både bokningssystemet och e-handelssystemet. Prototypen sattes i bruk och sedan har den följts upp kontinuerligt och förändrats allteftersom kunderna kommer med ytterligare krav och önskemål. Vi har använt oss av Aktiv Server Pages (ASP) till bokningssystemet och PHP Hypertext Preprocessor (PHP) till e-handelssystemet. ASP hade vi goda kunskaper i innan, men vill också genom magisteruppsatsarbetet lära oss något nytt varför vi valde PHP.

### **3.4.4 Enkät**

Efter en första prototyp programmerats sattes den i bruk och enkäter skickades ut till ett antal kunder. Vid kontakt med frisören bestämdes att frisören skulle sätta de grundläggande krav, vilket gjordes vid första intervju, sedan skulle kunderna få styra hela utvecklingen tills systemet var klar. Detta eftersom det är kunderna som i slutändan ska använda systemet och därmed ska vara nöjda med det. Vi kom överens med frisören att han skulle övervaka hela utvecklingen genom att han efter varje prototyp skulle godkänna förändringarna innan vi skickade ut ytterligare enkäter till kunderna. Vi kom också överens om att det främst skulle vara stamkunderna som skulle medverka i utvecklingsarbetet.

Enkäter skickades ut via mail till de kunder som skulle ingå i vår undersökning. Vi bestämde att 30 stycken svar av varje enkätomgång skulle räcka därför att det är ett ungefärligt antal kunder som behandlas av en frisör på en vecka. Vi hade talat personligen med alla kunder som skulle vara med i undersökningen innan enkätutskicken och förklarat hur viktig deras medverkan var för att systemen skulle bli så bra som möjligt. Enkäten bestod av öppna, halvstrukturerade frågor för att ge kunden möjligheter att fritt svara och kommentera vårt bokningssystem och e-handelssystem. Enkäten finns som bilaga 2. Vi var ute efter så mycket kritik som möjligt för att kunna förbättra systemet inför nästa utskick. Vid mer strukturerade frågor skulle många åsikter riskera att inte komma fram, men samtidigt behövdes lite struktur för att visa på vad kunderna skulle fokusera på.

Efter varje enkätutskick gjordes en sammanställning på de synpunkter kunderna hade och vi förändrade prototypen helt efter deras önskemål med övervakning av frisören. Efter förändringarna var gjorda skickades en ny enkät ut. Andra enkäten var utformad på samma sätt som första enkäten och även här var fokus på kundernas krav på systemets funktionsduglighet och användarvänlighet. Vi hann endast med två enkätutskick vilket berodde dels på tidsbrist och dels på att kunderna verkade vara relativt nöjda med systemen redan efter första ändringen.

### **3.5 Utvärdering av Prototyping**

För att utvärdera systemutvecklingsmetoden Prototyping gjorde vi intervjuer med den frisör som var vår kontaktperson och de kunder som medverkat under utvecklingsprocessen. Detta för att utvärdera hur användarna upplevde utvecklingsarbetet med deras medverkan. Denna del tillhör också Backmans (1998) observationsfas där all data samlas in. Intervjufrågorna finns som bilaga 3 och 4.

Easterby-Smith et al (1991) skriver ”...the main reason for conducting qualitative interviews is to understand ...’how individuals construct the meaning and significance of the situations ... from ... the complex personal framework of beliefs and values, which they have developed over their lives in order to help explain and predict events in their world”.

Backman (1998) nämner också kvalitativa metoder och skriver om intervjuer och observationer ”instrumenten är mestadels ostrukturerade och siktar ofta mot en holistisk förståelse”.

#### **3.5.1 Intervju med frisören**

Syftet med intervjun var att ta reda på hur frisersalongen upplevde utvecklingsarbetet av systemen de har beställt. Vår intervju bestod av halvstrukturerade och öppna frågor och gjordes personligen med hjälp av bandinspelning och anteckningar.

#### **3.5.2 Intervju med kunderna**

Vi intervjuade även kunderna som hade haft en mycket stor roll i utvecklingsarbetet för att se hur de upplevde arbetet enligt Prototyping. Vi intervjuade 30 stycken kunder för att få ett så bra underlag som möjligt och för att få en holistisk bild av kundernas upplevelse. Dessa



kunder var de samma som svarat på våra enkäter. Intervjuerna gjordes även här personligen med halvstrukturerade och öppna frågor med hjälp av bandinspelning och anteckningar. De nio kunder som inte kunde närvara vid den personliga intervjun intervjuade vi via telefon.

### **3.6 Analys och Tolkning**

Backman (1998) skriver om analysfasen "När observationsfasen är avslutad måste data organiseras och systematiseras". Syftet med analysen är vidare "att ge överskådlighet och systematik". Härfter sker en tolkning av den data som samlats in och analyserats. Även Yin (1984) skriver om detta ämne och understryker vikten av att veta hur analysen av materialet går till. "The analysis of case study evidence is one of the least developed and most difficult aspects of doing case studies". Vidare skriver han om problemet att många inte vet hur materialet ska analyseras. Analysfasen är mycket viktigt eftersom det gäller att kunna bevisa sina slutsatser samt som Backman (1998) skriver "Alla de steg man tar i det totala forskningsarbetet måste vara adekvat för problemställningen, så att man får rätt svar på rätt fråga".

För att kunna ge struktur för alla svaren på intervjuerna från frisören och kunderna efter utvecklingsarbetet använder vi en analysmetod som beskrivs i Easterby-Smith et al (1991). "Analysis of the data resulting from this method of inquiry is generally accomplished by drawing up the questions on a speciell prepared matrix or analysis sheet". Frågorna skrivs överst på matrisen och varje svar från varje kund får sedan var sin plats under frågan. Denna metod gav oss överskådlighet som krävs för att kunna analysera svaren och se likheter, skillnader och mönster i användarnas svar. Genom denna metod analyserade vi allt vårt material från intervjuerna och drog slutsatser som finns redovisade i kapitel 5.

### **3.7 Validitet och Reliabilitet**

Easterby-Smith (1991) skriver att validitet och reliabilitet använts oftast inom kvantitativ forskning för att värdera forskningskvaliteten, men kan också användas för att validera kvalitativa metoder. För att avgöra validitet skrivs att följande fråga kan ställas: "Has the researcher gained full access to the knowledge and meanings of informants?" Vidare för att avgöra reliabilitet kan följande fråga ställas: "Will similar observations be made by different researchers on different occasions?"

Dessa två frågor är naturligtvis svåra att svara på. Vi har dock försökt att vara lyhörda vid intervjuerna för att kunderna ska få säga sin mening. Vid oklarhet har vi ställt följdfrågor för att ta reda på vad informanten vill få fram. Intervjuer har många svårigheter som Easterby-Smith (1991) tar upp och som måste beaktas vid användning av metoden. "There is the question of how accurate one's information is, and how accurate it needs to be, or can be.". Vi anser att användarna som vi intervjuade har varit ärliga i sin bedömning av systemet och hur de tycker arbetet fungerar eftersom det ligger i deras intresse att få ett system som är så bra som möjligt. Vi tror också att de varit ärliga i vad de tycker om Prototypingprocessen. En annan svårighet är hur strukturerade frågorna ska vara. Easterby-Smith (1991) skriver att strukturerade frågor sollar bort onödig information som kan komma fram vid helt öppna frågor och visar vad det är man vill ha för svar, men här finns dock risken att frågorna blir för strukturerade och information som varit nyttig att få inte kommer fram. Vi har använt halvstrukturerade och öppna frågor för att styra användarna åt rätt håll men ändå få deras

personliga åsikter. Vi har under intervjuerna försökt vara objektiva och inte ställa ledande frågor. Vi tror att andra forskare skulle ha fått samma resultat vid observationen vid samma tidpunkt.

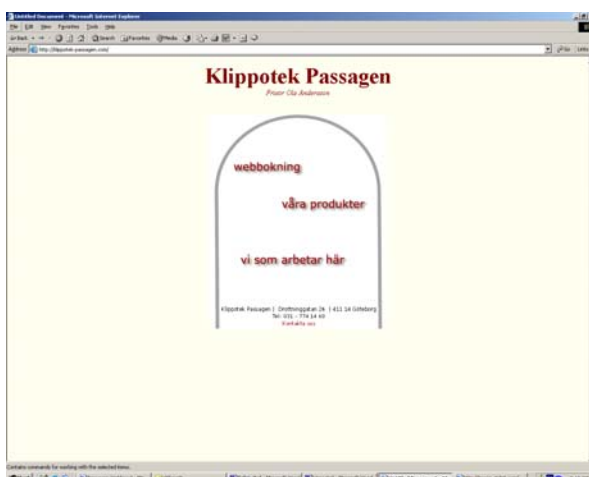
Vad gäller Prototypingprocessen har vi utökad den genom att använda Objektorienterad analys och design (OOAD) i utredningssektorn. Detta anser vi inte kan påverka resultatet av processen eftersom vi endast använt delar ur OOAD som uppfyller samma syfte som den ursprungliga dokumentationsdelen. Syftet med utredningssektorn är att få igång tankeverksamheten kring lösningen av systemet och härigenom få en grund inför programmeringen. Detta uppfyller vi även om vi inte använt den ursprungliga dokumentationsdelen i Prototypingprocessen.

För att verkligen säkerställa vårt resultat hur Prototyping fungerar i mindre system i mindre företag skulle naturligtvis krävas att vi testat Prototypingmetoden på flera system än två och också i flera mindre företag. Detta har ej kunnat genomföras på grund av tidsbrist. Vi har också endast gått två varv i cirkelmodellen för Prototyping, vilket kan bidra till att vi inte upplevt alla problem som kan uppstå. Detta gjordes också eftersom tiden inte räckte till och många kunder var nöjda redan efter två tester.

## 4 Fallstudie

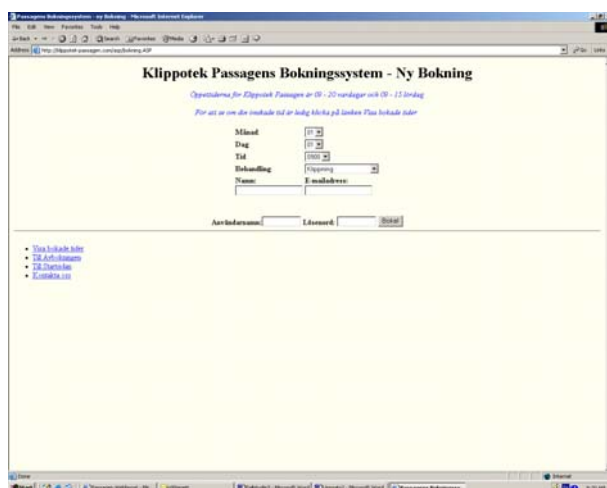
I detta avsnitt kommer vi att redogöra för hur prototypen för bokningssystemet och e-handelssystemet är uppbyggd och fungerar. Vi kommer dessutom att avsluta med ett användarscenario som förklarar kundens och frisörens nytta med systemet.

Både bokningssystemet och e-handelssystemet är webbaserade tjänster som är kopplade till en MySQL-databas. Själva startsidan som visas i figur 4 är enkelt utformad med endast tre länkar. Den första länken går till webbokningen, den andra till e-handelssidan och den tredje till en sida med bilder på personalen och miljön på Klippotek Passagen.



Figur 4 Huvudsidan för Klippotek Passagen

### 4.1 Prototypen för webbokningen



Figur 5 Klippotek Passagens bokningssystemet

Prototypen av bokningssystemet som visas i bild 5 är som sagt gjord i ASP (Aktiv Server Pages) och är alltså kopplad till en MySQL<sup>4</sup>-databas. Vi ansåg att webbokningen behövde ha ett lösenord för att undvika sabotage av bokningar. Varje kund tilldelas ett eget användarnamn och lösenord vilket gör bokningsapplikationen tillgänglig för att boka och avboka tider. Om

<sup>4</sup> SQL är en förkortning av Structured Query Language.

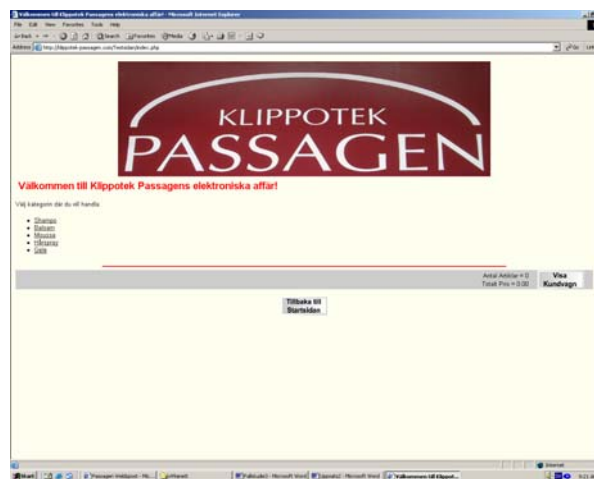
en kund vill ha ett lösenord tar kunden kontakt med frisersalongen via hemsidan eller genom ett besök på Klippotek Passagen. Kunden får då lämna några personliga uppgifter som namn, e-mail och telefon och sedan skickas användarnamn och lösenord till kunden. Bokningarna och uppgifterna om Klippotek Passagens kunder sparas i databasen. Frisörerna på Klippotek Passagen är administratörer i systemet och kan bland annat ta bort de bokningar som behandlats eller ändra uppgifter om kund. Det här sköts via MySQL-admin vilket är en tjänst som webhotellet har för att möjliggöra ett enkelt sätt att administrera databasen på.

Kunderna kan i webbokningssystemet boka tider, avboka tider och se sina bokade tider. Försöker kunden boka en tid som redan är upptagen kommer ett felmeddelande upp, vilket meddelar att tiden är bokad och kunden får försöka med en ny tid. Vidare erhålls felmeddelanden också vid fel användarnamn och lösenord samt om inte alla fält fyllts i. Vid avbokning gäller också att felmeddelande visas. Bokade tider kan ses och det är denna funktion som kunden ska starta sin bokning vid för att se om önskad tid är ledig.

