

# **Migration**

## **Dokumenthanteringssystem till Webbapplikation**

**Lenny Wall**

## **Migration**

### **Dokumenthanteringsystem till Webbapplikation**

Examensrapport inlämnad av Lenny Wall till Högskolan i Skövde, för Kandidatexamen (B.Sc.) vid Institutionen för kommunikation och information. Arbetet har handletts av Henrik Gustavsson.

**2009-06-17**

Härmed intygas att allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och att inget material är inkluderat som tidigare använts för erhållande av annan examen.

Signerat: \_\_\_\_\_

# Migration

## Dokumenthanteringssystem till Webbapplikation

Lenny Wall

### Sammanfattning

Migration mellan system används oftast när ett system har blivit föråldrat och då det märks att det inte längre är lika effektivt gentemot nyfunnen teknik. Då är det därför intressant att använda sig av migration, när det är av intresse att skapa ett nyare system utifrån det gamla. Detta arbete går igenom och löser problem vilka uppstår när en migration mellan ett dokumenthanteringssystem till en webbapplikation görs. Men fokuseringen har riktats mot att hitta den bästa migrations och övergångsmetoden för att kunna skapa en bra, enkel och effektiv migration mellan ett dokumenthanteringssystem till en webbapplikation.

**Nyckelord:** Migration, Dokument, Dokumenthanteringssystem, Webbapplikation.

## Innehållsförteckning

<b>1</b>	<b>Introduktion</b> .....	<b>1</b>
<b>2</b>	<b>Bakgrund</b> .....	<b>2</b>
2.1	Applikationsmigration .....	2
2.1.1	Praktiska exempel av applikationsmigration.....	6
2.2	Dokumenthanteringssystem.....	8
2.3	Webbapplikationer.....	10
2.4	Informationssystem.....	12
<b>3</b>	<b>Problem</b> .....	<b>13</b>
3.1	Mål.....	14
<b>4</b>	<b>Metod</b> .....	<b>15</b>
4.1	Sökning efter delmål 1 .....	15
4.2	Val av metoder för skapandet av implementation till delmål 2.....	15
<b>5</b>	<b>Resultat</b> .....	<b>16</b>
5.1	Resultat Delmål 1 .....	16
5.1.1	Migrationsmetoder på dokumenthanteringssystem.....	16
5.1.2	Övergångsmetoder mellan gamla och nya systemet .....	16
5.1.3	Mall för migreringsprocessen.....	17
5.2	Resultat Delmål 2 .....	20
5.3	Införandet.....	23
5.4	Analys .....	24
<b>6</b>	<b>Diskussion</b> .....	<b>26</b>
6.1	Sammanfattning .....	26
6.2	Reflektioner .....	26
6.3	Framtida arbete .....	28
	<b>Referenser</b> .....	<b>29</b>

# 1 Introduktion

Allt eftersom utvecklingen går framåt och datorsystemen blir allt större och mer komplicerade jämfört med äldre system, skapas det luckor mellan äldre och nyare system. Detta för att nya bättre och effektivare verktyg och komponenter skapas, vilket är av intresse att titta närmare på, samt vilka åtgärder kan göras för att minska dessa luckor. Det finns många äldre system vilket körs men är svåra att underhålla och förbättra. På grund av att dessa oftast är gjorda för ett enda ändamål och för ett specifikt system vilket gör att det är nästintill omöjligt att förnya dessa system. Äldre system blir lätt förlegade när utvecklingen av nya verktyg och komponenter ökar vilket gör att intresset för förnyandet av system växer mer, eftersom fördelarna med ett nyare system ökar. Därför är många intresserade att uppdatera sina gamla system för att kunna utnyttja nyare och bättre verktyg samt egenskaper vilket uppstått med den nya utvecklingen. Ett intressant område är webbapplikationer där fördelarna är många och uppenbara, där tillgängligheten för en webbapplikation är en av de största fördelarna. För att i praktiken gå från ett äldre till ett nyare system finns det en mängd olika faktorer vilket det måste tas hänsyn till. Det nya systemet skall innebära en förbättring och helst påminna om det gamla systemet för att på detta sätt få en bättre och positivare syn för det nya systemet. Eftersom det finns olika sätt som är både bättre och sämre är det därför intressant att hitta ett bra sätt att utföra denna migrationsövergång. Applikationsmigration är därför intressant att titta närmare på för att kunna förstå hur det fungerar när det är intressant gå från ett system till ett annat.

Exempel från tidigare undersökningar av både Zou & Kontogiannis (2000) och Sørensen et al. (2003) visar att området migration är intressant och förekommer ofta för att göra applikationsövergångar inom system. Där olika metoder och steg förekommer men där samma sak vill uppnås vilket är att applikationsmigrera från ett äldre system till ett nyare.

När ett företag kontaktades framgick det att deras behov var att göra en applikationsmigration. Detta för att eliminera deras nuvarande hantering av dokument genom att skapa en webbapplikation för att göra samma uppgift. I dagsläget används olika dokument för olika avdelningar och dessa finns mestadels i elektronisk form men också i pappersform. Elektroniska dokumenten förvaras på en server vilken är tillgänglig över deras intranät där dessa skrivskyddas när någon användare använder ett specifikt dokument. Det är därför intressant att undersöka begreppet applikationsmigration, för att få en ingående förståelse om hur det går till när en applikationsmigration av ett system görs och hur det fungerar med just migrationsövergången från ett äldre system till ett nyare. Här är det meningen att titta närmare på denna migrationsövergång från ett äldre dokumenthanteringssystem till en nyare webbapplikation. Det är också av intresse att se vilka problem som uppstår när en applikationsmigration behöver göras och hur dessa löses på ett bra sätt.

## 2 Bakgrund

I detta kapitel kommer de delar vilka är intressanta för arbetet att beskrivas för att skapa en förståelse angående de olika begreppen som migration där olika metoder visas till att gå in mera på detalj om migrationsövergång, dokumenthanteringssystemet där elektroniskt dokumenthanteringssystem förklaras till nackdelarna för detta system och webbapplikation där skillnaden mellan enkla och avancerade webbapplikationer visas.

### 2.1 Applikationsmigration

Migrering av applikationer ökar när det kommer nya system som är bättre än de gamla. Detta kan vara dels för att det nya systemet kan vara bättre inom kompatibilitet, effektivitet, billigare eller lättare att underhålla. Från Bisbal et al. (1999) framgår det att med dagens ökade teknologier inom datorområdet skapar klyftor mellan gamla och nyare system vilket gör att det är svårt att förbättra de äldre systemen genom att bara uppdatera dessa eftersom det hela tiden kommer nyare bättre teknologier med mera fördelar än de gamla.

Bisbal et al. (1999) beskriver att ett äldre system kan skapa problem genom att:

- Hårdvaran för dessa systemen är oftast föråldrad vilket gör att det blir svårare och dyrare att underhålla.
- Underhåll av mjukvaran är i dessa fall oftast dyrt med tanke på att det behövs äldre kompetens. Eftersom dokumentation ofta är bristfällig eller saknas, gör detta att hitta fel oftast blir en kostsam och tidsödande procedur.
- Integration påverkas negativt eftersom det oftast inte finns rena gränssnitt.
- Äldre system är oftast väldigt svåra om inte omöjliga att utveckla.

Från ovanstående problem hittas olika metoder vilka kommer att beskrivas där det förklaras vad varje metod kan åstadkomma för ett system samt vilka förändringar. Det intressanta är att se vad en metod kan göra för att påverka ett befintligt system och vad förändringarna kan åstadkomma för detta system.

### Wrapping:



**Figur 1: Utveckling – wrapping**

## Bakgrund

Figur 1 visar wrapping vilken används oftast av företag och organisationer där dessa vill skapa en snabb lösning på ett problem, där de kommit fram till att skapa en helt ny lösning är inte något alternativ. Saken är den att de inte räknar med att även denna nya komponent måste underhållas och uppdateras vilket gör att det i stora hela kan likväl varit värt att skapa ett helt nytt system. Med wrapping skapas en ny mjukvarukomponent för att kommunicera med den gamla och agerar server vilket gör att den inte behöver veta hur tjänsten är implementerad.

Fördelar:

- Enklare operation görs mot systemet
- Snabbt och smidigt
- Gamla systemet kan vara kvar

Nackdelar:

- Blir oftast en tillfällig lösning
- Ingen långsiktig lösning
- Mera underhåll

**Migration:**



**Figur 2: Utveckling - migration**

Migration är utifrån figur 2 en nivå mellan wrapping och redevelopment där det går ut på att skapa en lösning för problemet men där en wrapper inte uppfyller kraven och redevelopment gör en för stor ändring. Då kan migration användas där målet är att flytta från en miljö till en annan men att bibehålla ursprunglig data och funktioner detta blir därmed en bättre och mera långsiktig lösning.

