En studie i marknadsdriven systemgeneralisering

A Case Study on Market-Driven System Generalisation

Annie Bengtsson

Department of Communication Systems
Lund Institute of Technology, Lund University
Box 118
S-221 00 Lund
Sweden

AU-System AB
Ideon
S-223 70 Lund

LUTEDX (TETS-5425)/1-82/(2001)&local9

© Annie Bengtsson
Lund, 2001

Sammanfattning

Programvarusystem har på senare tid ökat kraftigt i komplexitet och blivit allt dyrare att utveckla vilket i sin tur medfört att det blivit svårare att få kunder att ta investeringen. Ett sätt som föreslås för att minska kostnaderna för utveckling och därmed investeringsbehovet är att generalisera ett befintligt programvarusystem. Genom generalisering blir systemet anpassat för en bredare kundkrets och kostnaden per kund minskar.

Syftet med detta examensarbetet är att finna en generaliseringsmetod för ett befintligt programvarusystem (ursprungligen byggt åt en specifik kund) med avsikt att använda det till en bredare marknad. Dessutom skall Generaliseringsmetoden verifieras genom en fallstudie av ett kundvårdssystem inom mobila telekommunikationsbranschen.

Genom att kartlägga utvecklingsprocessen har författaren funnit att delprocessen kravhantering kan förbättras. Därefter har en egen Generaliseringsmetod skapats som utifrån befintlig kund, befintligt programvarusystem och tänkt marknad resulterar i en generaliserad produkt. Generaliseringsmetoden kombinerar marknadsanalys och segmentering med en fritt vald utvecklingsprocess med tillägg av en efterprocess Kravurvalsprocess KUP till kravhanteringen. Kravurvalsprocessen beräknar de lönsammaste kraven att realisera utifrån befintligt system och kraven från befintliga eller potentiella kunder.

I fallstudien identifieras Europa och Australien som lämpligaste marknaderna för ett generaliserat kundvårdssystem. I Europa valdes Frankrike, Italien, Polen och Spanien som de intressantaste länderna att bearbeta. Under fallstudien inhämtades kundkrav som filtrerades genom en praktisk tillämpning av Kravurvalsprocessen KUP.

Generaliseringsmetoden har verifierats genom en fallstudie och torde därför kunna användas på alla typer av programvarusystem för att minska utvecklingstiden och därmed kostnaden.

Abstract

A Case Study on Market-Driven System Generalisation

Software systems has lately increased a lot in complexity and has become ever more expensive to develop, which implies difficulties in finding customers willing to invest. One way to decrease the cost of development and thereby also the need for investments is to generalise an existing software system. By reusing functions from a system that has been built earlier, it is possible to save time and money. By generalising the system it is adopted for a wider market and the cost per customer may be decreased.

The purpose of this Master Thesis is to find a method for generalisation of software systems (originally build to fit one customer) with the aim to offer it to a wider market. The method for generalisation is verified through a case study on a customer relationship management system within the mobile telecommunications area.

By surveying the development process the writer has found that the sub process Requirements Engineering can be improved. Then the writer has created a method of generalisation which on the basis of an existing customer, existing software system and a target market results in a generalised product. The method of generalisation combines market analysis and segmentation with a development process free of choice with the ad on of a post process to the Requirements Engineering – Requirement Selection Process (KUP). The Requirement Selection Process calculates the requirements most profitable to fulfil on the basis of an existing system an the requirements from existing or potential customers.

During the case study Europe and Australia is identified as the most promising markets for a generalised Customer Relationship Management System. Within Europe France, Italy, Poland and Spain was identified as the most promising countries to canvas. As part of the case study customer requirements were collected and filtered by the Requirement Selection Process (KUP) put into practise.

The method for generalisation has been verified by a case study and presumably the method may work on similar systems within other fields and thereby reducing the time for development and also the cost.

Förord

”En studie i marknadsdriven systemgeneralisering” är titeln på denna rapport, som bland annat beskriver hur man kan generalisera ett programvarusystem byggt åt en specifik kund, till en bredare marknad och den därav följande inverkan på kravhantering. De programvarusystem som berörs i denna rapport är komplexa och har en tekniskt avancerad struktur. Att utveckla sådana system kräver inte bara en avancerad teknik utan också mycket kompetens, kunskap och arbete. En generalisering av dessa system kan leda till minskad arbetsbörda samt tidsbesparing och därmed ökad kostnadseffektivitet för både kunderna och utvecklaren av systemet.