Bokningssidan är utformad så enkelt som möjligt efter frisörernas och kundernas krav och önskemål. Tydlighet och användarvänlighet har fått dominera.

## 4.2 Prototypen för e-handel

Prototypen för e-handelssystemet är gjord i PHP<sup>5</sup> och även här har används en MySQL-databas. Detta system är uppbyggt i så kallade sessioner, vilket används för att temporärt spara information knuten till en viss användare. Beställningen skall skrivas in i databasen först när kunden handlat färdigt alla sina varor och verkligen bestämt sig.

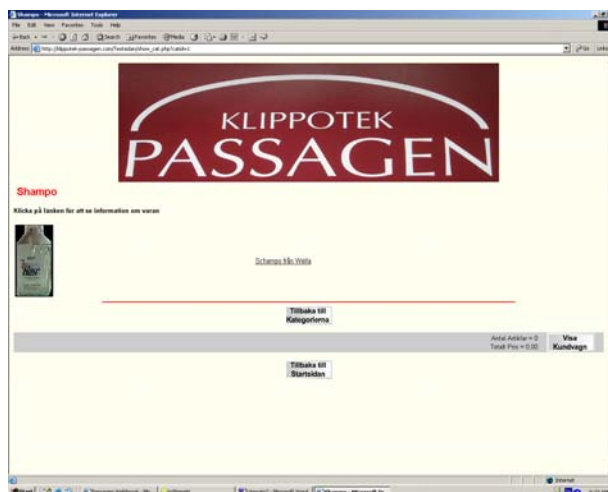


Figur 6 Klippotek Passagens e-handelssystem

När en kund kommer till startsidan för e-handelssystemet som visas i figur 6 görs ett val i vilken kategori kunden vill handla. Kunden kan därefter enkelt gå in och välja önskad vara. Genom att klicka på länken till vald kategori kommer varorna upp med bild och ännu en länk, som visas i figur 7.

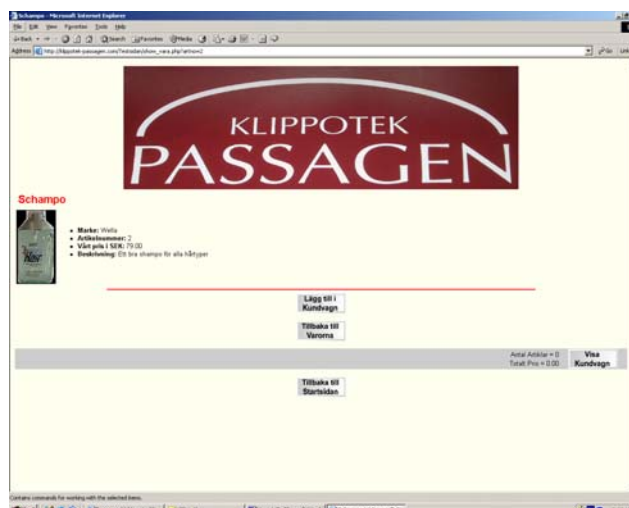
---

<sup>5</sup> PHP Hypertext Preprocessor



Figur 7 Klippotek Passagens e-handelssystem

Därpå får kunden möjlighet att välja vara genom att klicka på länken till den önskade varan. Både bild och text visas för att underlätta för kunden att känna igen sin produkt. Ofta kommer kunden ihåg färg och logotyp alternativt varumärke på den produkt han/hon varit nöjd med. Ovanstående var också ett önskemål från våra kunder under den enkätundersökning som genomfördes.



Figur 8 Klippotek Passagens e-handelssystem

När kunden valt vara genom att klicka på länken kommer detaljerad information om varje vara upp som kan ses i figur 8. Vi anser att det är viktigt att kunden tar del av denna information eftersom många produkter är väldigt lika varandra i utseende. Därför har vi byggt ett system på detta sätt så kunden måste läsa informationen om varan innan den läggs i varukorgen. Det här kan uppfattas som lite krångligt av användarna men på så sätt undviker vi en mängd returter på grund av att kunden tror sig beställt rätt produkt och sedan blir besviken när han/hon upptäcker att varan inte är den som förväntats. Om en mängd felaktiga varor köps in avhjälpas inte heller problemet med att sänka lagerkostnaderna i största möjliga mån för Klippotek Passagen, eftersom de då riskerar att få många returter.

Efter att kunden lagt till önskade varor i sin kundvagn fyller han/hon i ett beställningsformulär. Om kunden inte fyller i alla fält kommer ett felmeddelande upp. Detsamma gäller också om du inte har några varor i kundvagnen då kommer också ett felmeddelande upp som talar om att kundvagnen är tom och då har kunden inget att betala.

Därefter kommer nästa formulär upp som specificerar beställningen och frågar om kunden önskar hämta varorna på salongen eller få dem skickade med post. När det här valet är gjort skickar kunden sin beställning och köpet är avslutat. Härfter kommer ett formulär som repeterar alla uppgifter angående beställningen såsom beställda varor och hur varan ska fås av kunden. Önskar kunden en bekräftelse kan sista sidan sparas ner på datorn och/eller skrivas ut.

### **4.3 Teknisk beskrivning**

Båda systemen har en klient-server arkitektur. Klienten, vilket är kundens dator, ger kunden möjlighet att från sin dator skicka information till servern vilken finns på webbhotellet. Servern handhar de resurser som traditionellt finns hos stor- och minidatorer. Detta är bland annat resurser som kraftig bearbetningskapacitet, möjlighet att lagra och hantera stora datamängder. Klienten och servern arbetar tillsammans för att åstadkomma önskat resultat vilket innebär att den totala datakraften utnyttjas härmed på ett effektivt sätt. (Brown 1997)

Den webbserver som systemen ligger på är en Apache-server. Fördelar med denna servertyp är enligt webbhotellet bland annat att den är vältestad på grund av sin stora spridning, det är öppen källkod och det finns bra konfigurationsmöjligheter. Att det är öppen källkod brukar kallas för Open Source vilket innebär att koden är tillgänglig för alla och den är dessutom gratis att använda (Ljungberg 2000).

ASP som används i bokningssystemet är en teknologi som utvecklats av Microsoft. Syftet med ASP är att ASP-koden bakas in i HTML-koden för att på så sätt kunna göra hemsidan dynamisk. Dynamiken är viktig för att sidan ska kunna följa med när förändringar sker. Exempelvis om två kunder är inne på samma gång så skall ena kundens precis bokade tid visas på den andra kundens skärm. På så sätt skapas en kommunikation med användaren med hjälp av ASP. ASP är inte ett renodlat programmeringsspråk utan en lös samling objekt som kan kommunicera med servern och användarens webbläsare. Sedan används ett valfritt skriptspråk som till exempel JavaScript, JScript eller som Microsoft rekommenderar VBScript för att kunna använda objekten. Skillnaden ligger i att när skriptspråken används med ASP exekveras koden på servern istället för hos klienten. Vi har använt oss av VBScript i vårt bokningssystem. (Ronne 2000)

PHP som används i e-handelssystemet är ett skriptspråk som också exekveras på serversidan ([www.php.net](http://www.php.net)). Det är speciellt designat för att användas på webben. På samma sätt som ASP bakas koden in i HTML-koden och exekveras varje gång sidan används. PHP-koden översätts på webbservern och genererar det användaren får se. PHP har expanderat explosionsartat och år 2001 var det över 5 miljoner domäner som använde det över hela världen. PHP är en också en Open Source produkt. (Welling och Thomson 2001)

När det gäller databasen finns det en mängd olika varianter att välja på. Vi valde just MySQL för de möjligheter som finns att gå in i databasen via MySQL-admin, vilket är en tjänst som finns på webbhotellet och där finns möjlighet att utföra de operationer och eventuella ändringar som fordras. MySQL är en tillgänglig Open Source produkt men finns även att köpa kommersiellt. Fördelarna med MySQL är att den bland annat är multi-user, multi-threaded vilket innebär att flera användare kan göra sin beställning samtidigt. Via aktuellt webbhotell finns möjligheten att via MySQL-admin lägga till, ta bort eller ändra poster i databasen. Detta

har medfört att det ej behövs bygga in någon administratörsfunktion i systemen. MySQL är också mycket använd ihop med webbapplikationer (www.mysql.com).

## **4.4 Användarscenario**

### **4.4.1 Kund**

Lena tittar sig i spegeln och märker att hon behöver klippa sig. Eftersom klockan är över 20.00 och salongen antagligen är stängd använder sig Lena av tjänsten på Klippotek Passagens hemsida. Hon börjar med att titta i sin almanacka vilken dag hon anser vara bäst. Sedan klickar Lena på länken webbokning och klickar därefter på länken visa bokade tider. Där får hon upp de tider som är bokade och ser till sin förtjusning att tiden den 6/6 klockan 15.00 är ledig. Hon bokar tiden och får en bekräftelse på att bokningen är registrerad. För att inte glömma bort sin tid skriver Lena ut bokningen och sätter upp den på sin anslagstavla. Förra gången Lena var hos frisören provade han en ny produkt på henne som hon blev mycket nöjd med. Lena tänker att det skulle vara bra att ha en sådan produkt hemma och klickar på länken våra produkter. Efter det hon följt de olika stegen i köpprocessen kommer hon till sista sidan och där bestämmer hon sig för att hämta produkten på salongen samtidigt som hon skall klippa sig. Lena är nöjd och glad och ser fram mot sitt besök på Klippotek Passagen.

### **4.4.2 Frisör**

Frisören kommer till Klippotek Passagen på morgonen och slår på datorn och kontrollerar bokningarna som kommit in över natten. Han går även in och kontrollerar dagens beställningar på MySQL-admin vilket är en tjänst på webbhotellet där man kan administrera och ställa frågor till databasen. Därefter går han upp till Frick & Co som är en frisörgrossist på Drottninggatan i Göteborg och gör dagens inköp av beställda varor. Då han kommer tillbaka till salongen med varorna gör han ordning kundernas varor och lägger dem i påsar. Nu är klockan 09.00 och den första kunden kommer. Vid tidsbokning bokar varje kund en timma. Detta görs för att frisören eventuellt skall kunna ta mer än en kund samtidigt och på så sätt kunna ta in så mycket inkomster som möjligt. Om till exempel kund nummer ett beställt klippning och färgning kan denna sitta med färgen medans kund nummer två blir klippt och vice versa. En klippning tar i de flesta fall inte mer tid än mellan 30 och 45 minuter. Då telefonen ringer och en kund vill boka skriver frisören givetvis in bokningarna på hemsidan för att kunderna på ett lätt sätt skall kunna se vilka tider som är lediga. På så sätt är bokningen öppen 24-timmar per dygn vilket är en bra service för både kund och frisör.

### **4.4.3 Nyttan med systemet**

Kunden har nytta av bokningssystemet genom att han/hon mycket enkelt kan boka och avboka tider. Kunden behöver inte längre sitta i kö i telefonen för att boka tid utan det går snabbt och enkelt att gå in på hemsidan och där ser kunden om önskad tid är ledig och i så fall bokas den tiden.

Kunden kan i e-handelssystemet skicka e-mail till frisören om han vill ha en viss märkesprodukt som inte fanns i det gamla lagret. E-handelssystemet är tänkt att kunna erbjuda

just de produkter som kunderna önskar och därmed utöka servicen till kunderna. E-handelssystemet erbjuder alla varor som frisörerna kan få tag på genom sina grossister. Kunden kan bestämma om han vill ha produkten via post eller hämta den själv vid besöket på salongen.

Frisören behöver inte längre ligga ute med stora lagerkostnader för produkter utan köper in endast de produkterna som beställts via e-handelssystemet. Dessutom får han mer tid över till sina kunder om telefonen inte ringer lika mycket längre. Givetvis går det fortfarande att ringa in sin beställning men tanken är att telefonsamtalen kommer att minska väsentligt. På så sätt är det tänkt att frisören skall kunna tjäna både tid och pengar genom att använda sig av sin interaktiva hemsida.



## 5 Resultat

För att kunna beskriva vårt förarbete inför uppsatsen kommer inledningsvis en kommentar av resultatet från litteraturstudien. En förståelse av Klippotek Passagens dagliga arbete och behov av Internetetablering presenteras här efter med hjälp av resultatet från en inledande intervju med frisören samt observationerna som utfördes på frisersalongen. Här efter redogörs också för det resultat som modelleringen gav samt en summarisk redogörelse av resultatet från enkäterna och dess inverkan på den utvecklade prototypen. Därefter kommer en sammanfattning av hur väl Prototyping fungerar i småföretag baserad på intervjuer efter systemutvecklandet med frisören och kunderna. Slutligen kommer ett avsnitt som sammanfattar vårt resultat.

### 5.1 Litteratur

En omfattande litteraturstudie gjordes i början av arbetet med magisteruppsatsen. Resultatet av studien representeras av vår teoretiska referensram. Några reflektioner gjordes dock under litteraturstudien. Trots att vi lade ner mycket tid till att hitta bra litteratur upplevde vi det som mycket svårt att hitta litteratur som passade vårt ämne. Det mesta av vår litteratur är äldre och handlar om traditionell systemutveckling eller systemutveckling i allmänhet. Prototyping var ingen omskriven systemutvecklingsmetod, vilket förvånade eftersom de flesta systemutvecklingsböcker nämner begreppet men ger ingen ingående redogörelse om innebörden av metoden. Det var endast Mattsson et al (1988), Ahlandsberg och Sandström (1988) och Burman och Bäckman (1992) som gav oss lite mer information om vad som var känt om Prototypingmetoden.

Vi sökte också litteratur och artiklar på Internet och bibliotek med blandat resultat. Prototyping visade sig vara ett mycket vanligt begrepp vare sig det var systemutveckling, matlagning eller teknik. Efter kontakt med författaren till boken Objektorienterad analys och design Lars Mathiassen fick vi dock tips på en artikel av Rickard L. Baskerville och Jan Stage (1996), vilket gav oss referenser till bra och ny litteratur i ämnet Prototyping.