Fördelar:

- Migration till annan plattform
- Inte total ombyggnad av systemet
- Gamla systemet kan användas

Nackdelar:

- Oväntade problem kan uppstå
- Mera arbete än väntat
- Oväntade uppdateringar

## Bakgrund

### Redevelopment:



**Figur 3: Utveckling - Redevelopment**

Redevelopment är den sista nivån och består av att allt skapas på nytt från början exempelvis att utforma nya verktyg, databaser och förnyande av plattform. Här görs den största förändringsfaktorn vilket framgår av figur 3. Det är en del saker vilket behöver göras och se över för att kunna skapa en ny arkitektur till detta nya system med nya funktioner, strukturer samt data. Detta är en av de mest tidskrävande och svåraste metoder eftersom allt måste skapas från grunden.

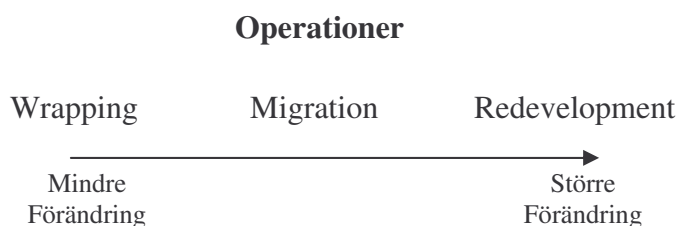
#### Fördelar:

- Ett helt nytt system skapas
- Nya komponenter och teknik kan användas
- Systemet blir mer framtidssäkert

#### Nackdelar:

- Tidskrävande och dyrt
- Inläring för användare
- Oftast specifikt för ett syfte

Utifrån ovanstående migrationsmetoder framgår det att det finns tre viktigare steg för att förbättra ett system och där dessa framgår från ovan Wrapping, Migration och Redevelopment. Det generella för dessa tre är att alla kan användas för att förbättra ett system men gör detta på sina egna sätt och egenskaper vilket också gör att det skapas en nivå för hur många ändringar som skall göras för att införa ett nyare system.



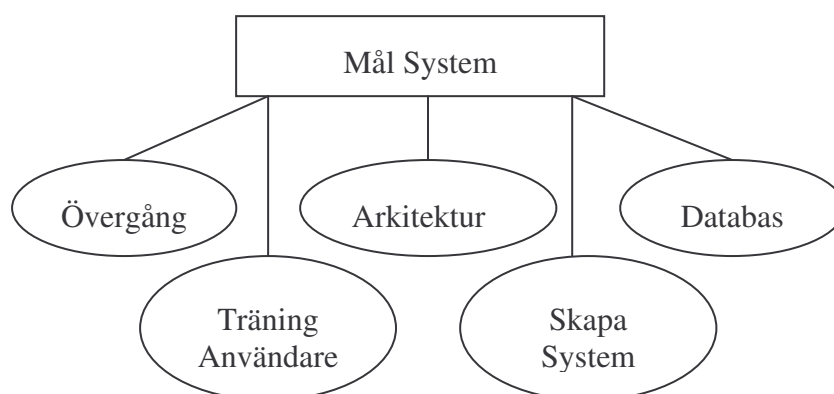
**Figur 4: Förändringsfaktorer för ett system**



## Bakgrund

Från figur 4 ses det hur nivån av en förändring påverkar systemet och där det visas att metoden wrapping används för att göra en så liten förändring som möjligt. Där redevelopment är den största förändringen för ett system.

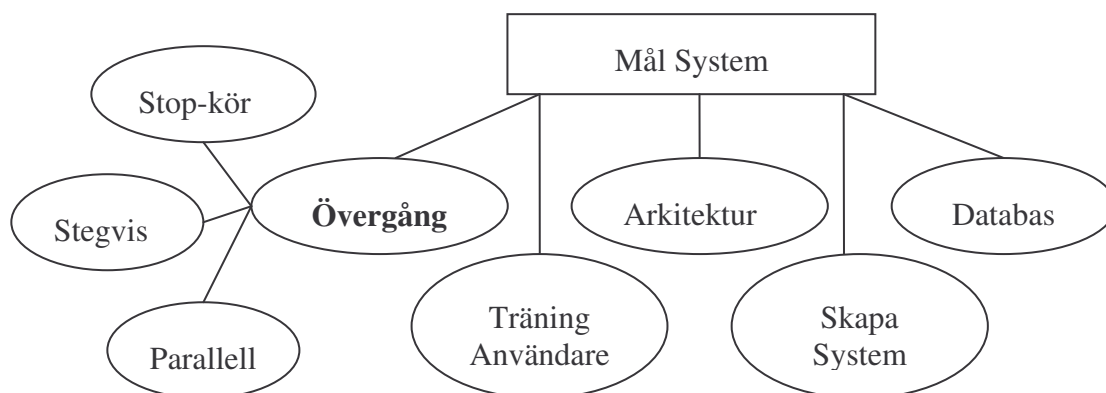
För dessa tre migrationsmetoder inom applikationsmigration är det intressant att gå in mer djupgående. Där fokuseringen för att använda en migrationsmetod är av intresse för att se på vilket sätt det går att förnya sitt system. Problematiken för att göra en applikationsmigration är ofta själva övergången. Där det är viktigt att välja den lämpligaste övergångsmetoden som passar bäst till den valda applikationsmigrationen.



Figur 5: Mål system från migration

Ett exempel för hur ett migrerat system kan se ut illustreras av figur 5 där den viktigaste fokuseringen hamnar på övergången vilket representerar det sista steget inom ett projekt inom migration.

När det sista steget med övergången skall göras från det gamla till det nya systemet finns det alltid kritiska steg som måste genomföras. På vilket sätt de sedan görs är upp till var och en enligt Simon (1992) finns det tre olika principer vilket kan användas för att skapa denna migrations övergång.



Figur 6: Migrations specificering

## Bakgrund

De tre migrations principer är enligt figur 6 och beskrivs av Simon (1992) nedan:

- **Stop-kör principen**

Denna princip fungerar så att det gamla systemet stängs av och det nya startas. Problemet är att detta sätt inte är realistiskt eftersom detta medför en väldigt stor risk att något kan gå fel vilket då medför att data går förlorad och kan medföra stora kostnader.

- **Stegvis principen**

Att använda denna princip gör att riskerna reduceras eftersom riskerna delas upp i flera olika steg för att på så sätt minska riskerna. Här testas de olika stegen och kontrolleras att föregående steg fungerar korrekt innan nästa startas. På detta sätt blir det mer kontrollerande och det syns tydligt om något blir fel varav det är lättare att felsöka.

- **Parallell principen**

Med denna princip menas det att båda systemen körs parallellt med varandra för att kontrollera att allt fungerar som det skall innan det äldre systemet slutligen stängs ner. När det nya systemet är kontrollerat och visar att det fungerar görs övergången och det nya systemet tar över.

### 2.1.1 Praktiska exempel av applikationsmigration

Migration från ett system till ett annat har gjorts av Sørensen et al. (2003) vilket har undersökt hur det är att migrera från webbaserat system till en mobil miljö. Där det framgår hur migration mellan dessa båda system diskuteras och att det är svårare än det kan verka. Eftersom texter och material från en vanlig hemsida inte alls har samma upplösning eller uppbyggnad för att på ett lätt sätt kunna stödja denna mobila miljö. Deras mål med att göra denna migrering är att öka rörligheten för sina användare. Detta genom att utveckla ett system där de kommer att kunna rapportera in sina uppdrag via mobila apparater såsom PDA eller med mobiler som stödjer WAP. Det arbetssätt de vill uppnå är med hjälp av en webbaserad applikation kunna spara information angående ett befintligt problem. Där detta sedan skickas till någon för att lösas och därefter kan denna person rapportera in med hjälp av sin mobila apparat när problemet är löst.

Ett annat migrationsexempel är av Zou & Kontogiannis (2000) där de menar att eftersom Internet växer och blir större behövs det snabbare tag att utveckla och förbättra de nuvarande webbapplikationerna. Där det också är viktigt att fortfarande bibehålla den gamla funktionaliteten utan att behöva bygga om hela systemet. Ett problem som nämns är att deras nuvarande applikationer skall kunna integreras för att kommunicera med de nyare applikationerna. En fördel är att kunna åstadkomma detta utan att behöva bygga om hela systemet.