Rapporten är ett resultat av ett examensarbete som har genomförts på uppdrag av institutionen Telekommunikationssystem vid Lunds Tekniska Högskola i samarbete med AU-System AB. Författaren till rapporten är Annie Bengtsson studerande vid Teknisk Fysik vid Lunds Tekniska Högskola. Handledarna till examensarbetet är Björn Regnell vid LTH och Lars Lundgren vid AU-System AB.

Rapporten är baserad på fallstudier av ett kundvårdssystem, litteraturstudier och ämnesfördjupning samt intervjuer och enkätundersökningar.

Jag vill framföra ett stort tack till alla de personer som ägnat tid och energi åt mitt examensarbete samt AU-System och institutionen Telekommunikationssystem vid Lunds Tekniska Högskola för de resurser som ställts till mitt förfogande. Ett särskilt tack för all hjälp, stöd och uppmuntran riktar jag till min familj: min mor Danuta Bengtsson, Lars Bengtsson, Björn Persson, Hans och Christiane Persson. Slutligen ett stort tack till alla mina vänner som ställt upp för mig.

Annie Bengtsson
Lund, februari 2001

Trevlig läsning!

Innehållsförteckning

Sammanfattning **3**

Abstract **5**

Förord **7**

1. Introduktion **13**

1.1. Syfte 14
1.2. Metod 14
1.3. Avgränsningar 14
1.4. Rapportens struktur 15

2. Programvaruutveckling - en översikt **17**

2.1. Introduktion 17
2.2. Problemlösning 18
2.3. Utvecklingsprocess 20
2.4. Kravhantering 20
2.4.1. Krav *21*
2.4.2. Klassificering av krav *21*
2.4.3. Kravhanteringsprocessen *21*
2.4.4. Intressenter *23*
2.4.5. Kostnadsuppskattning *23*
2.4.6. Dokumentation *24*
2.5. Programvaruarkitektur 24
2.6. Utvecklingsmodell 24
2.6.1. Vattenfallsmodellen *25*
2.6.2. V modellen *26*
2.6.3. Prototypmodellen *26*
2.6.4. Win-Win Spiralmodellen *27*
2.7. Slutord 28

3. Marknadsanalys och segmentering – en översikt **29**

3.1. Produkt 29
3.2. Marknadsföring och marknadsstrategi 29
3.2.1. Internationell marknadsföring *31*
3.2.2. Marknadsföring av tjänst *32*
3.3. Marknadstyper 33
3.4. Marknadssegmentering 33
3.4.1. Målinriktad marknadsföring *33*
3.5. Marknadsföringens informationssystem 34
3.5.1. Marknadsundersökning *34*
3.6. Slutord 35

4. Generalisering av programvarusystem **37**

4.1. Introduktion 37
4.2. Grafisk presentation av Generaliseringsmetoden 38
4.3. Principen med Generaliseringsmetoden 38
4.4. Generaliseringsmetodens beståndsdelar 39
4.4.1. Marknadsanalys- och segmenteringsmetoden *39*
4.4.2. Programvaruutvecklingsmetoden *40*

4.5. Slutord 45

5. Tillämpning av Generaliseringsmetoden 47

5.1. Inledning 47
5.2. Kundvårdssystem – översikt 47
5.3. Segmentering och analys av marknaden 47
5.3.1. Hur hitta ny marknad? 48
5.3.2. Branschutveckling - Vilka branscher? 49
5.3.3. Landsutveckling - Vilka länder? 50
5.3.4. Vilka företag? 51
5.4. Programvaruutveckling tillämpad på kundvårdssystemet KVS 52
5.4.1. Kravhantering 52
5.5. Kravurvalsprocess KUP 54
5.6. Resultat av fallstudien 60

6. Slutsatser 63

6.1. Rekommendationer till fortsatt utveckling av Generaliseringsmetoden 64

Referenser 67

Referenslista trycksaker 67
Referenslista Internet 68

Figur- och tabellförteckning 70

Terminologi 73

A Appendix Enkät ”Krav från kunden” 77

1. Introduktion

Den snabba utvecklingen inom IT-industrin medför ett ökat behov av kraftfulla verktyg för hantering av stora mängder inkommande information.