### 5.2 Intervju med frisören innan utvecklingsarbetet

En intervju med frisören hade två syften. Dels ville vi reda ut vad frisörerna förväntade sig och var ute efter för system och dels ville vi ta reda på rådande omständigheter på frisersalongen för att kunna anpassa systemet därefter. Intervjun med frisören tillhör utredningssektorn som skapar en grund inför modelleringen av systemet. Utredningssektorn beskrevs i den teoretiska referensramen.

Kraven på bokningssystemet beskrevs av frisören:

*”Det skall vara ett enkelt och lättförståeligt system där kunderna kan boka, avboka och se sina bokade tider. Det skall givetvis säga till vid dubbelbokning”.*

Vi hade nu alltså kraven att systemet ska kunna boka in kunder, avboka kunder och säga till när en kund försöker boka en redan upptagen tid. Detta innebär att systemet borde ha någon

form att sida som visar vilka tider som är bokade redan så kunder slipper chansa på en tid. Vidare bör det finnas ett formulär för dem att fylla i för att boka respektive avboka sin tid.

Vi fick också en beskrivning av e-handelssystemet.

*”Vi vill att kunden skall kunna beställa produkter. Tanken är att vi på så sätt skall ha ett betydligt större sortiment efter våra kunders önskemål. På så sätt behöver vi inte ligga på stora lager utan köper in allteftersom beställningar kommer in. Vi kan på så sätt ha alla kända märken som kunderna önskar. Denna beställning ska sedan antingen skickas med post till kunden via postförskott eller hämtas på salongen”.*

Om vi översätter svaret från frisören i krav så vill de ha en sida som visar de produkter frisören kan få tag på och köpa in åt kunden om kunden så önskar. Kunden ska kunna beställa dessa varor på systemet, men inte kunna betala dem via systemet. Betalningen ska ske vid antingen postförskott eller betalas kontant till frisören beroende på om kunden vill ha varan hemskickad eller hämta den på salongen. Detta innebär i princip ett fungerande e-handelssystem där kunden orienterar sig bland olika varor och lägger utvalda varor i någon typ av kundkorg. Kunden kan tänkas behöva någonstans fylla i uppgifter som namn, adress och leveransadress. Någonstans måste kunden också ange om den vill hämta sina beställda varor på salongen eller få dem hemskickade med post.

Vi ställde också frågor om situationen på frisersalongen om deras bokningsrutiner och lagerhantering. Omständigheterna kring det nuvarande arbetssättet på frisersalongen visade sig skapa en del olägenheter inte bara för frisörerna i sitt arbete utan också för kunderna.

*”Idag sker bokningen genom telefon vilket gör att vi ofta blir avbrutna i vårt arbete och på så sätt blir det lättare förseningar i tidsschemat. Dessutom finns det vissa behandlingar som är svåra att avbryta vilket kan leda till att vi helt enkelt inte kan svara just då. Ett annat problem är när alla är lediga då går det inte att boka tider”.*

Att frisörerna inte ens kan svara vid vissa behandlingar gör att de lätt kan tappa kunder och bli mindre konkurrenskraftiga i en redan stor konkurrens om kunder. Frisersalongen har vidare inga avsatta tider då de enbart tar emot bokningar utan kunder kan ringa så länge frisersalongen är öppen.

*”Även om en del av kunderna bestämmer sig för att ringa kommer det underlätta om en viss del bokar via nätet.”*

Frisören vi intervjuade hoppas att bokningssystemet kommer att ge dem nya fördelar som att de kan arbeta mer ostört och att de kan undvika förseningar i schemat. Kunderna kan med ett bokningssystem erbjuda kunder service dygnet runt via en webbsida där de kan boka tider när det passar dem.

I nuvarande rutiner kring lagerhantering finns olägenheten att en frisör får ligga ute med en stor summa pengar för varor med tidsbegränsad hållbarhet eller en lägre efterfrågan än vad som kunnat förutspås. Det här var ett problem frisersalongen ville få bukt med.

*”Vi ligger på lagerkostnader på ca 10 000 kr i månaden för olika typer av försäljningsprodukter. Tyvärr är det svårt att veta vad som går och det blir ofta att vi får sälja ut med förlust som följd”.*

Med ett e-handelssystem där kunden kan gå in och beställa behöver aldrig en frisör ligga ute med pengar utan kan köpa in just de varor kunden beställt och som skickas till kunden eller finns på frisersalongen för kunden att hämta.

Klippotek Passagen verkar kunna få fördelar av att etablerat sig på Internet. En webbsida för bokning och beställning av varor ger dem större konkurrenskraft och visar att de hänger med i utvecklingen.

*”Det finns en mängd frisörer som har hemsidor och konkurrensen är ytterst stor. Bara på vår gata finns det 14 frisersalonger.”*

### **5.3 Observationer**

För att få en djupare förståelse för situationen på frisersalongen Klippotek Passagen och få en förståelse för vad de förväntar sig systemen ska lösa för problem gjorde vi inledningsvis tre observationer vid frisersalongen. Observationerna tillhör precis som intervjun med frisören utredningssektorn som beskrevs i den teoretiska referensramen. Frisören hade vid intervjun berättat om de störande momenten som finns i sitt arbete. Detta var bland annat kunder som ringer mitt i deras arbete och vill boka tider. Det var inte heller alltid som frisörerna kunde svara och detta var irriterande för kunder som ringde utan att få svar. Dessa störande moment vill frisörerna ta bort eller åtminstone minska genom att skapa ett system på Internet där kunder själva kan gå in och boka sig eller köpa varor utan att behöva ringa till Klippotek Passagen.

Våra observationer bekräftade vad frisören berättade vid intervjun. Telefonen ringde åtskilliga gånger varje timme och ett fåtal av alla samtal kunde frisörerna inte ta på grund av att de var mitt i en behandling av en annan kund. Det fanns också gånger de svarade och då fick kunderna i salongen vänta tills samtalet var klart. Det här skapade också irritation hos en del av kunderna som var under behandling eftersom behandlingen tog längre tid än väntat.

Vi såg också ett antal varor som schampo och skumbalsam som såldes ut på grund av både utgående datum och dålig efterfrågan på produkten. Två kunder köpte några av dessa varor för utförsäljning under tiden vi var där, samtidigt som ingen köpte varor som såldes för fullt pris. En annan kund frågade en av frisörerna efter ett visst schampo av märket KMS, vilket frisersalongen inte hade i lager. Resultatet blir att kunden går till en av konkurrenterna och köper produkten.

Det var uppenbart ett ineffektivt arbetssätt och besvärande att behöva ta emot samtal. Försäljningen av produkter fungerade heller inte tillfredställande. Observationerna gav oss större insikt i vad frisersalongen var ute efter och vilka problem de ville lösa. Detta gav oss underlag till vad vårt bokningssystem och e-handelssystem skulle klara av och vilka krav de skulle tillgodose.

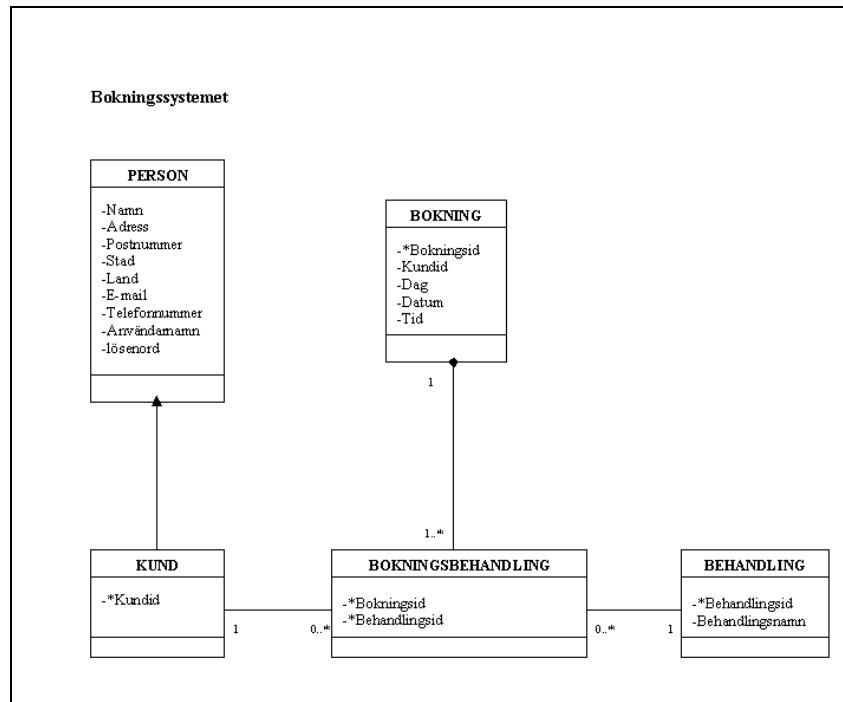
## 5.4 Modellering

Under utvecklingstiden för bokningssystemet och e-handelssystemet skapades en mängd kod och olika modeller av systemen som hjälp inför programmeringen. Modelleringen gjordes främst innan första prototypen skapades och därmed innan all kod skrevs. Allteftersom prototyperna ändrades kom också modellerna att ändras därefter. De olika modellerna ska verka som underlag för att kunna skapa systemen på bästa sätt. Att modellera det framtida systemet med vald modelleringsteknik gör att utvecklaren planerar och analyserar inför programmeringen. Detta för att underlätta och eliminera framtida problem som lätt kan uppstå annars vid programmeringen. Vi använde alltså Objektorienterad analys och design som metod för modelleringen och valde ut de delar som vi tyckte var lämpliga. Mindre system som vårt bokningssystem och e-handelssystem har inte den komplexiteten som eventuellt större system kan ha. Därför anser inte vi att all förarbete som den objektorienterade metoden innebär är nödvändig. Vi valde att göra objektmodell, klassspecifikation, användarfallsdiagram och händelsetabell. Modelleringen tillhör utredningssektorn vilket beskrivs i den teoretiska referensramen. Här nedan visas resultatet av dessa som vi arbetade fram innan programmeringen.

### 5.4.1 Objektmodell

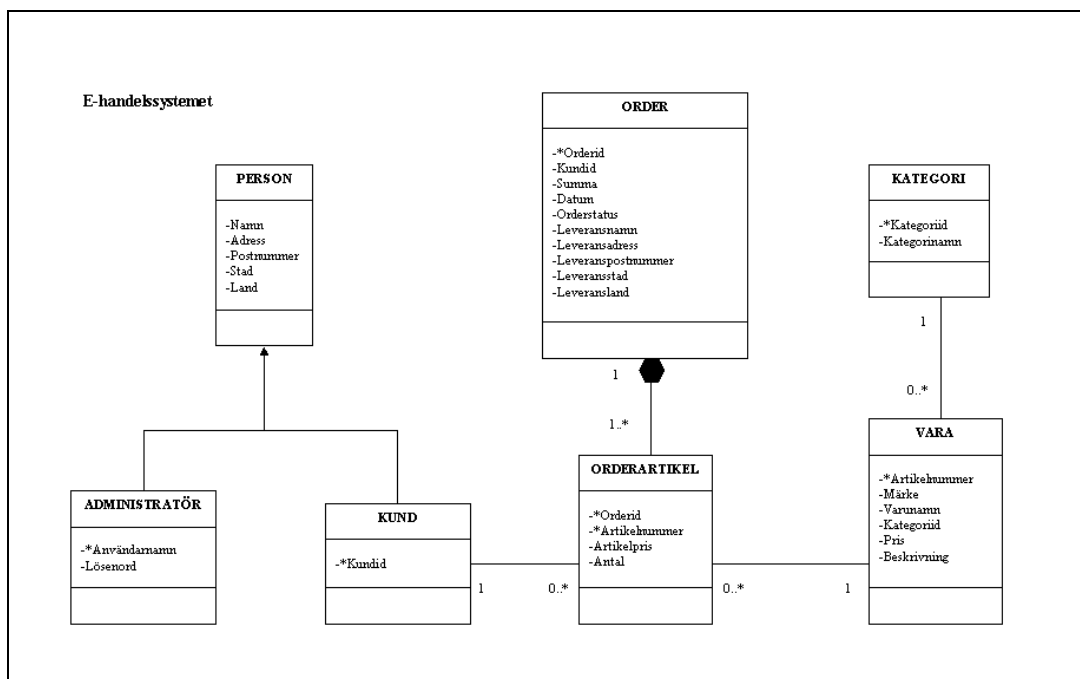
En objektmodells syfte är att underlätta för programmeringen genom att utvecklarna innan har tänkt igenom vad som ska vara med i systemet i form av klasser och vilka relationer som ska råda däremellan. Mathiassen (2001) skriver om objektmodellen att den ”ger en sammanhängande överblick över problemområdet genom att beskriva alla strukturella relationer mellan klasserna och objekten i vår modell”. Objektmodellen görs i Analysen av problemområdet under aktiviteten Struktur som vi beskriver i den teoretiska referensramen.

Objektmodellerna gjordes direkt efter intervjun med frisören och efter observationerna på Klippotek Passagen. En bild av systemen erhöles och nödvändiga klasser och attribut arbetades fram. Vi följde Mathiassens (2001) instruktioner från boken Objektorienterad analys och design och kom fram till att bokningssystemet skulle innehålla klasserna Person, Kund, Bokning, Bokningsbehandling och Behandling vilket visas i figur 9.



Figur 9 objektmodell av Klippotek Passagens bokningssystem

E-handelssystemet är lite större i sitt omfång och antal klasser blev därmed större. E-handelssystemet innehåller klasserna Person, Administratör, Kund, Order, Orderartikel, Kategori och Vara vilket visas i figur 10.