## Bakgrund

Zou & Kontogiannis (2000) använder nedanstående steg för att skapa sin arkitektur:

- Första identifieras komponenterna som behövs för att kunna skapa migrationen från det gamla till det nya systemet. För detta steg kontrolleras det vilka komponenter och mjukvara som kan gå att återanvändas och ta till vara på.
- Återanvändning av förlegade CORBA adapter och CORBA/IDL.
- Användning av JavaBeans för att kunna integrera CORBA adapter som tjänst till den Webbaserade applikationen.
- Definiera ett skriptspråk som använder XML för att kunna hjälpa gamla komponenter.

Det framgår dock inte hur själva migrationen görs eller vilken av de migrationsmetoder de använder för att migrera. De är mera ingående på hur det går till att bygga upp och skapa migration mellan två system för att kunna använda gamla komponenter från XML till att skapa övergången till CORBA objekt.

## Bakgrund

### 2.2 Dokumenthanteringssystem

Det finns olika dokumenthanteringssystem både elektroniska och icke elektroniska. Där intresset berör elektronisk hantering av dokument. Johnston & Bowen (2005) diskuterar olika elektroniska dokumenthanteringssystem där dessa är elektroniska dokument eller elektroniska register.

Dessa två beskrivs nedan:

- **Elektroniskt registerhanteringssystem**

Ett registerhanteringssystem är ett system vilket automatiskt hanterar skapandet, användningen och underhåll av de skapade posterna. Detta hanteringssystem utför hanteringen av pappersregister och skapar på detta sätt ett elektroniskt registerhanteringssystem.

- **Elektroniskt dokumenthanteringssystem**

För ett dokumenthanteringssystem går det ut på att förbättra arbetshandlingen för dessa elektroniska dokument. För dessa system är det inte säkert att det finns något register eftersom de oftast är av den informativa karaktären där de kategoriseras genom titel för att lätt hittas än genom ett nummer.

McLeod (1996) diskuterar en dokumenthanteringsstrategi vilket tar upp hur ett lyckat dokumenthanteringssystem skall fungera. Detta för att öka intäkterna genom att effektivisera uppgiftshantering för att öka säkerheten.

Det finns också en mängd olika fördelar som är viktiga att fokusera på för att kunna ta del av ett lyckat dokumenthanteringssystem vilket visas nedan. McLeod (1996) tar upp nedanstående punkter:

- Enskilda användaren
  - Att informationen alltid finns tillgänglig
  - Ökad effektivitet på arbetsplatsen
  - Minska irritationsmoment när letandet efter information
- Organisationen
  - Effektivisera arbetet för att kunna avsluta det snabbare
  - Skapa mindre ansträngning för att göra en uppgift
  - Öka kvalitén för hur en uppgift utförs
  - Ökning av omsättning

## Bakgrund

- Samhället
  - Organisationens processer är tillgängliga och kan övervakas
  - Att organisationen använder sig av lagar och regler
  - Livskvalitén förbättras
  - Äldre information är tillgänglig och pålitlig

Med dessa tre steg och förbättringar som beskrivs under respektive punkt skall användandet av dokument och register förbättras för alla parter.

Kranch (1998) beskriver att det finns allvarliga nackdelar med elektroniska dokument och där den största nackdelen är att bevara dessa dokument i dess originalform. Eftersom en användare som har tillgång till detta dokument enkelt kan ändra och spara denna ändring vilket reflekterar i att dokumentet inte längre är i sin originalform. Det är också viktigt att ta hänsyn till information som inte är till för obehöriga det vill säga om en användare av misstag eller mening hittar känslig information i ett dokument.

## Bakgrund

### 2.3 Webbapplikationer

Med Internet växande framfart gör att det har utvecklats till ett av de största nätverken där information finns tillgänglig dygnet runt. Ginige & Murugesan (2001) menar att eftersom nyare tekniker då utvecklas och lätt kan användas. Gör att detta överträffar de äldre systemen och applikationerna. Vilket medför att alla blir drabbade och måste utvecklas för att vara med i framtidens framfart. Där detta påverkar alla från enskilda företag till affärer och större industrier som vill vara med att utnyttja dagens teknik. Nu har även de äldre traditionella information och databas systemen övergått till att finnas tillgängliga över Internet vilket också gör att det måste vara pålitligt och fungera ordentligt eftersom alla numera litar på dessa webbaserade system och applikationer som finns till förfogande.

Deshpande et al. (2002) Beskriver vad som skiljer sig mellan en enkel och avancerad webbapplikation.

<b>Enkelt Webbapplikation</b>	<b>Avancerad Webbapplikation</b>
Informationen är statisk	Informationen är dynamisk
Enkelt att navigera	Svårt att navigera och hitta information
Begränsad användbarhet	Integrerar med databas och andra schemaläggning och planeringssystem
Prestanda är inte prioriterad	Hög prestanda är prioriterad
Skapat av en enskild eller en mindre grupp	Kräver oftast en större grupp för skapandet
Lätt att skapa	Behöver en konfigurations kontroll

**Tabel 1: Egenskaper för enkel och avancerad webbapplikation**

Från tabel 1 ser man tydligt hur det ligger till med de olika webbapplikationerna. Eftersom dessa är byggda i en miljö vilket hela tiden utvecklas och ändras kräver det att dessa applikationer hela tiden underhålls och utvecklas. En webbapplikation behöver vara funktionell, pålitlig, skalbar, underhållbar och säker för att kunna överleva.

## Bakgrund

Deshpande et al. (2002) Beskriver vidare hur olika kategorier används inom webbapplikationer

Kategorier	Exempel
Informativt	Nyhetsbrev, Service manualer, Produkt kataloger, E-böcker
Interaktivt	Registrerings formulär, Spel
Transaktion	Beställning av varor eller tjänster, Bank
On-line samhällen	Chat grupper, Marknadsplatser, Auktioner
Webb portaler	Elektronisk shopping
Arbetsflöde	Planering och schemaläggning, Lager hantering, Övervakning

**Tabel 2: Kategorier för webbapplikationer**

Från tabel 2 hittas några olika kategorier där en webbapplikation kan ligga under. För att kunna skapa någon av dessa webbapplikationer behövs en programmeringsmiljö och ett programmeringsspråk. Füricht et al. (2002) beskriver .NET konstruktionen med språket C-sharp vilket en ny mjukvaruplattform för att kunna bygga och utveckla Windows samt webbapplikationer. Själva .NET teknologin påminner om Java teknologin där C-sharp liknar Java. Eftersom .NET miljön med C-sharp har fördelen att kunna utveckla webbapplikationer och har stöd för en mängd olika verktyg såsom eventhantering och databasstöd. Detta gör att .NET är mycket intressant för att utveckla webbapplikationer i. En fördel med .NET är att program som skapas med detta redskap är gjort för Windows och körs mera effektivt mot andra liknande redskap exempelvis Java.

Ett problem som tas upp av Joshi et al. (2001) är säkerheten för dessa applikationer och hur användare hela tiden blir utsatta för olika risker. Att använda sin webbläsare medför många risker och där information om användarnamn eller maskinamn kan sparas för att skapa en användarprofil. Data vilket skickas mellan webbapplikationer och användaren såsom cookies kan användas för att hämta information för olika syften. Attackerna ökar vilket har gjort att brandväggar vilket hjälper till att förhindra data från att ta sig in i systemet har blivit populära för att försvara mot dessa attacker.

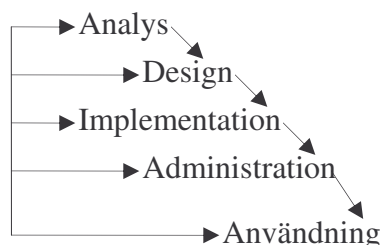
## Bakgrund

### 2.4 Informationssystem

Hevner et al. (2004) framgår det att för ökad av effektiviteten och verkningsgraden inom en organisation implementeras ett informationssystem.

Utvecklingen av ett informationssystem innehåller olika faser vilket är analys, design, implementation, administration och användning av informationssystemet. Dessa faser används för att utveckla ett informationssystem och där varje fas är en grundpelare för nästa steg.

Henriksson & Strandberg (2003) beskriver vattenfallsmetoden vilket grundar sig på faserna för informationssystemet. Där beskrivs olika aktiviteter vilket är en parallell mot faserna från det ovan nämnda informationssystemet.