Den ökade komplexiteten hos ett programvarusystem medför ökade utvecklingskostnader. Det finns flera möjliga tillvägagångssätt för att reducera både kostnader för utveckling och time-to-market[1] av ett programvarusystem. Kostnadseffektiva metoder som näringslivet använder sig av idag är att t.ex. i ett system skapa nya funktioner som ersätter flera av de gamla och därmed reducerar mängden kod som skall skrivas, att köpa in färdigtutvecklade delsystem eller att tillämpa den nyaste tekniken t.ex. ett effektivare programmeringsspråk.

En idé som är värd att pröva är att generalisera ett befintligt programvarusystem för att smidigt kunna anpassa och sälja systemet till andra kunder. Generaliseringen kan effektivisera utvecklingen av programvarusystem och därmed öka företagets produktivitet. Genom generalisering kan ett system även anpassas för en bredare kundkrets och marknaden ökar därmed.

Generaliseringen kan byggas på att återanvända delsystem av ett tidigare byggt programvarusystem. Genom att jämföra kundens önskemål och krav med funktioner från ett befintligt programvarusystem som uppfyller kundens krav kan dessa funktioner återvinnas vid nykonstruktion av ett liknande system. Detta kan leda till ökad tids- och kostnadseffektivitet för så väl användaren av det nya systemet som för den som bygger och utvecklar det. Genom att förbättra en del av systemet och låta andra delar förbli oförändrade borde det vara möjligt att generalisera ett befintligt programvarusystem utan att märkbart påverka systemets prestanda.

Dessutom kan man vid en återanvändning av ett befintligt system ta lärdom av och utnyttja kunskaper från tidigare erfarenheter och därmed få en bättre förståelse för kundens krav och behov. Arbetet med de nya systemen underlättas eftersom man får bättre insikt i de problem som kan uppkomma i samband med systemutveckling. Det generaliserade systemets tillförlitlighet och kvalitet ökar efterhand som fler kunder använder det.

Företag som utvecklar programvarusystem skulle med hjälp av generalisering kunna antas bli mer attraktiva på marknaden än sina konkurrenter genom kortare utvecklingstider, lägre utvecklingskostnader samt bättre och djupare förståelse för kundens behov.

Hur skulle det kunna gå till? En intressant fråga att ställa sig blir:

Finns det en metod som kan användas för att anpassa ett befintligt system så att det passar fler kunder?

[1] Hur lång tid det tar innan en produkt eller tjänst kan lanseras på marknaden.

1.1. Syfte

Avsikten med detta examensarbetet är att finna en Generaliseringsmetod för att utifrån ett programvarusystem, ursprungligen byggt åt en specifik kund, kunna generalisera detta system till en bredare marknad.

Dessutom skall Generaliseringsmetoden verifieras genom en fallstudie av ett kundvårdssystem inom mobila telekommunikationsbranschen.

1.2. Metod

Genom litteraturstudier har författaren fördjupat sig i ämnena programvaruteknik[2], projekthantering, marknadsföring, kravhantering, kundvård[3] mm.. Utifrån denna kunskap har författaren skapat den teoretiska Generaliseringsmetoden.

För att verifiera Generaliseringsmetoden har en fallstudie på ett programvarusystem utförts. Programvarusystemet är ett kundvårdssystem[4] för mobila telekommunikationsbranschen. Under fallstudien har författaren:

- Skapat förståelse för systemets funktioner genom att läsa systembeskrivningar och intervjuat utvecklare.
- Inhämtat tänkbara krav på funktioner hos ett kundvårdssystem från potentiella kunder. Inhämtningen har gjorts i form av en enkätundersökning.
- Inhämtat kunskap om kostnad per nyutvecklad funktion. Kunskapen har inhämtats genom enkätundersökning hos utvecklare av befintligt kundvårdssystem.
- Skapat en beräkningsmodell för att identifiera de lönsammaste funktionerna att nyutveckla.

Generaliseringsmetoden kan anses verifierad om fallstudien leder fram till ett antal generaliserade funktioner.

1.3. Avgränsningar

För att skapa ett bättre djup i rapporten blev det nödvändigt att göra avgränsningar i innehållet. Avgränsningar är gjorda enligt följande:

- Generaliseringen av programvarusystem har enbart testats på ett kundvårdssystem. Att försöka göra en generalisering på fler system skulle kräva betydligt mer tid än vad som är avsett för ett examensarbete.

- Avgränsningar i samband med fallstudien finns beskrivna i Kapitel 5.