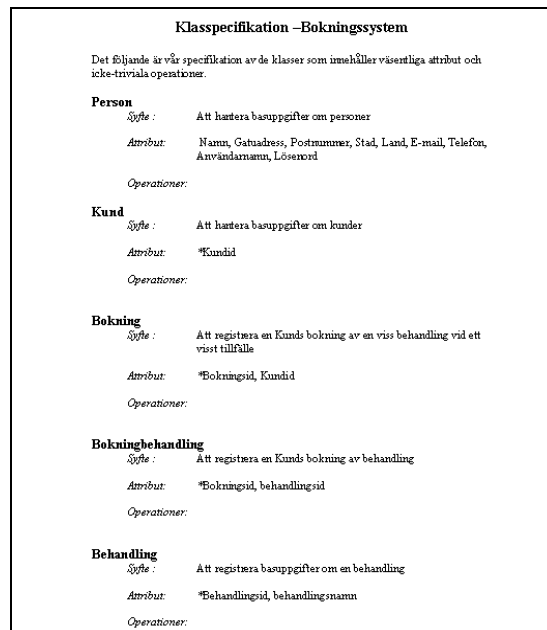


Figur 10 Objektmodell av Klippotek Passagens e-handelssystem

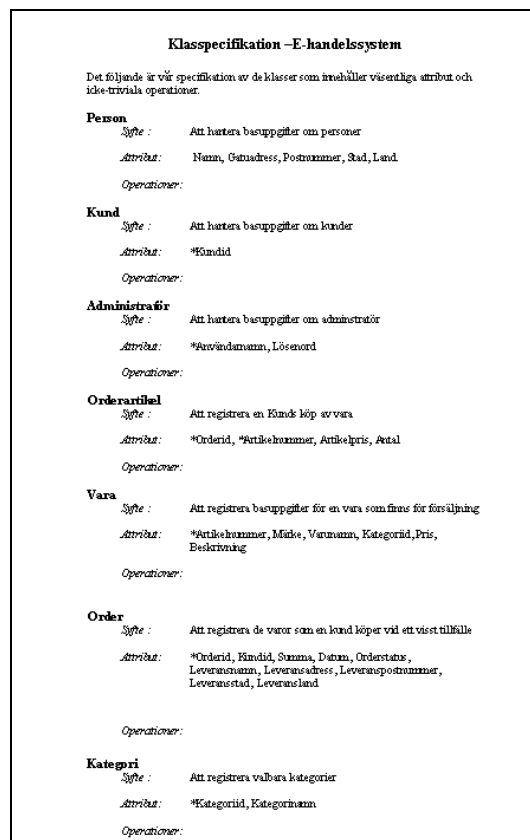
## 5.4.2 Klassspecifikation

Syftet med en klassspecifikation är att specificera vilket syfte klassen har samt vilka attribut och icke triviala funktioner klassen innehåller. Utgångspunkten är därmed objektmodellen och

dess klasser beskrivs i denna klassspecifikation som visas i figur 11 och 12. Klassspecifikation skrivs efter objektmodellen och detta sker under Analys av problemområdet under aktiviteten Klasser som vi beskriver i den teoretiska referensramen.



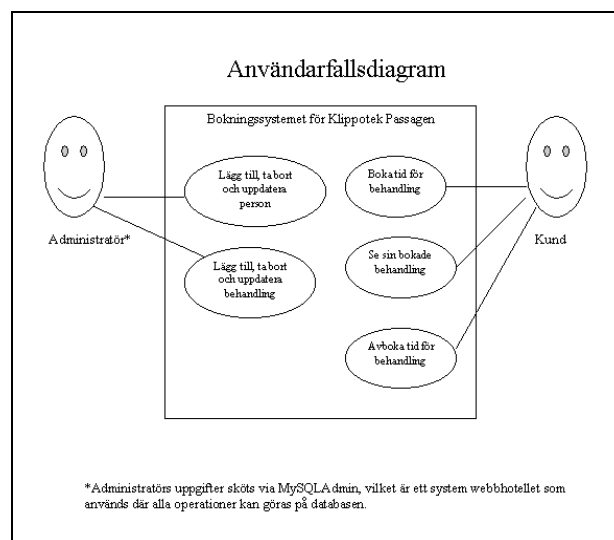
Figur 11 Klassspecifikation för Klippotek Passagens bokningssystem



Figur 12 Klassspecifikation för Klippotek Passagens e-handelssystem

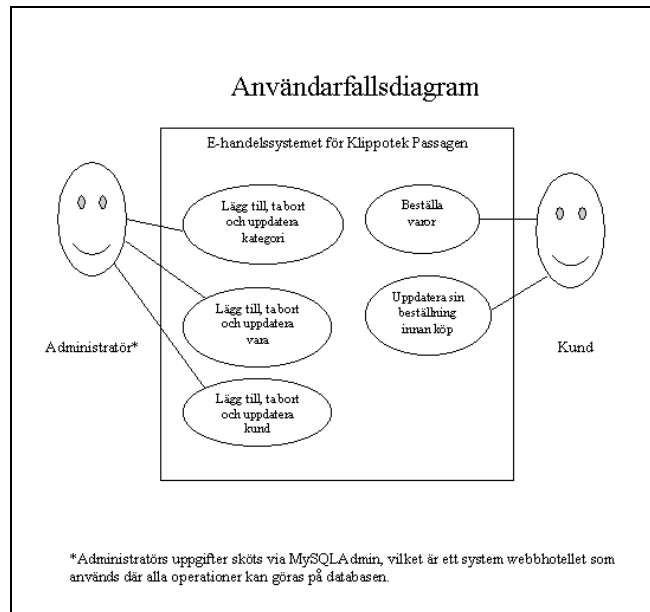
### 5.4.3 Användarfsdiagram

Syftet med ett användarfsdiagram är att hitta och ange alla användarfsfall, vilket innebär att hitta alla möjliga saker aktörerna i systemet kan göra. En aktör är personer eller andra system som integrerar med systemet i fråga. I vårt fall finns kund och administratör som aktörer till bokningssystemet som visas i figur 13. Administratören är frisören när systemet tas i bruk. Administratören ska kunna lägga till, ta bort och uppdatera person och behandling. Detta kommer inte att ske via systemet utan via det webbhotell som frisersalongen har. En kund ska däremot med systemet kunna boka, avboka och se sin bokade behandling. Mathiassen (2001) skriver ”Användarfsfall ger en överblick över systemkraven från användarnas perspektiv och ger en grundval för att definiera och utvärdera de viktigaste kraven på funktioner och gränssnitt”. Denna del tillhör Analys av användningsområdet och aktiviteten Användning som beskrivs i den teoretiska referensramen.



Figur 13 Användarfsdiagram för Klippotek Passagens bokningssystem

E-handelssystemet har också endast kund och administratör som aktörer till systemet vilket visas i figur 14. Även här är det via webbhotellet som administratören ska kunna lägga till, ta bort och uppdatera kategori, vara och kund. Kunderna ska i detta system kunna enbart beställa varor och uppdatera sin beställning innan köp. Det sistnämnda innebär att kunden ska kunna lägga till fler eller ta bort varor i sin beställning innan beställningen är skickad till frisersalongen.



Figur 14 Användarfsdiagram för Klippotek Passagens e-handelssystem

#### 5.4.4 Händelsetabell

Syftet med en händelsetabell är att reda ut vilka händelser som berör vilka klasser. En händelse är något som sker i systemet och i detta fall det kunden kan göra i systemet. Händelsetabell är en utmärkt del att utföra vid komplicerade och komplexa system, då det tvingar utvecklaren att reda ut klasser, funktioner och dess samband. Händelsetabell görs vid Analys av problemområdet under aktiviteten Klasser som beskrivs i den teoretiska referensramen.

I figur 15 visar vi att händelserna kunden kan aktivera påverkar klasserna Kund, Bokning och Bokningsbehandling. Endast dessa tre klasser är med i tabellen eftersom vi bedömde dem som mest intressanta att reda ut. Figur 16 visar också att händelserna kunden aktiverar påverkar många klasser. I detta fallet påverkas också alla klasser vi valde ut att ha med i tabellen.

|Händelsetabell för bokning

Händelser	Klasser som påverkas av händelserna		
	Kund	Bokning	Bokningsbehandling
bokar	X	X	X
avbokar	X	X	X
se sin bokning	X	X	X

Figur 15 Händelsetabell för Klippotek Passagens bokningssystem



Händelsetabell för e-handelssystem					
Klasser som påverkas av händelserna					
Händelser	Kund	Order	Orderartikel	Kategori	Vara
Beställa vara	X	X	X	X	X
Uppdatera sin beställning	X	X	X	X	X

Figur 16 Händelsetabell för Klippotek Passagens e-handelssystem

## 5.5 Prototyp

Första prototypen skapades efter intervjun med frisören, observationer på Klippotek Passagen och modelleringen av verksamheten. Frisörerna specificerade i grova drag vid intervjun vad det var de var ute efter. Vi skapade en prototyp efter deras krav som kom fram under intervjun och därefter var det kunderna som vägledde oss genom utvecklingsarbetet med övervakning av frisören. Den första prototypen innehöll alla funktioner som vi genom intervjun med frisören hade bestämt. Denna prototyp ändrades sedan efter kundernas önskemål. Utvecklandet av systemen tillhör konstruktionssektorn och själva installationen av systemen tillhör införandesektorn som beskrevs i den teoretiska referensramen.

Första prototypen som installerades på Klippotek Passagen innehöll som sagt krav som vi tolkat från intervjun med frisören. Denna första prototyp innehöll åtskilliga misstag. Vi hade stavfel i många felmeddelanden, blandade svenska och engelska främst vad gäller knappnamn och många instruktioner som behövdes saknades. Detta angavs på de enkätsvar vi fick in. Vi fick mycket respons och kritik på självklara saker som helt enkelt glömts bort.

En reflektion vi gjorde var att kunderna hade en del synpunkter om tillvägagångssättet för att utföra vissa funktioner. Till exempel så var de missnöjda vid bokningssystemet med tillvägagångssättet för att boka en tid. Vi upptäckte också att det var skillnad i svaren mellan de med datavana och de utan större datavana. De utan datavana tyckte det mesta var bra samtidigt som de med större datavana efterlyste andra lösningar på vissa funktioner.

Förändringarna som skapade en andra prototyp var främst utseendeförändringar för båda systemen. Funktionerna formades till första prototypen och dessa funktioner tycktes räcka långt, även om tillvägagångssättet för att utföra vissa funktioner skapade något missnöje. Det var endast några kunder som önskade fler funktioner och då var det funktioner som var mindre. Exempelvis så önskade någon kund att man kunde få upp en ruta med sina bokade tider listade. Kunderna kunde med prototypen gå in och titta på vilka tider som var bokade, men prototypen listade alla kunders bokade tider och inte endast vald kunds.

Andra prototypen innehöll till vår förvåning också stavfel och andra slarvfel. Andra prototypen verkade göra kunderna tillfreds och mycket av kritiken vi fick vid andra enkätutskicket var småsaker. Några kunder klagade exempelvis på att det inte fanns nog med varor att orientera sig runt bland. Dock kom fler önskemål vid andra enkätutskicket om fler funktioner. Vid första enkätutskicket fanns bara utseendeförändringar bland önskemålen, men nu fanns alltså även önskemål om fler och utökade funktioner. Några kunder hade önskemål att få en bekräftelse på sin beställning vid e-handelssystemet hem via e-mail. Det önskades

också av bokningssystemet att ett nytt fönster skulle komma upp med de redan bokade tiderna.

På grund av tidsbrist och att önskemålen vid andra enkätutskicket blev väsentligt färre bestämde vi att en tredje prototyp skulle bli den sista och att det var den som ska sättas i bruk vid Klippotek Passagen. Vi ändrade våra stafvel och andra uppenbara slarvfel en sista gång och satte tredje prototypen i system.

## **5.6 Intervju med frisören efter utvecklingsarbetet**

En intervju gjordes med frisören som varit vår kontaktperson på Klippotek Passagen för att undersöka hur frisören upplevt utveckling med Prototyping som inneburit ett nära samarbete med användarna.

Vi började med att fråga om frisören överhuvudtaget känt sig delaktig och i så fall på vilket sätt.

*”Jag har känt mig delaktig och givetvis varit angelägen att systemen vi gör skall bli så bra som det bara går. Jag har kommit med förslag och synpunkter på saker som jag tror förbättrar systemen. Sedan har jag lämnat över till utvecklarna för att de skall värdera det jag tyckt till om.”*

Vi har varit angelägna om att det ska vara en positiv upplevelse för användarna att vara med och påverka utvecklingsarbetet och försökt att göra kommunikationen så lättvindig som möjligt. Naturligtvis finns det risk att det kan uppfattas av användarna som krävande arbete eftersom Prototyping oftast innebär att användarna får vara med ett antal gånger innan systemen är färdigutvecklade.

*”..det känns bra när det man anser skall ändras åtgärdas och blir både snyggt och bra. Mina krav har tagits om hand på ett bra sätt och jag har själv sett förändringen som skett.”*

Det verkar dock som att motivation att vara med i utvecklingsprocessen skapas av att användaren ser en förändring av det som användaren angivit som sämre. Detta är en process som oftast uteblir vid traditionell systemutveckling som oftast utgår från en kravspecifikation där användarna får vara med att ange krav och önskemål men sen oftast inte har någon större chans att påverka utseendet eller tillvägagångssättet vid vissa funktioner. Att få användaren med i processen från kravspecifikation till färdigt system kan ge en större tillfredsställelse hos användaren.

*”Det känns bra och det är både stimulerande och motiverande att vara med och utveckla ett system som man själv tror skall hjälpa både oss som arbetar på salongen och givetvis ge fördelar till våra kunder i form av att de kan boka tider när de vill och att de får ett betydligt bredare sortiment med varor att välja på.”*

Ett använda Prototyping som systemutvecklingsmetod gör att det oftast finns en svårighet i att uppskatta tidsåtgången. Vi hade mycket svårt att avgöra hur lång tid processen skulle ta och hur många omgångar vi skulle behöva involvera användarna för att de skulle vara nöjda med

systemen. Systemutvecklingen kan med andra ord ta mycket längre tid än man förväntat sig och kan då återigen vara mycket krävande för användarna.