Figur 7: Vattenfallsmodellen

- **Analys**

Från analysfasen skapas en kravspecifikation vilket är ett kontrakt över vad själva projektet går ut på. Även vilka krav som måste tillgodoses för det tilltänkta systemet.

- **Design**

I designfasen skapas system och mjukvarudesignen vilket beskriver hur arkitekturen för systemet skall se ut och fungera. Här beskrivs algoritmer och metoder vilket kan tänkas behövas i systemets design samt hur användargränssnittet skall utformas.

- **Implementation**

För denna fas skapas mjukvaran utifrån det framkomna ur designfasen samt testning av de skapade modulerna.

- **Administration**

Här testas de olika modulerna för att kontrollera att systemkraven är uppfyllda. Efter att testningen är klar levereras systemet till kunden.

- **Användning**

Under denna fas hamnar drift och underhåll vilket pågår så länge systemet är i drift där fel och andra komplikationer korrigeras.



### 3 Problem

För att i praktiken gå från ett äldre till ett nyare system finns det en mängd olika faktorer vilket det måste tas hänsyn till. Bisbal et al. (1999) berättar att fler problem finns med migrering, vilka är intressanta att undersöka och där det framgår hur övergången mellan olika system kan fungera. Meier & Sprague (1996) tar upp och behandlar hur ett dokumenthanteringssystem fungerar och visar både fördelar och nackdelar med dessa system. Det finns många exempel på olika applikationsmigreringar exempelvis beskrivs två i kapitel 2.1.1, det finns i litteraturen inte några beskrivna applikationsmigrationer av dokumenthanteringssystem. Att undersöka de specifika problem och egenskaper som just dokumenthanteringssystem har jämfört med andra applikationer är därför intressant.

Det nya systemet skall innebära en förbättring och helst påminna om det gamla systemet för att på så sätt få en bättre och positivare syn för det nya systemet. Eftersom det finns olika sätt vilka är både bättre och sämre är det därför intressant att hitta ett bra sätt att utföra denna övergång. Ett företag ville ha hjälp med var att eliminera deras nuvarande hantering av dokument och att skapa en webbapplikation för att göra samma uppgift. I dagsläget används olika dokument för olika avdelningar och dessa finns mestadels i elektronisk form men också i pappersform. De elektroniska dokumenten förvaras på en server vilken är tillgänglig över deras intranät och där dessa skrivskyddas när någon användare använder ett specifikt dokument. Kranch (1998) tar upp och beskriver hur säkerheten för dokument hanteras och där det framgår att det är svårt att bevara ett dokumentets ursprungsform. Vilket kommer att resultera i att det behövs ett visst säkerhetstänkande för skapandet av denna nya webbapplikation.

Problemet som skall lösas är att undersöka hur det fungerar att göra en migration mellan ett dokumenthanteringssystem till en webbapplikation med databasstöd. Området inom migration av desktopapplikationer till webbapplikationer är något nytt och välbeskrivet område vilket framgår av Zepeda, J & Chapa, S (2007) och Pigora, Mary & Thompson Eric (2008). Dock har inte det specifika problemet att migrera dokumenthanteringssystem undersökts lika noggrant men är minst lika intressant att undersöka.

## Problem

### 3.1 Mål

Migration av applikationer är ett stort område och därför behövs tidigare metoder för migration granskas mer i detalj. Resultat från bland annat området informationssystem behövs ta hänsyn till. Detta för att kunna göra en bra bedömning om hur det på ett bra sätt kan göra denna övergång mellan de båda systemen.

Arbetet har två tydliga delmål där dessa är:

#### Delmål 1:

Det är av intresse att med resultaten kunna skapa en slags mall. Detta för hur det metodiskt kan skapa denna applikationsmigration. En viktig del är också att välja rätt migrationsövergång mellan de olika typerna av system.

#### Delmål 2:

Ett slutgiltigt mål är att med hjälp av en implementation bygga ett system, Vilket visar hur dessa tekniker kan användas som en grund för att illustrera tekniska problem och lösningar när det gäller applikationsmigration från dokumentbaserade system till en webbaserad lösning. Ginige & Murugesan (2001) diskuterar och beskriver hur webbapplikationer påverkar alla som vill vara med att använda den nya tekniken och hur den kan användas för att förbättra sina gamla system.

För att ses som en helt lyckad migrationsprocess måste det färdigbyggda systemet uppfylla nedanstående kriterier. Dessa är datadelning, flexibilitet, övergång och säkerheten vilket är viktiga för uppbyggnaden av den nya webbapplikationen.

- Datadelning är en viktig del som är intressant att lösa och titta närmare på just för att den vägen visar hur det på ett bra sätt går att göra för att få med de olika delarna från ett dokument till att realisera detta i en och samma databas. Där dessa används för att beskriva den sparade informationen. Datadelningen är grunden för att veta vilken information där den används för skapandet av den nya databasen.
- Flexibilitet är en annan viktig kriterie. Med tanke på att det finns delar från de olika dokumenten som är specifika för just en avdelning och där det finns delar som inte alls behövs för en annan avdelning.
- Säkerheten och åtkomsten är ett annat viktigt delproblem att lösa. Dokumenten kommer inte längre att finnas kvar, vilket gör att säkerheten måste inkluderas i tänkandet för att bibehålla de rättigheter som redan finns inom det dokumentbaserade systemet men inte är vanliga för webbapplikationer.

### 4 Metod

Arbetet går ut på att undersöka hur applikationsmigration mellan ett traditionellt dokumenthanteringssystem till ett nyare webbapplikationssystem fungerar. Där vikten läggs på att visa hur delproblemen vilka beskrivs i kapitel 3 löses på bästa möjliga sätt.

#### 4.1 Sökning efter delmål 1

Information om hur applikationsmigration mellan system fungerar finns det mycket beskrivet inom vetenskapliga studier, vilket gör att sökandet efter lämpliga migration och övergångsmetoder därför kommer att inriktas på att göra en litteraturstudie för att hitta de olika migration och övergångsmetoder. Sedan för att välja ut den bästa migration och övergångsmetod till det specifika systemet blir det till att hitta argument, vilket genererar för och nackdelar för att bestämma vilken av denna migrationsmetod kan användas och är mest lämplig just för dokumenthanteringssystem.

#### 4.2 Val av metoder för skapandet av implementation till delmål 2

Meningen är att skapa en implementation för att kunna praktiskt uppnå kunskap och förståelse för hur en applikationsmigration kan gå till. Genom olika steg för hur ett system byggs upp från början till att slutligen hamna i produktionen. För att ersätta ett äldre system det vill säga att i verkligheten se hur denna applikationsmigration görs. För att på detta sätt kunna avgöra om de valda metoderna löser problemet. Samtidigt för att hitta de bästa metoderna görs tester mot implementationen med de olika metoderna vilka hittades i delmål 1, för att kunna avgöra om en lösningsmetod passar eller ej. Vilket avfärdas om kriterierna för den tänkta applikationsmigrationen inte uppfylls. Genom att genomföra en implementation kan teorier testas. Vilket kan användas för att uppnå den praktiska kunskapen för hur en applikationsmigration kan göras på ett enkelt och effektivt sätt.

## 5 Resultat

I detta kapitel redovisas hur problemet löstes samt dess delproblem. För att kunna redovisa hur en applikationsmigration från ett dokumenthanteringssystem till en nyare webbapplikation går till på ett enkelt och smidigt sätt.

### 5.1 Resultat Delmål 1

För just applikationsmigrering av ett dokumenthanteringssystem är det intressant att hitta en bra och passande migrationsmetod. Detta för att lösa problemet med att applikationsmigrera från detta gamla system till nyare webbapplikation och hur detta kan göras på ett effektivt sätt.