*”Det har tagit betydligt mer tid av mig än jag trodde från början. Men samtidigt har det blivit som vi själva vill ha det.”*

Att involvera användare i processen upplever vi ge många fördelar. Bland annat så minskar inlärnings tiden för kunderna att lära sig systemen.

*”För mig känns det mycket bra att jag och mina kunder utvecklat systemen tillsammans. Kunderna har på så sätt lärt sig systemen och det blir inget problem för dem att använda det när det börjar användas.”*

Alla kunder har självklart inte varit med, men om nu en representation av kunderna varit med och utformat systemen är förhoppningen att de är så enkla och välformade att resten av kundkretsen också kan tänka sig att använda systemen.

Trots nackdelar som tidsaspekten så verkar frisören uppleva systemutvecklingen som positivt. Frisören var med i början innan utvecklingen börjat för att ge oss krav och önskemål på de båda systemen. Under hela utvecklingen har sedan i stort sett endast kunderna varit med och påverkat. Efter varje ny prototyp har vi meddelat frisören att ändringar har skett och sedan stämt av vid ett senare tillfälle för att få ett godkännande från frisören om ändringarna som skett efter kunderna önskemål. Detta arbetssätt bestämdes tidigt vid första intervjun att kunderna skulle vara de som styr utvecklingen med sina krav och önskemål eftersom det i slutändan är de som ska använda systemen.

## **5.7 Intervju med kunder efter utvecklingsarbetet**

Syftet med intervjuer med kunderna var att ta reda på hur de upplevde sin roll att påverka utvecklingsarbetet med sina krav och önskemål. Även kunderna verkar ha en positiv syn på arbete och vi har fått en mängd positiv respons genom våra intervjuer. De allra flesta uppger att de har upplevt sin medverkan i systemutvecklingen som positiv och känner att de fått en riktig möjlighet att påverka.

*”Först tyckte jag det var svårt eftersom jag visste för lite om systemutveckling men sedan förstod jag att det var ju faktiskt mina åsikter som skulle komma fram och vad just jag tyckte var bra med ett sådant system.”*

*”Jag tycker att mina åsikter och frågor blivit bemötta på ett tillfredsställande sätt.”*

*”...det har varit intressant och spännande att se hur uppbyggnaden av ett system går till.”*

*”Enkäter är inte min starka sida men det har ändå varit rätt kul eftersom det har handlat om något konkret, det vill säga jag har själv kunnat sitta och leka lite med eran 'produkt' för att skapa en åsikt.”*

*”Ja jag är inte så van vid vare sig datorer eller systemutveckling så jag tycker att det varit både kul och intressant att se hur ett system växer fram.”*

Det är självklart svårt att tillgodose allas önskemål, speciellt i avseende på utseendet av systemen. Många hade synpunkter på utseendet på exempelvis knappar och många hade olika förslag som delvis sa emot varandra. Vi sådana tillfällen var det upp till oss att avgöra vad som skulle ändras och vilket förslag som skulle ignoreras. Vi utvecklade dock prototypen enbart efter kundernas och frisörens synpunkter och detta verkade ge ett stort engagemang bland användarna att de kunde se förändringen så tydligt efter sina önskemål.

*”Man känner att man kan vara med att påverka. Dessutom har mina krav tillgodosetts. Det känns bra att kunna påverka och att systemen blir enligt mig mer användarvänligt.”*

*”På många sätt har man tagit om hand om önskemål och det har i sin tur blivit att utformningen av både e-handel och bokningssystemet har blivit som man själv önskade.”*

*”Det känns skönt att kunna vara med och påverka utseende och funktioner i systemen. Detta gör att motivationen att delta ökar väsentligt vilket är viktigt för om systemen skall utvecklas till något bra där användarvänlighet och tydlighet ligger i fokus.”*

*”Det känns alltid bra när ens personliga åsikter tas om hand och du ser att systemen blir som du själv vill ha dem. Jag tror också att det är nödvändigt att man som användare verkligen känner att man är med och påverkar. Först då föds engagemang och motivation.”*

*”Delaktighet är alltid stimulerande! Tack vare att min respons tagits på allvar har det känts som ett lagarbete.”*

En annan viktig del i processen att involvera användarna är att de samtidigt lär sig systemen och känner sig positiva till att senare använda dem. Det blir inte senare ett nytt system som kunden ska lära sig utan är de själva med och styr utvecklingen kommer systemen att upplevas på ett positivare sätt.

*”Det har känts väldigt bra. Jag har fått en ökad insikt i hur systemen fungerar och att det är möjligt att påverka. Detta känns extra viktigt då datorer kan vara ’skrämmande’ och svåra att rå på i andra situationer.”*

*”Jag har lärt mig mer om datorer och hur systemen fungerar så när jag skall använda dem så är det inte några frågor eller problem.”*

*”Det känns kul att se att systemen blir bättre och bättre. Det stärker trovärdigheten för systemen och viljan att använda dem ökar betydligt.”*

*”Systemen blir som vi användare vill ha dem vilket är positivt. Detta gör att delaktighet kan ses som både engagemangökande faktor och också en*

*lärandefaktor eftersom vi lär oss systemen på ett bra sätt genom att testa dem på olika sätt.”*

Några negativa aspekter med detta arbetssätt är som vi skrev i förra avsnittet tidsåtgången. För kunderna har det inneburit två omgångar då de utvalda har varit tvungna att testa systemen och sedan svara på en enkät för att meddela sin åsikt. Detta har varit mer påfrestande för vissa och utan problem för andra.

*”Men det var bra att ni inte hade fler enkätutskick för det hade nog varit lite jobbigt att svara på dessa frågor många gånger. Jag vill vara med och påverka men efter ett tag har man inte fler åsikter.”*

*”..kanske skulle det räckt med endast en enkät. Systemen var ganska bra från början och säger man allt man tycker så kanske det hade räckt med en enkät.”*

*”Kanske skulle ha fått fler enkäter att svara på och kunna påverka mer och under längre tid.”*

*”Jag tror att man skulle behövt haft mera tid på sig för att kunna komma på bättre idéer.”*

*”Krävs att man har tid att svara på era frågor se det blir så konstruktivt som möjligt, men har man en 10 minuter ledigt så går det bra. Hade det varit många fler enkätutskick tror jag många svar skulle fallit bort på grund av att folk har annat för sig.”*

*”Har varit lite tidskrävande att behöva svar på enkäterna, men jag förstår att ni fått bättre resultat med flera enkäter.”*

*”Det har varit lätt att vara med eftersom jag enbart skulle svara på enkäter. Tror det är viktigt att det är lite man ska prestera om alla ska känna att de har tid och lust. Att svara på några frågor i en enkät orkar alla, så arbetsformen har varit mycket bra.”*

Kunskapen om datorer har också varierat mycket, vilket har lett till olika åsikter om det krävts för mycket av dem som användaren. Några tycker att det krävts för mycket av dem och vissa tycker det varit oproblematiskt.

*”I tid har det nog krävts cirka 30 minuter per enkät. Kvantiteter med kopiering och så vidare har gjort att jag fått skriva flera gånger vilket naturligtvis tagit tid. Kanske har ni gett väldigt basic instruktioner för hur ni vill ha svaren. Jag trodde t ex att man kunde spara ändringar i ert dokument och bara sända i retur vilket inte funkade på hotmailen.”*

*”...man blir ju alltid irriterad över att få ägna något en timma som man kunde gjort på en halv.”*

*” Alla kan gå in på Internet och svara på några frågor i ett worddokument.”*

*”...tycker att det var enkelt och lättfattligt. Det har varit mycket pedagogiskt upplagt och tydliga instruktioner. Det hela har varit väldigt användarvänligt.”*

Vi märkte också att det var vanligt att de kunder som var mer ovana vid arbete med datorer var nöjdare med systemen än de som hade mer erfarenhet. De ovana verkade glada över att ens klara av en bokning respektive beställning av en vara.

*”Som sagt är jag inte så van och tyckte att systemen var ok redan från början.”*

Vi frågade också kunderna vad de ansåg om traditionella systemutvecklingsmetoder där utvecklarna inte involverar användarna i lika hög grad utan en kravspecifikation skrivs och är utgångspunkten för utvecklandet. Ingen av våra tillfrågade kunder tyckte att den metoden skulle ha fungerat bättre, trots att några angav att det var tidspressande och krävande att medverka i vår process.

*”Ni har nog fördel av att kunder är med och säger sitt, då det faktiskt är de som ska använda systemen. Bättre med denna metod alltså. Verkar lite svårt att få med alla kraven i en ’specifikation’. Man kan ju ändra sig när man ser systemen och komma på fler grejer.”*

*”Det kan ju vara småsaker man vill ändra på också även om alla funktioner är med, småsaker som färg eller knappar.”*

*”Det kan kanske vara jobbigt för användarna om utvecklingen av systemen tar lång tid och då kanske den traditionella utvecklingen vore bra, men användarna förlorar ju hela möjligheten med att påverka och forma systemen som man vill. Jag tycker ni valt rätt systemutvecklingsmetod för det har varit roligt att få vara med och jag tror Klippotek Passagen tjänar på det också eftersom vi kunder får mer intresse av ett system som vi själva varit med och påverkat.”*

*”Den metoden fungerar ju utmärkt också, men den metoden som har använts här har ju klara fördelar för användarna vars förståelse och kunskaper oftast ligger på väldigt olika nivåer. Jag tycker den metoden som har använts var rätt val.”*

*”Användarna lär sig inte systemen på samma sätt vilket är en nackdel. Systemen blir som utvecklarna vill med funktionerna som bestämts. Detta kan göra att användarna inte vill använda systemen. Man vet ju inte hur systemen fungerar innan det används. I Prototyping får man ju pröva sig fram vilket jag tror är bättre.”*

De allra flesta har upplevt arbetet som mycket positivt och tyckt att det varit intressant, nyttigt och kul att medverka. Trots att några tyckte att det var lite krävande var det ingen som ville ändra arbetssättet väsentligt utan beskrev hela processen som intressant, engagerande och kul. Enda ändringar de gav förslag på var tillvägagångssättet på arbetsmetoden som fler enkätutskick, färre enkätutskick eller mer tid för deras del till förfogande. Alla var alltså överens om att användarna skulle vara med i stor utsträckning vid systemutveckling. Nästa alla angav att de skulle använda systemen vid behov istället för att ringa in sin bokning respektive

beställning. Det var endast 2 av 30 stycken som uppgav att de skulle fortsätta ringa in sin bokning eller beställning.

Vi tycker också att det har fungerat bra med att använda kunderna med i utvecklandet. Det har varit en stor trygghet i att ha dem med i och med att det är i slutändan de som ska vara nöjda med systemen och använda dem. Att då ta med kunderna i processen ger oss en bra guidning om vad de tycker är en bra boknings- och e-handelssida på Internet. Några negativa upplevelser har vi dock också fått. Det har varit svårt att engagera vissa användarna så de svarat i god tid. Många har skickat sina svar till oss flera veckor efter vi börjat med ändringar på prototypen och hela processen har dragit ut på tiden mer än vi planerat för. De allra flesta kunder var motiverade och engagerade, men de som inte motiverades av tillgodosedd kritik var svåra att få intresserade av vårt utvecklingsarbete. Vidare har det varit mycket svårt att ge instruktioner som passar alla oavsett datorvana så de har kunnat göra allt som var tänkt. Några har exempelvis haft svårigheter med filen vi skickade till dem med enkäten och inte vetat hur de sparar ner den och skickar tillbaka den till oss. Några har till och med skickat den via posten för att underlätta för sig, vilket tyvärr gett oss mer arbete istället.

## 5.8 Sammanfattning

För att ge en bättre överblick över de resultat som vi kommit fram till finns här resultaten i en tabell.

Intervju med frisören innan utvecklingsarbetet	<ul style="list-style-type: none"> <li>Nuvarande bokningsrutin är ineffektivt och till nackdel för både frisören och kunderna.</li> <li>Att ha ett inestående lager leder ofta till förlust för frisören.</li> </ul>
Observationer	<ul style="list-style-type: none"> <li>Kunder begär varor som frisören inte har i lager och inte vågar köpa in på grund av förlustrisken.</li> <li>Det bekräftade att bokningsrutinerna var ineffektiva.</li> </ul>
Prototyp	<ul style="list-style-type: none"> <li>Många skönhetsmissar på systemen fångades upp av metoden Prototyping, då det innebar flera tillfällen för användare att kritisera och göra oss uppmärksamma på fel och brister.</li> <li>Ju fler tillfällen som erbjuds och allteftersom kunderna testat systemen ju fler krav och önskemål har kunderna.</li> </ul>
Intervju med frisören efter utvecklingsarbetet	<ul style="list-style-type: none"> <li>Användare känner tillfredsställelse när kritiken åtgärdas</li> <li>Motiverande och stimulerande att påverka ett system som ger användaren fördelar.</li> <li>Tidsåtgången för Prototyping mycket osäker och processen kan bli krävande för användaren.</li> </ul>
Intervju med kunder efter utvecklingsarbetet	<ul style="list-style-type: none"> <li>Det är viktigt att användarna känner delaktighet.</li> <li>Prototyping skapar delaktighet för användarna och ger dem en möjlighet att påverka.</li> <li>Delaktigheten skapar intresse för systemen</li> <li>Att ha användare med ger många krav och önskemål för utvecklarna att styra utvecklingen efter.</li> <li>Många krav och önskemål kan vara svåra att tillgodose.</li> <li>Tillgodosedda krav och önskemål skapar ytterligare delaktighet, ökat intresse och tillfredsställelse och motivation att fortsätta delta i utvecklingsprocessen.</li> <li>Kunderna lär sig det framtida systemen genom delaktighet, vilket minskar tiden för inläring vid ett senare skede.</li> <li>Intresse för det framtida systemen ökat när användarna varit delaktiga i dess utveckling.</li> <li>Ju fler tillfällen användarna ges möjlighet att påverka och ju längre tid processen pågår ju mer krävande upplever användarna deras roll.</li> </ul>

	<ul style="list-style-type: none"><li>• Ju mindre datorkunskap en användare har ju mer krävande blir deras roll i utvecklingsprocessen.</li><li>• Ju mindre datorkunskap en användare har ju mindre dålig kritik ges från dem.</li><li>• Prototyping är att föredra då det är viktigt att de som ska använda systemen är med aktivt under hela utvecklingen.</li><li>• Användare som är med i utvecklingsprocessen ger för utvecklarna en bra riktlinje för utvecklandet, vilket skapar trygghet för utvecklarna att systemen utvecklas i rätt riktning.</li><li>• Svårt att uppskatta tidsåtgången</li><li>• Svårt att motivera användare som inte låter sig motiveras av tillgodosedda behov.</li></ul>
--	---



## 6 Diskussion

Vid utvecklandet av system för företag är det viktigt att en bra systemutvecklingsmetod används. Företag som anlitar systemutvecklare har rätt att kunna kräva att de slutgiltiga systemen ska stämma med de krav och önskemål de angivit. Att för en systemutvecklare lyckas fånga upp användarnas krav är dock inte alltid så lätt. I teoriavsnittet skriver vi om fyra svårigheter med andra systemutvecklingsmetoder. En av dessa svårigheter är just att fånga upp alla krav som användarna ställer på systemen. Enligt de traditionella utvecklingsmetoderna skrivs alltså oftast en kravspecifikation tillsammans med beställarna av systemen och detta dokument ska utvecklarna sedan utgå ifrån för att utveckla system efter beställarnas önskemål. Detta kräver också oftast en komplicerad och noggrann analys innan designen av systemen kan börja. Mattsson et al (1988) skriver att följderna blir som svårast när kravspecifikationen inte är rätt, eftersom resten av utvecklingsarbetet bygger på den. Ett misslyckat projekt är både tids- och ekonomiskt krävande. Det är viktigt att ett projekt fungerar och att utvecklarna använder en bra och pålitlig systemutvecklingsmetod som fångar upp alla krav och önskemål, eftersom beställarna litar på att få sitt system inom överenskommen tid och kostnad samt med de krav som sattes.