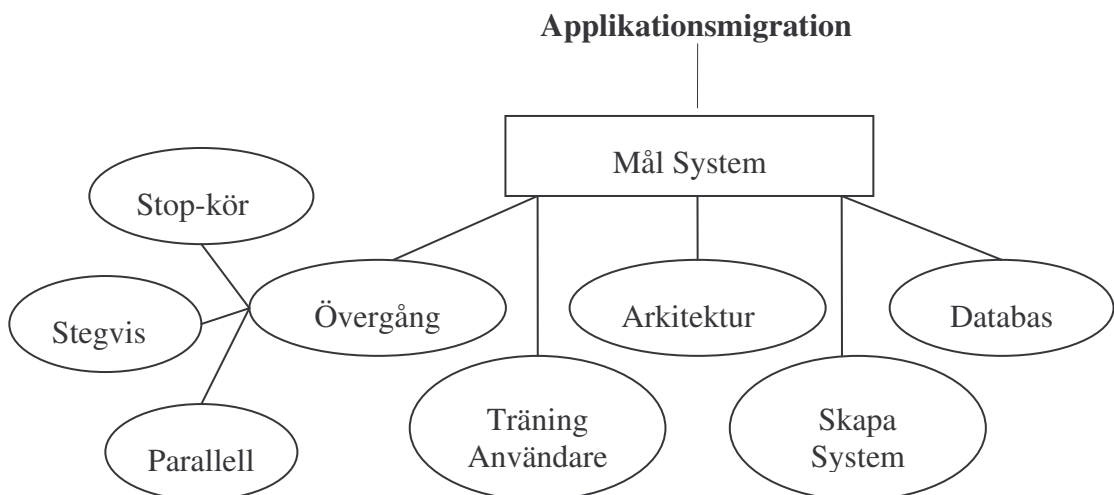
#### 5.1.1 Migrationsmetoder på dokumenthanteringssystem

**Wrapping:** Att använda denna migrationsmetod vilket diskuteras i kapitel 2.1 för att göra en applikationsmigration mot ett dokumenthanteringssystem skulle innebära att en mindre förändring görs mot systemet. Detta för att uppnå en snabb och smidig förbättring. Det skulle kunna vara att skapa en mindre komponent, för att exempelvis automatiskt skriva in saker i ett dokument eller att skapa en uppdatering för att lösa ett problem. För dessa ändamål skulle metoden wrapping passa ypperligt att använda.

**Migration:** En andra migrationsmetod för att utveckla ett system är migration där migration för ett dokumenthanteringssystem skulle kunna vara att skapa en lösning, för att kunna använda samma dokumenthanteringssystem på en annan plattform. Vilket då gör att förändringen är större och besvärligare än för metoden wrapping. Eftersom mycket mer måste göras för att migrera hela systemet till den andra plattformen. För att kunna göra detta miljöbyte måste källkod finnas och helst dokumentation över detta dokumenthanteringssystem.

**Redevelopment:** Den sista och mest krävande migrationsmetoden är redevelopment vilket används för att skapa ett helt nytt system. För att göra redevelopment på dokumenthanteringssystemet skulle det kunna vara att ett helt nytt dokumenthanteringssystem skapas från grunden. Där ny teknik introduceras och används för att göra det nya systemet framtidssäkert.

#### 5.1.2 Övergångsmetoder mellan gamla och nya systemet



Figur 8: Migrationsmetoder

## Resultat

Från figur 8 visas det hur migrering kan se ut med det nya målsystemet där övergången är ett eget steg. Varvid det finns tre olika övergångsmetoder för att gå till väga med en övergång från gamla till det nya systemet. Dessa tre övergångsmetoder är beskrivna i figur 8 och är stop-kör, stegvis och parallellt. Tillsammans utgör dessa tre de eftersökta metoderna för en övergång mellan två system.

**Stop-kör:** Att använda stop-kör principen för att införa en ny webbapplikation är riskabelt. Vilket beskrivs från kapitel 2.1 att det inte är realistiskt eftersom det medför en för stor risk. Med att stänga ner det gamla systemet för att starta det nya vilket antas fungera direkt från början.

**Stegvis:** Att introducera den nya webbapplikationen stegvis och mer kontrollerat genom att skapa moduler, vilka testas och introduceras till den nyare applikationen stegvis är mindre riskabelt och bra sätt för att hitta och korrigera fel vilket kan uppstå men enkelt hittas och lösas stegvis.

**Parallellt:** Om den nya webbapplikationen och gamla dokumenthanteringssystemet körs parallellt minskar riskerna för att störa det gamla systemet, vilket är bra eftersom då kan det gamla köras vid sidan om det nya för att se om det fungerar som det skall. Tills att slutligen stänga ner och ersätta gamla mot det nya systemet.

### 5.1.3 Mall för migreringsprocessen

Meningen med att skapa denna mall är att visa hur en migreringsprocess går till. Syftet är att visa hur det på ett enkelt och effektivt sätt fungerar att göra en applikationsmigration. Denna mall har följts för att kunna göra en korrekt migrationsprocess av de migrations och övergångsmetoder vilka finns beskrivna i kapitel 2.1 Om applikationsmigration. Med mallen är det meningen att visa hur tillvägagångssättet gick till när en applikationsmigration gjordes mellan två system, där dessa är dokumenthanteringssystem och webbapplikation. Genom att applikationsmigrera ett elektroniskt dokumenthanteringssystem vilket beskrivs i kapitel 2.2 Där detta system används för att underlätta och hantera information åt användaren, om en applikationsmigrering skall göras för detta system. Bör dokumentation och alla resurser angående det berörda systemet finnas tillgängliga för att en applikationsmigration skall kunna göras på enklaste sätt. Det gamla systemet är inte specifikt för sin uppgift och kan användas för en mängd olika saker, vilket gör det flexibelt men inte effektivt. För att kunna skapa det nya systemet och göra denna implementation är det en fördel att veta för och nackdelar angående det gamla systemet.

Fördelar:

- **Flexibelt och mångsidigt**

Detta är ett generellt system och är inte gjort för något specifikt syfte mer än att spara information

## Resultat

- **Lätt att använda**

Det är enkelt att förstå sig på systemet och att i gränssnittet hitta det som söks

- **Säkerheten**

Säkerheten fungerar på det sätt att när någon skriver låses filen och när informationen sparas låses filen upp

Nackdelar:

- **Svårt att hitta**

Filerna från systemet är svåra att hitta om användaren inte vet exakt vad denna letar efter

- **Ej utbyggbart**

Systemet är inte utbyggbart eftersom det är skapat att fungera generellt för alla möjliga sorters behov och att källkod samt tillhörande dokumentation ej är tillgängligt

- **Gammalt**

Det nya systemet har blivit föråldrat på grund av den nya tekniken

Nu när nackdelar och fördelar är listade går det lättare att utesluta och välja en migrationsmetod att testa för en kommande implementation. De tre migrationsmetoderna är wrapping, migration och redevelopement vilka beskrivs i kapitel 5.1.1 hur dessa skulle deras applikationsmigration skulle påverka det nuvarande systemet. Att utveckla ett system med någon av dessa tre migrationsmetoder innebär att den eftersökta migrationsmetoden måste kunna uppfylla de ovanstående för och nackdelarna vilka är listade ovan för det gamla systemet. Det är mer intressant att direkt titta och jämföra nackdelarna mot de listade migrationsmetoderna. Nackdelen beskriver att det gamla systemet inte är utbyggbart samt gammalt. Detta gör att det skulle bli mycket jobba om inte omöjligt att använda migrationsmetoderna wrapping och migration för att skapa det nya systemet. Däremot om migrationsmetoden redevelopement används kan alla fördelar och nackdelar uppfyllas för det nya systemet. Därför har just denna migrationsmetod valts att användas för den tilltänkta implementationen. Anledningen till att just redevelopement blev den vinnande migrationsmetoden är att det nya systemet byggs upp från grunden med den senaste tekniken inom webbapplikationer. Vilket beskrivs i kapitel 2.3 det är tänkt att med redevelopement utveckla en avancerad webbapplikation från grunden. Där det går att göra webbapplikationen specifik och anpassat för det speciella syftet.

Att inom applikationsmigration välja hur en övergång mellan två system skall gå till beror på vad det är för system, samt dess egenskaper och vilka resurser det finns tillgång till. Från kapitel 5.1.2 beskrivs att det finns tre olika övergångsmetoder för att göra en migrationsövergång från det gamla systemet till den nya webbapplikationen. Dessa tre är stop-kör, stegvis och parallellt där stop-kör har nackdelar att det är riskabelt och inte realistiskt att använda denna övergångsmetod eftersom det gamla systemet stängs av och det nya startas, vilket skapar problem om det inte fungerar direkt. Den andra övergångsmetoden är stegvis vilket gör att systemet byggs upp i

## Resultat

moduler, för att lättare kunna hitta och rätta till fel där de uppstår och varje modul introduceras stegvis. Den sista övergångsmetoden är den parallella och är dessutom den bästa övergångsmetoden att introducera för en ny webbapplikation och detta med tanke på att applikationen kan skapas och köras parallellt med det gamla systemet vilket gör att testning och skapandet inte på något sätt stör den befintliga organisationen.

Från nackdelarna ovan har en migrations och övergångsmetod valts för att användas att lösa delmål 2. Detta innebär att de två största stegen för en applikationsmigration har gjorts för att kunna börja utveckla den nya applikationen.

**Steg 1:** Analysera det nuvarande systemet och hitta de krav för att kunna avgöra vilken migrationsmetod som är bäst för detta dokumenthanteringssystem.

**Steg 2:** Jämför och kontrollera vilken övergångsmetod som är den bästa för just denna applikationsmigration.

Sammanfattningsvis har dessa ovanstående steg gjorts. Där meningen är att kunna på ett enkelt sätt följas för liknande problem och är tänkt att illustrera ett sätt att göra en applikationsmigration för ett dokumenthanteringssystem till en webbapplikation. Där det är meningen att kunna använda detta kapitel som en migrationsmall.

## Resultat

### 5.2 Resultat Delmål 2

Genom att skapa en .NET implementation av den nya webbapplikationen går det att praktiskt angripa delmål 2 vilket diskuteras i kapitel 3 för vilka kriterier som måste vara uppfyllda för att lyckas med denna migrationsprocess. Det är också meningen att illustrera med bilder från den nya skapade webbapplikationen för att kunna visa om kriterier har uppfyllts eller inte.



	Namn	Bakgrund	Prioritet	Typ
±	 Transportsäkring (gul)	<a href="#">Öppna</a> Missfärgad och sliten	Mellan	8D
±	 Dörrlist trasig	<a href="#">Öppna</a> Dörrlisten kläms	Mellan	Förbättring
±	 Dörrens gångjärn	<a href="#">Öppna</a> Felplacerade gångjärn	Mellan	Förbättring
±	 Grönsakslåda missformad	<a href="#">Öppna</a> Missformad grönsakslåda	Mellan	Haveri
±	 Kompressorljud	<a href="#">Öppna</a> Surrande ljud från kompressorn	Mellan	Förbättring

Figur 9: En användares startsida

Figur 9 visar när ett antal aktiviteter och underaktiviteter hämtas från den skapade databasen för att presenteras och listas i webbapplikationen och meningen med att visa denna bild är att den illustrerar alla aktiviteter för en specifik användare med administrations behörighet. Inom säkerhetssynpunkt fungerar listandet av aktiviteter på det sättet att användarens egna aktiviteter visas och det är bara sina egna som går att administrera.



## Resultat

Aktivitets Databas		
<a href="#">Start</a> / Översikt		
<b>Aktivetsnamn</b> Transportsäkring (gul)		
<b>Aktivetsområde</b> Q	<b>Aktivitetstyp</b> 8D	<b>Problem</b> Missfärgad och sliten
<b>Prioritet</b> Mellan	<b>Referens</b> as	
<b>Artikel Namn</b> 1	<b>Artikel Nr</b> 2	<b>Grundorsak</b> Är ej anpassad till OMEGA-packningen
<b>Aktivetsansvarig</b> MALMSTRÖM JENNY	<b>Begärd Tid</b> 2009-06-07	
<b>Beräknat Slut Datum</b> 2009-06-12		<b>Åtgärd</b>
<b>Reviderat Slutdatum</b> -----	<b>Aktivitet klar</b> -----	

Figur 10: Datadelning

Med figur 10 är det meningen att illustrera den hämtade informationen vilket kommer från den skapade databasen där datadelningen visar de olika realiserade delarna vilket härstammar från det gamla dokumenthanteringssystemet men där de generella delarna valts för i detta fall representera en aktivitet i den nya webbapplikationen.

8D - Rapport		
<b>1 Activity name</b>	<b>Upptäckt datum</b>	<b>Utfärdare</b>
Kondensskydd	2009-05-18	73422
	<b>Artikel namn</b>	<b>Artikel nummer</b>
<b>Problembeskrivning</b>		
Kondensskyddet skadas ofta på dom produkter där kondensatorskyddet slutar nedanför trådkondensatorn. Pga att trådkondensatorns trådar är vassa och skaver hål när man monterar trådkondensatorn över kondensskyddet och när man gör rördragningen i processen.		
<b>2 Kortsiktig åtgärd</b>		<b>Införd datum</b>
		Ej Färdig!
		<b>Ansvarig</b>
		-----

Figur 11: Flexibilitet

Figur 11 visar hur flexibilitet uppnåtts mellan det gamla dokumenthanteringssystemet och den nya webbapplikationen. Det intressanta är att bilden visar en rapportmall som är genererad av den nya webbapplikationen och används för en specifik avdelning.

## Resultat

Där grunden fortfarande är aktiviteter vilka sammanfattas i denna rapportmall. Detta gör webbapplikationen användarvänlig och flexibel för olika behov.

Från ovanstående bilder över den nyskapade webbapplikationen syns det tydligt hur de olika kriterierna för delmål 2 är lösta och uppfyllda. Där säkerheten garanterar att ingen annan än användaren själv kan ändra i sina aktiviteter. Datadelningen uppfylls genom att använda tillvaratagen information för att realisera och bygga upp den nya databasen samt webbapplikationen vilket visas från figur 10. Sist men inte minst flexibiliteten löstes också på ett bra sätt för den nya implementationen av webbapplikationen eftersom webbapplikationen är dynamisk mot respektive avdelning och behov. Detta gör att kraven för dessa kriterier är uppfyllda och löser delmål 2.

## Resultat

### 5.3 Införandet

Detta kapitel förklarar hur införandet av den nya webbapplikationen skall införas för organisationen på ett enkelt och effektivt sätt.

Migrationsmetoden redevelopment valdes för denna implementation innebär det att en helt ny webbapplikation skall skapas för att på detta sätt förbättra, effektivisera och ersätta det gamla dokumenthanteringssystemet. Att skapa en webbapplikation från grunden med hjälp av litteraturstudier. Detta gjordes genom att först hitta en bra migrationsmetod. Där denna efter att jämfört för och nackdelar blev migrationsmetoden redevelopment, vilket kan hittas i kapitel 5.1.3 där fördelarna med redevelopment var övervägande för att använda just denna migrationsmetod, med att utveckla den nya webbapplikationen. Därefter behövdes en modell att följa för att ge struktur på uppbyggandet av detta nya system. Varav vattenfallsmodellen valdes eftersom den används inom utveckling för informationssystem. Kapitel 2.4 visar anledningen till att just denna modell valdes var för att en kravspecifikation redan var upprättad och skapas från resultatet av analysfasen och där organisationen insisterade på att användandet av denna modell skulle användas. Att sedan för övergången av en migration använda den parallella övergångsmetoden för att introducera systemet för organisationen berodde på att det gamla dokumenthanteringssystemet användes parallellt medan det nya byggdes upp för att slutligen gå över helt till det nya systemet.

**Steg1:** Det första steget för att skapa en implementation var att först bilda sig en uppfattning om organisationen och hur det gamla dokumenthanteringssystemet fungerade samt vad deras behov var, vilket framgick från kravspecifikationen men det är viktigt att få en insikt i hur allt skall fungera och hänga ihop. Vilket innebar att intervju med beställarna gjordes för att eliminera eventuella missförstånd och för att få information om hur deras förväntningar av systemet var.

**Steg2:** För det andra steget förklarades hur systemet skulle fungera och vilka verktyg, programmeringsspråk samt programmeringsmiljö som fanns till förfogandet för att skapa denna nya webbapplikation. Här framgick det också att det fanns tillgång till en egen testmiljö för att kunna skapa den nya webbapplikationen medan det gamla dokumenthanteringssystemet fortfarande användes.

**Steg3:** Här började det genast bli klart att en ny applikation skulle skapas genom att använda den nyare tekniken webbapplikation, vilket beskrivs i kapitel 2.3 där det ska vara en avancerad webbapplikation med dynamiskt innehåll. Det är också i detta steg där insikten om att migrationsmetod redevelopment bör användas för att göra denna migration.

**Steg4:** I detta steg intervjuas beställarna angående den nya webbapplikationen för att få en försäkring om att allt är korrekt. Med implementationen av den nya webbapplikationen och där det nu diskuteras hur migrationsövergången från det gamla till det nya systemet skall gå till.

## Resultat

Att införa den nya webbapplikationen för en organisation innebär att en övergångsmetod behövs. Från kapitel 5.1.3 framgår det att den parallella övergångsmetoden är den bästa för introduceringen av den nya webbapplikationen, vilket gör att systemet används och utvecklas samtidigt som det gamla körs vid sidan om. När intervju med beställarna gjordes framgick det att de tyckte då att det gamla dokumenthanteringssystemet skulle användas parallellt medan det nya utvecklades till att sedan helt ta över verksamheten.