För att en kravspecifikation ska bli korrekt måste användarna tala om vad det är för system de vill ha och vad det ska utföra. Mattsson et al (1988) skriver ”Det kan vara så att användarna inte vet vad de vill ha, eller att de vet vad de vill ha, men kan inte uttrycka kraven på ett förståeligt sätt”. De traditionella systemutvecklingsmetoderna har som sagt en noggrann analysfas där utvecklarna skapar modeller av världen runtomkring systemet och av systemet själv för att sedan kunna använda dessa som modeller för hur systemet ska byggas. Det finns däremot ingen garanti för att de förstått användarna rätt i sin förklaring vad de vill ha för system och detta upptäcks då efter systemet är klart och sätts i bruk eftersom användarna inte är delaktiga i utvecklingsprocessen. Problem kan också uppstå när användarna ändrar sina krav eller kommer på nya önskemål efter tidens gång.

Prototyping löser som också skrivs i den teoretiska referensramen många av de traditionella systemutvecklingsmetodernas problem genom att bygga en snabb och enkel prototyp av systemet så som utvecklarna förstår användarnas krav. Prototyping använder sig av ett systematiskt tillvägagångssätt som hjälper att lösa de problem som finns vid specificerad systemutveckling (Boar 1984, Ehn 1988, Lanz 1988, Vonk 1990). Mattsson et al (1988) skriver att prototypsystemet ska fungera som en brygga mellan användare och utvecklare och ska göra det lättare för de båda parterna att förstå varandra. Genom prototypen får både användare och utvecklare en försäkran om att de förstått varandra rätt. Vid missförstånd är det nu lättare att förklara vad som missuppfattats, genom att användarna direkt kan peka på något konkret de vill ändra. Har utvecklarna olika idéer är det inte heller alltför kostbart att göra en prototyp av de olika idéerna och sedan låta användarna bestämma vilken som passar deras krav och önskemål bäst.

### 6.1 Intervjun med frisören och observation innan utvecklingen

Intervjun med frisören innan utvecklingen började samt observationerna visade att deras nuvarande arbetssätt var ineffektivt. Förväntningarna på systemen var att de skulle minska de störande momenten samt att de slipper ha ett lager inestående som kostar pengar. Många kunder uppgav vid intervjun att de har blivit positivt inställda till systemen eftersom de

medverkat i hög grad vid utvecklandet. De uppger också ha lärt sig systemen bra genom testkörning vid olika tillfällen. Endast 2 av 30 trodde att de skulle ringa in sin bokning och beställningen i framtiden och detta tyder på att Klippotek Passagens problem kommer att minska inom den närmaste tiden. Efter att ha fått många av sina önskemål uppfyllda har en positiv inställning framkommit hos kunderna och vi tror med stor sannolikhet att kunderna kommer att använda systemen då de ger en stor fördel för dem genom att systemen är tillgängliga dygnet runt.

## **6.2 Modellering**

I traditionella systemutvecklingsmetoder deltar användarna endast inledningsvis i processen, men med Prototypingmetoden måste användarna aktivt ge synpunkter på det framväxande systemet. Men hjälp av användarnas aktiva respons kan dokumentationsdelen härmed kännas mindre viktig i Prototypingmetoden jämfört med traditionella utvecklingsmetoder som inte har användarnas aktiva respons som riktlinje. Dokumentationsdelen kan jämföras med analysfasen hos traditionella systemutvecklingsmetoder. Respons från användarna fungerar som en ständig riktlinje på om projektet är på rätt spår och modellering av klasser och dylikt upplever vi inte lika behövande i ett Prototypingprojekt. Vårt beslut om att ersätta Prototypingmetodens ursprungliga dokumentationsfas med delar från Objektorienterad analys och designmetoden kan kännas väl ambitiöst och inte nödvändigt. Vi hade efter första intervjun med frisören en förklaring på vad de tänkte sig för system och sedan hjälpte kunderna oss under övervakning av frisören att peka på vad som bör förändras så sidan blir så enkel och förståelig som möjligt. Vidare var systemen vi skulle utveckla av den mindre karaktären vilket gjorde att vi redan från början hade en preliminär lösning uttänkt. Ingen större analys för att hitta systemens svårigheter kändes nu i efterhand nödvändigt. Exempelvis händelsetabellen gav oss egentligen ingenting eftersom systemen var små. Syftet med en händelsetabell är som skrivs i teoriavsnittet att reda ut vilka händelser som påverkar vilka klasser, men i vårt fall berör alla händelser alla klasser.

## **6.3 Prototyp**

Metoden Prototyping innebar att många slarvfel fångades upp genom de olika tillfällena kunderna fick säga sitt. Trots att vi ansåg oss vara noggranna tycktes slarvfel uppkomma lätt. Eftersom alla önskemål härmed inte tillgodosågs, då kunderna vid första enkätomgången kommentera felen i hög grad, fångades dessa fel upp av metoden Prototyping och därmed rättades till. Dock verkade det som att många kunder hade mer att klaga på ju fler enkätutskick som gjordes. Dessa kunder verkar ha blivit bekant med systemen vid första enkätomgången och sedan har fler krav och önskemål vuxit fram. Tack vare Prototyping är detta en positiv egenskap att användarna kommer på fler saker innan utvecklingsarbetet är klart. Vid traditionell systemutveckling kan detta bli en nackdel då användarna oftast inte får se systemet förrän det är klart och att då komma med fler krav och önskemål kan vara till olägenhet för utvecklarna. Vidare märkte vi mycket av Mattssons et al (1988) teori om att användarna inte riktigt vet vad det är de vill ha. De kanske vet vad de vill ha men inte exakt hur det ska se ut. Vid enkäterna upplevde vi att de efterlyste funktioner men sen när funktionen kom så var det tillvägagångssättet de ville ändra. Funktionen utförde rätt saker men det var ändå ett annat tillvägagångssätt de efterlyste. Exempelvis så vill de kunna se vilka tider som var bokade redan och trots att det var möjligt genom en länk att komma till en sida som radade upp alla bokade tider var det ändå inte exakt så det skulle gå till enligt många.

Några ville att ett nytt fönster kom fram eller att de bokade tiderna fanns på samma sida som där kunden fyller i uppgifter om önskad tid för behandling. När utvecklarna ger användaren något de tror användarna vill ha visar det sig att det inte var exakt det utan det krävs fler förändringar. Med en prototyp kan kommunikation mellan utvecklaren och användaren betydligt förbättras då användaren med prototypen kan visa vad som ska ändras eller läggas till istället för att kunna formulera sig så utvecklaren förstår. Härigenom får inte bara användaren funktionen den vill ha utan också tillvägagångssättet för funktionen.

Vi märkte också att kunskapsnivån bland kunderna varierade mycket. Enkäten skickades ut via mail och vissa kunder hade exempelvis problem med att skicka tillbaka enkäten. Denna kunskapsvariation märktes även i svaret på enkäterna. De som hade mycket kunskap var mer missnöjda och hade fler krav och önskemål än dem med mindre kunskap om datorer. Många med liten kunskap verkade glada över att ens klara av en bokning eller att beställa en vara samtidigt som de med datorvana kritiserade tillvägagångssättet i högre grad. De med mer kunskap kunde också ge oss konkreta förslag på förändringar i hög grad samtidigt om de utan större data vana mest visste att något inte var bra men hade inga förslag på hur det skulle förändras. För resultatet för metoden Prototyping tror vi inte att kunskapsnivån hos användarna spelar en så stor roll om användarurvalet är blandat då det räcker med att för utvecklarna få reda på att något är mindre bra. Utvecklarna kan då bara göra förändringar tills användarna är nöjda och det är inte nödvändigt för användarna att komma på egna lösningar. Deras roll är att påverka utvecklingsarbetet av sitt framtida system och komma med krav och önskemål. En utvecklare kan aldrig kräva av en användare att ha systemutvecklingskunskaper och kunna förklara hur något ska göras.

## **6.4 Intervju med frisören och användarna**

Intervjun med frisören efter utvecklandet visade att användare känner tillfredställelse när kritiken tas omhand och åtgärdas. Frisören tyckte också att det var lättare att påverka ett system som ger honom fördelar. För kunderna gav detta ungefär samma positiva känsla. Både frisören och kunderna var överens om att detta var arbetssättet att föredra och att det är mycket viktigt att användaren är delaktig i hög grad. Prototyping gör att användarna kan vara med och få denna positiva känsla inför utvecklingen av deras system. Vid traditionell systemutveckling kan aldrig användarna få denna chans i så stor utsträckning och kanske aldrig blir lika positivt inställda till systemen som utvecklats. Prototyping ger alltså inte bara fördelar i att användarna verkar som en riktlinje för systemutvecklarna utan det handlar också om deras inställning till det framtida systemet. Ju mer användarna får systemet som de vill ju bättre inställda blir de gentemot att sedan använda systemet. I detta fall var det viktigt för Klippotek Passagen att kunna arbeta mer ostört än tidigare och därmed är det viktigt att kunderna har en bra inställning gentemot systemen. Genom att de känt sig delaktiga och sett sina synpunkter tagits omhand har motivationen och intresset ökat för systemet.

Metoden Prototyping kräver dock lite mer av användarna som här aktivt ska delta i hela utvecklingsprocessen. Det krävs en positiv inställning och tålamod av användarna samt krävs det att de vågar kritisera systemet. I vår fallstudie upptäckte vi svårigheten med att få alla användarna att inom en rimlig tidsgräns svara på våra enkätfrågor som gav dem möjlighet att påverka utvecklingsarbetet. Det var främst enkätutskick ett som tog lång tid för vissa av kunderna att svara på och skicka tillbaka. Detta förbättrades inför enkätutskick två och verkar bero på att kunderna kände motivation till att många av deras krav hade tillgodosetts. Det verkar skapa intresse, motivation och tillfredställelse att se sina krav och önskemål tas tillvara.

Det fanns dock en grupp som fortsatte att vara svårmotiverade, men de var få till antalet. Inför första enkätutskicket fanns mycket som skapade missnöjde och motivationen verkade inte finnas där innan deras krav tagits omhand. Några kunder angav vid intervjun att det kändes krävande att medverka då de hade mycket på förhand att stå i som exempelvis jobb. Detta kan för utvecklaren bli ett dilemma då det är bra för utvecklaren att få in så mycket respons och kommentarer som möjligt för att kunna skapa ett så bra och användarvänligt system som möjligt. Det gäller att i processen hitta en lämplig nivå av antal omgångar som användare ska delta i.

## **6.5 Prototyping inom småföretag**

I vår fallstudie var frisörer och kunder användarna av systemet. Frisörerna var sammanlagt tre och de hade en relativt begränsad kundkrets. En utvecklingsmetod som Prototyping fungerar därför i dessa sammanhang väl eftersom mycket stor del av användarna här kan vara med och påverka. Vi utredde i ett tidigt skede att det var kundernas och framförallt stamkundernas önskemål som var viktiga eftersom det är de i slutändan som ska vilja använda sig av systemet för att ge sig själv och frisörerna fördelar. Eftersom Klippotek Passagen alltså har en relativt begränsad kundkrets kunde många av stamkunderna vara delaktiga i utvecklingen och därmed få sina önskemål till stor grad uppfyllda. I ett större företag kan man tänka sig att ett urval får göras som verkar som en representation av ett stort antal användare för att få fram krav och synpunkter. I sådana fall är det omöjligt att låta alla påverka men i vår studie kunde alla frisörerna vara med och påverka samt ett stort antal av stamkunderna. Det tordes vara sant att ju fler som inte får vara med och påverka ju större är risken att fler inte får sina önskemål de hade haft om de hade medverkat tillgodosedda och därmed inte får samma positiva känsla inför användning av systemet. Tiden för inläring minskar också för de medverkande. Om det då i ett stort företag finns många som inte medverkar kommer det ändå att ta en del tid för denna inläring och därmed upplever de inte samma fördel med Prototyping.