### 5.4 Analys

Inom analys skall det avgöras om det praktiska arbetet löser det aktuella problemet från kapitel 3. Det är därför intressant att genom testning mot förstudien kunna kontrollera om den nya implementationen uppfyller de krav och förväntningar organisationen hade. För att påvisa att problemet från kapitel 3 är löst är det intressant att först titta närmare på de idéerna från förstudien vilket har sammanfattats till kriterierna i kapitel 3.

De delmål vilka beskrivs i problemkapitlet har uppfyllts och kommer att redovisas här nedan genom att förklara vad som gjorts inom varje delmål.

**Delmål 1:** Detta delmål går ut på att skapa en typ av migrationsmall vilket har realiserats i kapitel 5.1.3 där det går att följa hur denna applikationsmigration gått till. Samt följa genom för och nackdelar med att ta fram rätt migration och övergångsmetod just för denna applikationsmigration, mellan ett dokumenthanteringssystem och en webbapplikation uppfylldes. Skapandet av denna migrationsmall gör att delmål 1 är uppfyllt och där valet av rätt övergångsmetod också beskrivs i denna migrationsmall.

**Delmål 2:** Efter att den nya webbapplikationen skapats med den valda migrationsmetoden vilket beskrivs i kapitel 5.1.3 där det framgår att den är mest fördelaktiga migrationsmetoden för introducerandet av det nya systemet. Det fördelaktiga med migrationsmetoden redevelopment för att utveckla denna webbapplikation är just att allt byggs upp från början. Från kapitel 5.2 vilket beskriver implementationen av den nya webbapplikationen och där visas hur applikationen ser ut efter att skapats med migrationsmetoden redevelopment. Denna implementation verifierar att det faktiskt går att använda denna migrationsmetod för att lösa detta delmål.

För skapandet av den nya webbapplikationen skulle fungera enligt kraven delades problemet upp i olika kriterier. Detta för att på detta sätt kunna få bättre översikt i vad implementationsproblemet egentligen bestod av.

För att lyckas med detta delmål måste alla underliggande kriterier från kapitel 3 uppfyllas. Dessa har uppfyllts och redovisas också nedan hur varje kriterium för detta delmål har uppfyllts vilket också kan ses i kapitel 5.2 där bilder över implementationen visas.

**Datadelning:** Är ett av de kriterier vilket måste uppfyllas för att kunna visa att de olika delarna har kunnat sparas och användas från det gamla dokumenthanteringssystemet till den nya webbapplikationen. Figur 10 i kapitel 5.2

## Resultat

vilket visar en bild från webbapplikationen och där det syns tydligt att datadelning är uppfyllt, genom de fetstilta rubrikerna vilket är de olika delarna vilket härstammar från gamla dokument men nu är realiserade i den nya webbapplikationen. En verifikation av problemspecifikationen beskriver att datadelning uppfylls genom att migrera gamla delar från det gamla systemet till det nya vilket gör att detta har uppfyllts.

**Flexibilitet:** inom den nya webbapplikationen har flexibilitets kriterium uppfyllts genom att använda samma layout med olika dynamiska innehåll för olika behov. Detta framgår av figur 11 i kapitel 5.2 där en rapportmall visas för att sammanfatta olika delar av sparad information i ett och samma dokument. Dessa delar kan också nå individuellt vilket illustreras i figur 9. Från figur 10 vilket beskriver datadelning visar hur rubrikerna ser ut för en specifik avdelning men även detta kan ändras för att på detta sätt uppfylla flexibilitet. Detta är en ny och förbättrad flexibilitet jämfört mot det gamla dokumenthanteringssystemet eftersom dessa möjligheter inte fanns överhuvudtaget.

**Säkerheten:** att bibehålla säkerheten för den nya webbapplikationen är ett viktigt kriterium att uppfylla. Med figur 9 i kapitel 5.2 visas det hur en användare har tillgång till sin information och ingen annans. För den nya webbapplikationen kontrolleras hela tiden behörigheten för den visade informationen. Detta gör att om informationen skall modifieras på något sätt, görs en behörighetskontroll innan. Eftersom den nya applikationen är lika säker som den gamla gör detta att säkerhetskriterium är uppfyllt.

## Webbapplikation

- Datadelning
- Flexibilitet
- Säkerhet

### Figur 12: Problem och delproblem

Utifrån figur 12 kan respektive metod eller modell kopplas ihop med respektive problem för att på detta sätt kunna visa att detta löser problemen från kapitel 3. För själva huvudproblemet från figur 12 vilket är webbapplikation. Används migrationsmetoden redevelopment för att kunna lösa problemet samt delproblemen. Där datadelning och flexibilitet använder analysfasen från vattenfallsmodellen vilken beskrivs i kapitel 2.4 för att skapa den detaljerade kravspecifikationen. Säkerhetskriteriet använde implementationsfasen för samma modell. Detta för att kunna skapa och använda denna säkerhetskontroll för varje användare. Sammanfattningsvis löstes problemen med respektive utvalda metod från kapitel 5.2 vilket innebär att denna lösningsmetod fungerar för att applikationsmigrera och utveckla ett gammalt dokumenthanteringssystem till en ny webbapplikation samt att delmålen från kapitel 3 uppfylldes.

# 6 Diskussion

## 6.1 Sammanfattning

Problemet vilket är av intresse att lösa är att göra en applikationsmigration från ett dokumenthanteringssystem till en webbapplikation. För att lösa detta problem måste delmålen med underliggande kriterium uppfyllas för att kunna skapa den nya webbapplikationen. Utifrån resultat kapitlet hittas en lämplig migrationsmetod vilket för denna webbapplikation var bäst med migrationsmetoden redevelopment, och där övergångsmetoden användes den parallella lösningen för att inte störa den befintliga organisationen. Slutligen från analysen framgår det att både migrationsmetoden och övergångsmetoden uppfyller och löser det slutliga problemet med att göra denna applikationsmigration mellan dessa olika system.

Från kapitel 3 vilket beskriver problemet och dess underliggande delmål framgår det att utifrån dessa delmål hittas kriterierna mall, datadelning, flexibilitet, övergång och säkerhet. Från kriteriumet mall är uppfyllt genom att lösa metoden för detta arbete och där datadelning samt flexibilitet handlar om att hitta och hantera den nya informationen för uppbyggnaden av den nya webbapplikationen. Övergångskriteriet löstes när det nya systemet användes och sattes i bruk. För säkerhetskriteriet löstes detta med att kontrollera vilken användaren var för att se om behörighet fanns eller ej. Eftersom dessa kriterier är lösta gör detta att delmålen också är uppfyllda vilket resulterar i att själva problemet löstes. Utifrån resultatet framgår det hur metoden fungerade att utesluta och välja redevelopment till migrationsmetoden och parallella till övergångsmetod för att göra denna applikationsmigration mellan ett dokumenthanteringssystem till en webbapplikation. För att sedan i analysen beskriva hur varje delmål löstes med den valda migration och övergångsmetod. Där det framgår att det fördelaktiga med migrationsmetoden redevelopment är att utveckla denna webbapplikation, för att allt byggs upp från början. Sedan beskrivs hur övergångsmetoden parallella valdes eftersom det var den bästa övergångsmetoden för en migrationsövergång mellan dessa två system. Sammanfattningsvis har arbetet bidragit till en lyckad lösning för att applikationsmigrera mellan ett dokumenthanteringssystem och en webbapplikation. Där detta arbete går att använda som en riktlinje för andra sorters applikationsmigrationer mellan andra system.

## 6.2 Reflektioner

Från detta arbete har det varit intressant att från ett praktiskt arbete följa hur det fungerar att göra en applikationsmigration mellan ett äldre dokumenthanteringssystem till en ny webbapplikation. För att göra denna applikationsmigration är det migrationsmetoden och övergångsmetoden som är intressanta. Anledningen till att det var intressant med just migrationsmetoden var för att just detta problem var tvunget att lösas för detta praktiska arbete när detta gamla system skulle migreras till ett nyare. Sedan för att kunna välja den rätta migrationsmetoden att kunna lösa problemet och att använda den nyare tekniken gjorde att det var intressant att hitta och välja en lämplig migrationsmetod som kunde uppfylla dessa krav. Det var därför inte konstigt att migrationsmetoden redevelopment valdes eftersom denna migrationsmetod används för att skapa en applikation från grunden och tillåter därför implementation av verktyg eller nyare teknik på ett enklare sätt därför att allt kan anpassas från start av denna nya applikation. Detta gör att det blir smidigare och effektivare att använda denna migrationsmetod för att lösa applikationsmigrering av ett dokumenthanteringssystem. Anledningen till att de andra migreringsmetoderna inte

## Diskussion

valde är för att dessa inte passade lika bra som redevelopment eftersom de används huvudsakligen på existerande system vilket gör att det skulle exempelvis bli krångligare att implementera nyare teknik eller verktyg för dessa migrationsmetoder och var anledningen till att dessa valdes bort. Att sedan hitta den rätta övergångsmetoden för dessa system är också intressant för att hitta den bästa övergångsmetoden vilket är meningen för att göra denna övergång mellan två system på ett bra, enkelt och effektivt sätt. Att det blev den parallella övergångsmetoden är för att de andra två var för riskabla att använda för denna implementation vilket gjorde att den övergångsmetoden vilket var minst riskabel och mest effektiv valdes.

Genom att hitta den bästa migrationsmetoden för detta problem gör att andra med liknande problem kan utnyttja samma migrationsmetod. Detta för att slippa gå igenom samma steg och testningar vilket gjorts för att hitta en passande migrationsmetod till det speciella problemet. Med att hitta den bästa migrationsmetoden bidrar också till att detta arbete kan hjälpa till och ses som en mall för hur det går att göra för att applikationsmigrera just ett dokumenthanteringssystem till en nyare bättre webbapplikation för nutida bruk.

Från kunskapen av detta arbete är det meningen att förhoppningsvis kunna bidra till att andra fall med applikationsmigration och med liknande problem kan utnyttja den genomgående metoden vilket skapats i detta arbete eller utnyttja den skapade mallen för hur det går att göra för att göra denna applikationsmigration mellan dokumenthanteringssystem och webbapplikation. Det går också att använda de underliggande metoder som migrationsmetoder eller övergångsmetoder för att tillämpa på andra typer av applikationsmigration mellan system.

Från resultatkapitlet under införandet kan också användas för att tillämpas på andra implementationer. Detta för att kunna bygga ett system där det beskrivs att det är viktigt att ta reda på för och nackdelar med det gamla systemet vilket innebär att detta kan tas med för uppbyggandet av det nya systemet. Just detta kapitel beskriver inte bara hur införandet av en applikationsmigration utan också generellt hur en implementation för ett system kan skapas vilket är av intresse för att skapa andra implementationer. Det generella är huvudsakligen de fyra stegen vilket används för att visa stegvis hur ett system kan byggas upp.

En närmare kontroll visar att även kriterierna under delmål 2 från kapitel 3 är generella och kan användas för andra system vilka behöver byggas.

- Datadelning kan användas för att hitta informationsgrunderna för ett system vilket kan användas för skapandet av en ny databas.
- Flexibilitet gör att systemet kan verka hantera olika saker men i grund och botten gör samma saker på olika sätt.

Det intressanta är att genom detta arbete för hur applikationsmigrering mellan system kan göras och bidrar också till möjligheter för att hjälpa andra system vilka behöver applikationsmigreras. Det skulle här vara intressant att kunna utveckla en egen modell vilket skulle kunna användas till att ses som en utvecklingsmodell för organisationer vilka har behovet att göra en applikationsmigration. En idé är att denna modell skulle kunna likna den vilket ses i kapitel 2.4 där vattenfallsmodellen från informationssystem beskrivs. Att skapa en modell med olika faser gör att det skulle kunna gå att bygga upp och göra dessa migrationer lättare och effektivare, där modellen skulle byggas upp från detta arbete men delas in i olika faser för att enkelt stegvis göra en applikationsmigration.

### 6.3 Framtida arbete

Fortsätt att utveckla den nya webbapplikationen skulle innebära att mindre förändringar och modifieringar i första hand skall kunna utföras. Där dessa förändringar skulle vara att utöka funktionaliteten för denna nya webbapplikation och där funktionaliteten uppstått från önskemål vilka kommit under arbetets gång, men har fått prioriteras bort på grund av tidsbrist och inkompatibilitet. Efter att denna nya webbapplikation har körts och används av användare skulle det vara en idé att intervjua dessa användare. Detta för att kontrollera vilka nya behov som har uppstått och därefter kunna utveckla och uppdatera den designmässiga delen samt andra funktioner. Detta för att framhäva systemets egenskaper och på detta sätt göra systemet mer användarvänligt och lätt manövrerat. En annan ska vilket skulle vara av högsta intresse är att kunna lägga fokuseringen på effektivisering och utbyggnad mot nuvarande system för att på detta sätt kunna tillgodose mera behov för organisationen. Eftersom systemet nu är byggt med en bra grund och är anpassat för att kunna byggas ut, skulle en utbyggnad av systemet göras för att kunna stödja flera funktioner och önskemål.

En idé för framtida forskning är att gå igenom denna skapade mall för hur applikationsmigration kan göras till att fokusera mera på att förfinas och utveckla mallen för att bli bättre och ännu mera generell. Detta för att kunna användas mellan alla system vilka behöver hjälp med att applikationsmigreras. Denna mall skulle kunna realiseras genom att skapa en utvecklingsmodell för att på detta sätt kunna tillämpas för organisationer vilket behöver migrationshjälp.



## Referenser

- Bisbal, Jesús, Lawless, Deirdre, Wu, Bing & Grison, Jane (1999). *Legacy Information System Migration: A Brief Review of Problems, Solutions and Research Issues*. IEEE Software, Vol. 16: 5, ss. 103-111
- Deshpande, Yogesh, Murugesan, San, Ginige, Athula, Hansen, Steve, Schwabe, Daniel, Gaedke, Martin & White Bebo (2002). *Web Engineering*. Journal of Web Engineering, Vol. 1: 1, ss. 3-17
- Füricht, Reinhard, Prähofer, Herbert, Hofinger, Thomas & Altmann, Josef (2002). *A component-based application framework for manufacturing execution systems in C# and .NET*. Proceedings of the Fortieth International Conference on Tools Pacific: Objects for internet, mobile, and embedded applications; ss. 169-178
- Ginige, Athula & Murugesan, San (2001). *Web engineering: an introduction*. Multimedia IEEE, Vol. 8: 1, ss. 14-18
- Henriksson, Jonas & Strandberg, Hans (2003). *Utveckling av web services – en komparativ analys av utvecklingsmetoder*. Examensarbete. Göteborgs universitet.
- Hevner, Alan, March, Salvatore, Park, Jinsoo & Ram, Sudha (2004). *Design science in information systems research*. MIS Quarterly, Vol. 28: 1, ss. 75-105
- Johnston, Gary P & Bowen, David V (2005). *The benefits of electronic records management systems: A general review of published and some unpublished cases*. Records Management Journal, Vol. 15: 3, ss. 131-140
- Joshi, James, Aref, Walid, Ghafoor, Arif & Spafford, Eugene (2001). *Security models for web-based applications*. Vol. 44: 2, ss. 38-44
- Kranch, Douglas A (1998). *Preserving electronic documents*. International Conference on Digital Libraries, ss. 295-296
- McLeod, J (1996). *A review of 'Document management strategy report - the virtual opportunity'*. Records Management Journal, vol. 6: 1, ss. 64-4
- Meier, Johannes & Sprague, Ralph (1996). *Towards a better understanding of electronic document management*. Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences; 3-6 January 1996; Vol. 5, ss. 53-61
- Pigora, Mary & Thompson Eric (2008). *Rich and Thin: Migrating Simulations from Desktop to Web*. The Interservice/Industry Training, Simulation & Education Conference (IITSEC)
- Simon, Alan (1992). *Systems Migration – A Complete Reference*. Van Nostrand Reinhold.
- Strandberg, Hans & Henriksson, Jonas (2003). *Utveckling av web services – en komparativ analys av utvecklingsmetoder*. Examensarbete. Göteborg University.
- Sørensen, Carl-Fredrik, Wang, Alf Inge & Hoftun, Øystein (2003). *Experience Paper: Migration of a web-based system to a mobile work environment*. Proceedings of Applied Informatics; 10-13 February 2003; Innsbruck: Austria

Zepeda, J & Chapa, S (2007). *From desktop applications towards ajax web applications*. Electrical and Electronics Engineering, 2007, ICEEE 2007, 4<sup>th</sup> international Conference On; 5-7 Sept 2007; ss. 193-196

Zou, Ying & Kontogiannis, Kostas A (2000). *Web-based Legacy System Migration and Integration*. In Proceedings of CASCON'00; November 2000; Toronto ON, ss. 262-272