En Prototypingmetods resultat påverkas i hög grad av förutsättningarna vid utvecklandet. I ett projekt där man har en begränsad tid och budget kan det vara svårt med Prototypingmetoden att förutspå hur lång tid projektet tar och vilken kostnad projektet kommer att ha. Utvecklingen av systemet är trots allt klart när användarna är nöjda. De får chansen att kommentera prototypen i flera omgångar och antalet omgångar beror helt på när deras krav och önskemål är uppfyllda med reservation för förändrade krav, önskemål och förutsättningar. Vid mindre system kan inte tidsåtgången förändras så väldigt mycket. Även om en mängd oförutsägbara händelser inträffar så fördröjs projektet inte så mycket som ett större system kan göra. Ett projekt som vårt bokningssystem och e-handelssystem hade begränsat antal funktioner och kan inte ta hur lång tid som helst samtidigt som ett system som ska omfattas av fler användningsområden och har längre projekttid löper större risk att drabbas av fler oförutsägbara händelser.

Beck och Fowler (2001) skriver ”Building a product the right way still *sounds* like a laudable goal, but - let’s face it - what really matters today is building it *fast*”.

Vid användning av Prototyping är det som sagt svårt att avgöra hur länge processen kommer att pågå. Detta är viktigt att veta dels för beställaren och dels för användarna. Beställaren vill veta när de kan få systemet i bruk och hur mycket det kommer att kosta, vilket allt beror på under hur lång tid projektet kommer att pågå. Om användarna ska vara delaktiga i processen gäller det att det inte tar för lång tid eftersom ju längre tid projektet pågår ju mer kommer att

krävas av användarna. Det beror också på vilka användarna är i processen. Är det ett företags anställda som ska ha systemet kan man tänka sig att de anställda kan vara delaktiga i processen på arbetstid som en del i deras arbetsuppgifter för en period. Härigenom får de kompensation i form av lön vid medverkan, vilket bör göra att de känner mer motivation. Är användarna däremot som i vårt fall en grupp personer som tagits med som testpersoner i ett urval finns ingen kompensation mer än att de eventuellt kommer att ha nytta av systemet genom användning. I detta fall är det speciellt viktigt att projektet inte drar ut på tiden så användarna känner att det krävs för mycket av dem. Då skulle man kunna tänka sig att inställningen blir mindre positiv till systemet, vilket kan ha negativa konsekvenser när det gäller graden av användning. Det är med andra ord viktigt att motivera användarna att hjälpa utvecklarna, speciellt om utvecklingen drar ut på tiden. Det beror till stor del på hur utvecklarna formar arbetsuppgiften för användarna. Det finns tre kritiska faktorer vid utformning av arbetsuppgifter: (1) Individerna behöver uppfatta sitt arbete som meningsfullt och värdefullt. Det gäller att arbetet leder fram till en viktig och synlig helhet. (2) Det är också viktigt att de genom sitt arbete får använda sitt omdöme och sina erfarenheter så de kan känna ansvar för sina resultat. (3) Dessutom behöver individerna få återkoppling på sina prestationer så att de utvecklas och blir bättre och bättre (Hackman och Oldham 1980, Hackman et al 1987, Bolman och Deal 1997). Utvecklaren måste lägga stor vikt vid att få användaren att förstå hur viktig deras insats är och känner ansvar för sina resultat samt att de får återkoppling på sin insats. Om testningen och utvärdering av en prototyp är i en användares arbetsuppgifter på arbetet där användaren erhåller lön som kompensation kanske dessa tre punkter inte är lika intressanta för utvecklaren, men om användaren som i vårt fall inte får någon kompensation blir detta en mycket viktig del.

Med tanke på detta kan tänkas att Prototyping passar bäst vid mindre företag och mindre system. Det är lättare att göra ett bra urval då de allra flesta användarna kan vara med och påverka. Det är också lättare att ta tillvara fler önskemål eftersom det är ett begränsat antal användare. Det är följaktligen lättare att skapa motivation eftersom tillvaratagna önskemål skapar motivation och det är ett mer begränsat antal önskemål jämfört med större antal användarmedverkande. Vidare löper större system större risk att dra ut på tiden och drabbas av oförutsägbara händelser.

## **6.6 Prototyping som systemutvecklingsmetod**

Att ha användarna med som ger synpunkter på systemet under utvecklingens gång ger alltså många fördelar. Förutsättningar och krav kan ändras under utvecklingsarbetets gång. Många traditionella systemutvecklingsmetoder missar den flexibilitet som Prototyping ger genom att användarna under hela projektet måste ge respons på utvecklingsarbetet och därmed komma med fler krav. Mattsson et al (1988) skriver ”Eftersom användarna besitter stor kunskap inom sitt arbetsområde och därmed om det framtida informationssystemets funktioner, anser vi att det är självklart att användarna måste få framföra sina åsikter i utvecklingsarbetet”. Användarnas medverkan ger andra fördelar än att de fungerar som en riktlinje för utvecklarna. Användarna får genom aktiv medverkan en stor kännedom om deras framtida system. Detta gör att deras inlärningstid kommer att minska när väl systemet tas i bruk. Eftersom de själva är med och påverkar utvecklandet kan det tänkas ge tillit till systemet samt en känsla för användarna av delaktighet. Det slutgiltiga systemet kommer med användarnas aktiva medverkande att förhoppningsvis stämma överens med de krav och önskemål som sattes och detta skapar tillfredsställelse hos beställaren samt hos användarna. Som vårt resultat visar är det så att användarna vill vara med men det får inte ta för lång tid eller vara för stor

ansträngning som krävs. Att ha användarna med i utvecklingen ger utvecklare också tillåtelse att göra fel.

Vid Prototyping behövs ingen större kravspecifikation vilket underlättar betydligt. Vid intervjun med frisören fick vi endast några meningar som vi översatte till krav. Med dessa krav skapade vi en prototyp och visade den sen som ett förslag som de sedan kunde redigera om och om igen. Istället för kravspecifikationen har man användarna som fungerar som pekpinne. Det skapas genom denna kommunikation en social process som inte finns bland de traditionella systemutvecklingsmetoderna. Om inte användarna hade varit med och vi haft denna interaktion med dem hade vi aldrig fått veta vad användarna egentligen ville ha. Systemutvecklare vill att användarna ska vara nöjda och genom att använda Prototyping finns den sociala processen där för att kommunicera aktivt mellan användare och utvecklare och därmed är det större chans att tillgodose användarnas behov på ett bättre sätt. Nu är de själva med under hela utvecklingsprocessen och säger hur de vill ha det. Denna fördel och den sociala interaktionen mellan utvecklare och användare går ofta förlorad vid de traditionella systemutvecklingsmetoderna. Prototyping kan härmed sägas vara en kvalitetsmetod vid systemutveckling, då processen är klar när användarna är nöjda. Tidsaspekten som annars kan ses som ett problem blir härmed en fördel för användarna då de kan vara säkra på att få ett system de förstår och som krävs för just deras verksamhet.

## **6.7 Prototyping i ett vidare perspektiv**

Som vi diskuterat tidigare finns det en mängd risker med att använda sig av Prototypingmetoden. Enligt Baskerville och Stage (1996) beror det på vilket system som skall utvecklas om Prototyping passar eller ej. Om det är ett system som har hög teknisk komplexitet behövs tydligare specificering av systemet innan det börjar byggas vilket kan medföra att Prototypingmetoden inte är det naturliga valet. System som Prototypingmetoden passar för kan till exempel vara kontrollsystem av olika slag (Duke et al 1989) och den har också använts med framgång i objektorienterade system. Enligt Coad och Yourdon (1991) kan alla system som bygger på en objektorienterad design använda Prototyping. Vid användningen av Prototyping metoden minskas specificeringen av systemet vilket kan leda till ett mer effektivt sätt att göra bra system. Prototyping kritiserar momentet att specificera för mycket när det gäller objektorienterade system (Gupta et al 1989). Vi förstod på ett tidigt stadium att vi inte behövde göra en alltför ingående specificering eftersom användarna skulle vara med och påverka utvecklingsarbetet av både systemets utseende och dess funktioner. Enligt Martin (1990) är Prototyping speciellt användbar vid utveckling av interaktiva system.

Förr har bland annat nackdelarna med metoden Prototyping varit enligt Lantz (1988) att databaserade prototyper kan bli ineffektiva. Dessutom kan det bli svårt att hantera om prototypen blir för komplex (Alavi 1984) men genom att datorer har fått allt mer kapacitet har problem som dessa lättare kunnat avhjälpas. Ett av problemen som fortfarande kvarstår är svårigheten för utvecklarna att styra Prototypingprocessen (Alavi 1984, Boar 1984, Connell och Schafer 1989). Det problem som var mest överhängande för oss när det gällde styrning av själva processen var att svarstiderna inte hölls för enkätsvaren. Vi kunde ju inte börja arbeta förrän vi fått in alla svarsenkäter och sammanställt dem och bestämt vad som skulle åtgärdas. Användarna är med andra ord den avgörande faktorn för att projektet ska lyckas. Utvecklarna har svårt att planera och kontrollera processen. Planerna kan lätt ändras för varje varv systemet går igenom Prototypingprocessen. Kontrollfaktorn är mycket svår att hantera

eftersom det är användarna som styr projektet framåt genom sitt testande. (Baskerville och Stage 1996)

Något som Baskerville och Stage (1996) rekommenderar för att få en mer kontroll över projektet är att efter den första prototypen utvärderar riskerna som kan uppstå i projektet. Utvecklarna försöker att definiera risker, specificera konsekvenser och väljer med vilken strategi risken skall lösas. Detta görs för att förebygga de problem som kan uppstå. Med riskanalysen som underlag går det sedan också att förstå vad som gått fel och varför. Utan denna dokumentation är det lätt att skylla på andra faktorer än de verkliga. Eftersom vi var relativt oerfarna med att använda Prototypingmetoden så var vår riskanalys mycket begränsad. Enda risken vi hade förutspått var att användarna inte skulle svara på våra enkäter som skickades ut eller åtminstone inte hålla svarstiderna. Strategin vi hade satt upp var att från början försöka förklara för användarna hur viktig deras insats var för systemets utveckling. Trots detta hade vi problem att få in en del enkäter i rätt tid. Vår andra strategi var helt enkelt att då ringa eller skicka e-mail till dessa personer och ytterligare förklara vikten av hur viktiga deras svar var. Vi borde dock lagt ner mer tid på att utvärdera tänkta risker och hur dessa skulle kunna undvikas och lösas på bästa sätt. Vid ett större projekt där det satsats stora ekonomiska resurser ser vi det här som en nödvändighet att göra en riskanalys. Beställarna av systemet måste få veta vilken risken är med projektet och vad som kan gå fel samt hur detta i så fall kommer att hanteras.

Rapid Prototyping är idag ett vanligt redskap inom en mängd olika områden. Det är både ingenjörer och designers som använder sig av denna teknik (Tripp och Bichelmeyer 1990). En viktig aspekt som vi upptäckte är att man måste väva in en mängd eftertanke i själva utvecklandet. Det går inte bara att köra på i full fart framåt utan man måste även ibland luta sig tillbaka och tänka efter om lösningen är smartare än föregående lösning. Tanken är att systemet skall bli bättre och bättre och det är utvecklarnas uppgift att se till att kvalitén kvarstår. Givetvis får man en del förslag som istället för att utveckla systemet gör det sämre. Då är det nödvändigt att gå in som systemutvecklare och försöka få systemet på rätt spår igen.

## 7 Slutsats

Vår frågeställning är: *Hur fungerar systemutvecklingsmetoden Prototyping vid systemutveckling av mindre system för småföretag?*

De positiva effekterna med att använda metoden Prototyping är många. För oss systemutvecklare är det en trygghet att ha med användarna i processen. De verkar som en riktlinje för oss och processen är klar när användarna är nöjda. Användarna själva har i vår undersökning också upplevt många fördelar med att vara delaktiga. Största fördelen för dem är att de lär sig systemet de senare skall använda. Systemet utvecklas helt efter deras krav och önskemål och följaktligen får det ett system de själva vill ha. Detta ger en större garanti för att systemet verkligen kommer att användas. Prototyping i sig fångar upp systemutvecklarnas misstag eftersom processen är iterativ och de olika faserna genomgår flera gånger.

Användarna är den avgörande faktorn för om systemutvecklingen kommer att bli lyckad eftersom de med sina krav och önskemål utvecklar systemet. På grund av detta är det av yttersta vikt att användarna känner sig motiverade och engagerade. Ett stort ansvar hos systemutvecklaren är att hålla motivation och engagemang uppe hos användarna. Vi upptäckte att en stor motivationsfaktor var att deras krav togs om hand och att systemet ändrades efter deras önskemål. Härigenom kände de delaktighet och ökat intresse av systemet. Det ökade intresset medförde att användarna härigenom kom upp med fler krav på systemet.

Vi märkte också en stor skillnad bland användarna beroende på vilken datakunskap de hade innan projektet startade. De med mer vana hade betydligt fler åsikter och var mer kritiska än de med mindre vana. Samtidigt så var det mer krävande för de mindre datavana eftersom en del saknade grundläggande datakunskaper som filhantering. Metoden är givetvis krävande för alla berörda användare eftersom de har en sådan stor roll i själva systemutvecklingen. Ju längre tid utvecklingen pågick märkte vi att några användare var svårare att motivera att fortsätta att testa systemen och komma med ytterligare åsikter. Som systemutvecklare är det omöjligt att tillgodose alla krav vilket gör att motivationen sjunker för vissa användare som inte får sina krav tillgodosedda. En annan negativ aspekt är att tidsåtgången är ytterst osäker med Prototyping. Det beror helt på när användarna är någorlunda nöjda.

Vårt systemutvecklingsprojekt med metoden Prototyping fungerade på ett tillfredställande sätt. Vi lyckades på en relativt kort tidsperiod skapa två system för Klippotek Passagen som de flesta var nöjda med. Vi hade dock inte räknat med att motivationen av användarna skulle ha en så stor inverkan på utvecklingen. En riskanalys enligt Baskerville och Stage (1996) skulle ha varit användbart för att kunna hantera riskerna på ett bra sätt. Eftersom processen med Prototyping förlitar sig så mycket på användarna är det svårt att dra generella slutsatser att Prototyping skulle fungera i alla små system. Vi anser att Prototyping är en lämplig metod för utveckling av små system i små företag eftersom det är möjligt att ha med användarna i så stor utsträckning så de får ett system som de verkligen vill ha. Processen tycks lämpa sig bättre för mindre system eftersom tidsåtgången annars blir näst intill omöjlig att avgöra.

Med utgångspunkt från vår forskningsfråga och vårt praktiskt genomförda fall har argument tagits fram som bevisar att Prototyping kan fungera väl vid systemutveckling av mindre system i småföretag. Det är emellertid viktigt att komma ihåg att Prototyping innebär osäkerhet och risker. Vi rekommenderar att börja utvecklingsprocessen med en riskanalys enligt Baskerville och Stage (1996) för att verkligen analysera de risker och problem som kan



uppstå. På detta sätt kommer systemutvecklaren att få en större kontroll på utvecklingen med systemutvecklingsmetoden Prototyping.

## 8 Referenser

### 8.1 Litteratur

- Ahlandsberg R, Sandström J, Systemutveckling – utan skygglappar, Lunds Universitet, Institutionen för Informationsbehandling – ADB, 1988
- Backman J, Rapporter och uppsatser, Studentlitteratur, Lund, 1998
- Beck K, Fowler M, Planning Extreme Programming, Addison-Wesley, Boston, 2001
- Boar B H, Application Prototyping: A Requirements Definition Strategy for the 80s, John Wiley and Sons, New York, 1984
- Bolman L G, Deal T E, Nya perspektiv på organisation och ledarskap, Jossey-Bass Publishers, San Francisco, 1997
- Brown D, An Introduction to Object-Oriented Analysis objects in plain English, John Wiley and Sons, 1997
- Burman S, Bäckman B, Systemering, Almqvist & Wiksell Dataförlagen, 1992
- Coad P, Yourdon E, Object-Oriented Design, Prentice-Hall, Englewood Cliffs, NJ, 1991
- Connell, J L, Schafer L B, Structured Rapid Prototyping, Yourdon Press, Englewood Cliffs, NJ, 1989
- Dahmström K, Från datainsamling till rapport – att göra en statistisk undersökning, Studentlitteratur, Lund, 2000
- Easterby-Smith M, Thorpe R, Lowe A, Management Research, SAGE publications, 1991
- Ehn P, Work-Oriented Design of Computer Artifacts, Arbetslivscentrum, Stockholm, 1988
- Hackman J R, Oldham G R, Work Redesign, Addison-Wesley, 1980
- Hackman J R, Oldham G R, Janson R, Purdy K, The Great Writings in Management and Organizational Behaviour, Random House, New York, 1987.
- Lantz K E, The Prototyping Methodology, Prentice-Hall, Englewood Cliffs, NJ, 1988
- Martin J, Information Engineering, Book II, Prentice-Hall, Englewood Cliffs, NJ, 1990
- Mathiassen L, Munk-Madsen A, Nielsen PA, Stage J, Objektorienterad analys och design, Marco Publishing ApS, 2001

Mattsson C, Sandberg A, Bjarke U, Florén J, Karlsson L, Eriksson S, Ringblom E, Wahlin C, Wiberg M, Nordström P, Pettersson K-U, Prototyping – för högre grad av användarstyrning vid systemutveckling, Lunds Universitet, 1988

Nielsen J, Användbar webbdesign, Liber AB, Stockholm, 2001

Oskarsson Östen, Programutveckling i liten skala – en praktisk handbok, Studentlitteratur Lund, 1994.

Rieber L P, Computers, graphics and learning. Madison, WI:Brown & Benchmark, 1994

Ronne E, ASP Active Server Pages, Docendo Läromedel AB, Stockholm, 2000

Tripp, S, Bichelmeyer B, Rapid Prototyping: An alternative instructional design strategy, Educational Technology Research & Development, 1990

Vonk R, Prototyping. The Effective Use of CASE Technology, Prentice-Hall, New York 1990

Welling L, Thomson L, PHP and MySQL Web Development, Sams Publishing, USA, 2001

## **8.2 Internet**

[www.php.net](http://www.php.net)

[www.mysql.com](http://www.mysql.com)

## **8.3 Artiklar**

Alavi M, An Assessment of the Prototyping Approach to Information Systems Development, Communications of the ACM (27:6), 1984

Baskerville R L, Stage J, Controlling Prototype Development Through Risk Analysis, MISQ, 1996

Duke E, Brumbaugh R, Disbrow J, A Rapid Prototyping Facility for Flight Research in Advanced Systems Concepts, Computer (22:5), 1989

Gupta R, Cheng W, Hardonag I, Breuer M, An Object-Oriented VLSI CAD Framework: A Case Study in Rapid Prototyping, Computer (22:5), 1989

Ljungberg J, Open Source Movements as a Model for Organising, European journal of information systems (9:4), 2000

Yin R, Analyzing Case Study Evidence. Kapitel 5 ur: Case Study Research. Design and Methods, Sage Publications, 1984

## 9 Bilagor

### 9.1 Bilaga 1 – Intervju med frisören innan utvecklingsarbetet

#### Intervju med frisör innan utvecklingsarbetet påbörjas

Syftet med denna intervju är att ta reda på vad Klippotek Passagens frisörer förväntar sig av utvecklingsarbetet av bokningssystemet och e-handelssystemet samt att klargöra vilka omständigheter som råder vid friörsalongen.

1. Vad har ni för krav och önskemål på bokningssystemet?
2. Vad har ni för krav och önskemål på e-handelssystemet?
3. Hur går bokningar av kunder till på er salong idag?
4. Har ni avsatt tid då ni tar emot samtal från kunder som ska boka?
5. Hur bokförs bokningarna?
6. Hur går försäljningen av produkter till som det är idag?  
-Beställer kunderna i förväg?
7. Har ni något inestående lager?
8. Hur mycket kan ni i så fall behöva ligga ute med i lagerkostnad?
9. Händer det att ni går förlust på försäljningen genom exempelvis produkter som blir för gamla och måste slängas?
10. Hur stor är er kundkrets i ungefärligt antal?
11. Har några av era konkurrenter etablerat sig på Internet?

12. Vilka fördelar förväntar ni er genom ett bokningssystem och ett elektroniskt handelssystem på Internet?
13. Är det något du vill tillägga angående detta?

## 9.2 Bilaga 2 – Enkätundersökning

### Undersökning av Klippotek Passagens system på Internet

Denna enkät ingår i 20 poängskursen ”Magisteruppsats” på Institutionen för Informatik vid Göteborgs universitet. Enkätens syfte är att med hjälp av vanliga användare utvärdera och utveckla system efter deras krav och önskemål. Denna systemutvecklingsmetod kallas Rapid Prototyping och utgår från ett nära samarbete med användarna av systemet. Klippotek Passagen är en frisersalong på Drottninggatan 26 i Göteborg och vill genom att etablera sig på Internet utöka sin service till sina kunder. Systemen vi vill ha synpunkter på är en elektronisk handelssida och ett elektroniskt bokningssystem. Denna enkät förutsätter inte någon större kunskap inom IT i allmänhet. Användarna skall endast använda systemen för att kunna med sina krav och önskemål hjälpa oss att utveckla systemen. Om du känner att det är någon fråga du inte kan besvara så hoppa över den och gå vidare till nästa. Alla som svarar kommer att förbli anonyma. Vi är tacksamma om svaren ni lämnar blir så utförliga som möjligt och att ni skickar med denna fil med svaren som en bilaga i mailet tillbaka till oss.

Markera det alternativ som du anser vara det som passar dig bäst.

Kön:            Kvinna ( )                            Man ( )

Ålder:        18 – 25 ( )    26 – 35 ( )    36 – 45 ( )    46+ ( )

#### Frågor om Klippotek Passagens E-handelssystem

Fråga:	Svar:
1. Vad anser du om helhetsintrycket? Vad var bra? Vad var mindre bra?	
2. Var sidan enkel att använda? Vad var enkelt? Vad var mindre enkelt?	
3. Var allt förståligt? Vad var förståligt? Vad var mindre förståligt?	
4. Vilka förändringar tycker du skulle behöva göras för att sidan skall bli så enkel och förstålig som möjligt?	
5. Hur gick det att orientera sig bland varorna? Vad gick bra? Vad gick mindre bra?	
6. Hur gick det med att lägga till varor i kundkorgen? Vad var bra? Vad var mindre bra?	
7. Hur gick det när ni handlat klart och skulle skicka beställningen? Vad var bra? Vad var mindre bra?	

## Frågor om Klippotek Passagens Bokningssystem

Fråga:	Svar:
1. Vad anser du om helhetsintrycket? Vad var bra? Vad var mindre bra?	
2. Var sidan enkel att använda? Vad var enkelt? Vad var mindre enkelt?	
3. Var allt förståligt? Vad var förståligt? Vad var mindre förståligt?	
4. Vilka förändringar tycker du skulle behöva göras för att sidan skall bli så enkel och förstålig som möjligt?	
5. Hur gick det att boka en tid? Vad var bra? Vad var mindre bra?	
6. Hur gick det med att se sin bokade tid? Vad var bra? Var vad mindre bra?	
7. Hur gick det att avboka? Vad var bra? Vad var mindre bra?	

### **9.3 Bilaga 3 – Intervju med frisören efter utvecklingsarbetet**

#### **Intervju med frisören efter utvecklingsarbetet är klart**

Syftet med denna intervju är att ta reda på hur Klippotek Passagens frisör upplevde utvecklingsarbetet av bokningssystemet och e-handelssystemet. Utvecklingen av systemen gjordes med systemutvecklingsmetoden Prototyping och innebar en tät kontakt med användarna av de framtida systemen. Efter vi utvecklade en första prototyp av systemen baserade på frisörernas krav och önskemål var det sedan kundernas krav och önskemål som styrde utvecklingen av systemen.

Vi vill nu ställa några frågor om hur ni som frisör har upplevt er medverkan och vad ni tycker om detta samarbete mellan utvecklare och användare vid Klippotek Passagens Internetetablering. Vi får be er att svara så utförligt på frågorna som möjligt och motivera era svar, då vi genom era svar kommer att utvärdera hela systemutvecklingsarbetet.

1. Har du känt dig delaktig i utvecklingsarbetet med bokningssystemet och e-handelssystemet?
2. På vilka sätt har du känt dig delaktig?
3. Tycker du det har varit kul att medverka i utvecklingsarbetet?
4. I vilka avseenden tycker du dina krav och önskemål har tagits omhand?
5. Har du krav och önskemål på de båda systemen som inte tagits omhand?
6. Vad har fungerat bra med att medverka i utvecklingsarbetet?
7. Vad har fungerat mindre bra med att medverka i utvecklingsarbetet?
8. Har denna arbetsform känts bra med att ni användare är delaktiga aktivt under hela utvecklingsarbetet?
9. Vill du förändra arbetsformen i något avseende?
10. Den traditionella arbetsmetoden vid systemutveckling är att beställarna av systemet skriver en så kallad kravspecifikation tillsammans med systemutvecklarna. Utifrån denna kravspecifikation utvecklar sedan systemutvecklarna systemet utan medverkan från användare under själva utvecklingen. Vad anser du om denna arbetsmetod? Hur skulle den ha fungerat i detta fall tror du?
11. Har det krävts mycket av er som användare under projektet?
12. Skulle du ha behövt mer datorvana för att medverka och gällande vad i så fall?
13. Har utvecklingsarbetet tagit för lång tid?



14. Är det något som du vill tillägga angående detta?

## **9.4 Bilaga 4 – Intervju med kunder efter utvecklingsarbete**

### **Intervju med kunder efter utvecklingsarbetet är klart**

Syftet med denna intervju är att ta reda på hur användarna, Klippotek Passagens kunder, upplevde utvecklingsarbetet av bokningssystemet och e-handelssystemet. Utvecklingen av systemen gjordes med systemutvecklingsmetoden Prototyping och innebar en tät kontakt med användarna av de framtida systemen. Efter vi utvecklade en första prototyp av systemen baserade på frisörernas krav och önskemål var det sedan kundernas krav och önskemål som styrde utvecklingen av systemen.

Vi vill nu ställa några frågor om hur ni kunder har upplevt er medverkan och vad ni tycker om detta samarbete mellan utvecklare och användare vid Klippotek Passagens Internetetablering. Vi får be er att svara så utförligt på frågorna som möjligt och motivera era svar, då vi genom era svar kommer att utvärdera hela systemutvecklingsarbetet.

1. Har du känt dig delaktig i utvecklingsarbetet med bokningssystemet och e-handelssystemet?
2. På vilka sätt har du känt dig delaktig?
3. Tycker du det har varit kul att medverka i utvecklingsarbetet?
4. I vilka avseenden tycker du dina krav och önskemål har tagits omhand?
5. Har du krav och önskemål på de båda systemen som inte tagits omhand?
6. Vad har fungerat bra med att medverka i utvecklingsarbetet?
7. Vad har fungerat mindre bra med att medverka i utvecklingsarbetet?
8. Har denna arbetsform känts bra med att ni användare är delaktiga aktivt under hela utvecklingsarbetet?
9. Vill du förändra arbetsformen i något avseende?
10. Den traditionella arbetsmetoden vid systemutveckling är att beställarna av systemet skriver en så kallad kravspecifikation tillsammans med systemutvecklarna. Utifrån denna kravspecifikation utvecklar sedan systemutvecklarna systemet utan medverkan från användare under själva utvecklingen. Vad anser du om denna arbetsmetod? Skulle den ha fungerat bättre på detta utvecklingsarbete tycker du?
11. Har det krävts mycket av er som användare under projektet?
12. Skulle du ha behövt mer datorvana för att medverka och gällande vad i så fall?

13. Kommer du att fortsätta ringa in din bokning eller kommer du att använda bokningssystemet?
14. Kommer du att använda e-handelssystemet för att beställa eller kommer du att ringa in din beställning?
15. Är det något som du vill tillägga angående detta?