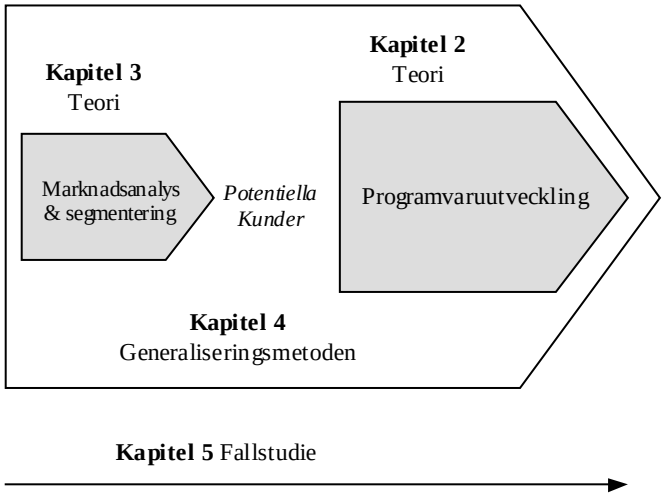
[2] Verktyg och metoder som studeras och sedan används vid konstruktion av program och system
[3] Hantering av kundrelationer, *se Appendix A.*
[4] Ett programvarusystem för hantering av kundrelationer, *se Appendix A.*

1.4. Rapportens struktur

Rapporten inleds med två teorikapitel, kapitel 2 och 3, vilka ger grunder till Generaliseringsmetoden i kapitel 4. Därefter prövas Generaliseringsmetoden i en fallstudie i kapitel 5. Rapporten avslutas med analys och slutsatser i kapitel 6. Nedan följer kort beskrivning av kommande kapitel.

2. **Programvaruutveckling – en översikt;** Kapitel två är en introduktion till ämnet ”Programvaruteknik” och redogör steg för steg för hur utveckling av programvara för stora system går till, samt ger en beskrivning av ett antal modeller för utvecklingsprocesser.
3. **Marknadsanalys och segmentering – en översikt;** I detta kapitel behandlas teoretiskt marknadssegmentering och marknadspositionering samt analyseras hur en marknadsundersökning går till.
4. **Generalisering av programvarusystem;** I detta kapitel presenteras författarens Generaliseringsmetod. Metoden beskriver hur delar av ett befintligt programvarusystem kan återanvändas när ett nytt liknande system skall byggas åt andra kunder.
5. **Tillämpning av Generaliseringsmetoden;** Kapitlet behandlar en fallstudie av Generaliseringsmetoden på ett kundvårdssystem som skall säljas internationellt. Fallstudien innehåller både marknadsundersökning och val av krav att realisera.
6. **Slutsatser;** I detta kapitel diskuteras resultatet av examensarbetet samt slutsatser dras. Kapitlet innehåller dessutom en utvärdering av rapporten samt lämnar upplysningar som kan vara till praktisk nytta för en eventuell vidare forskning inom området.

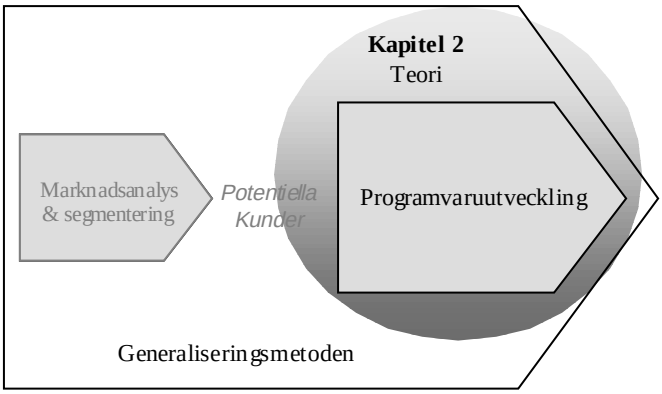
Generaliseringsmetoden presenteras grafiskt i Figur 1. Figuren visar även kapitelindelning som beskrivs utifrån Generaliseringsmetoden.



• Figur 1 Generaliseringsmetoden - en översikt

2. Programvaruutveckling - en översikt

I detta kapitel behandlas grunderna till programvaruteknik samt diskuteras kravhantering och kravspecifikation. Programvaruteknik används som en av grundpelarna till uppbyggnad av Generaliseringsmetoden beskriven i kapitel 4, enligt Figur 2.



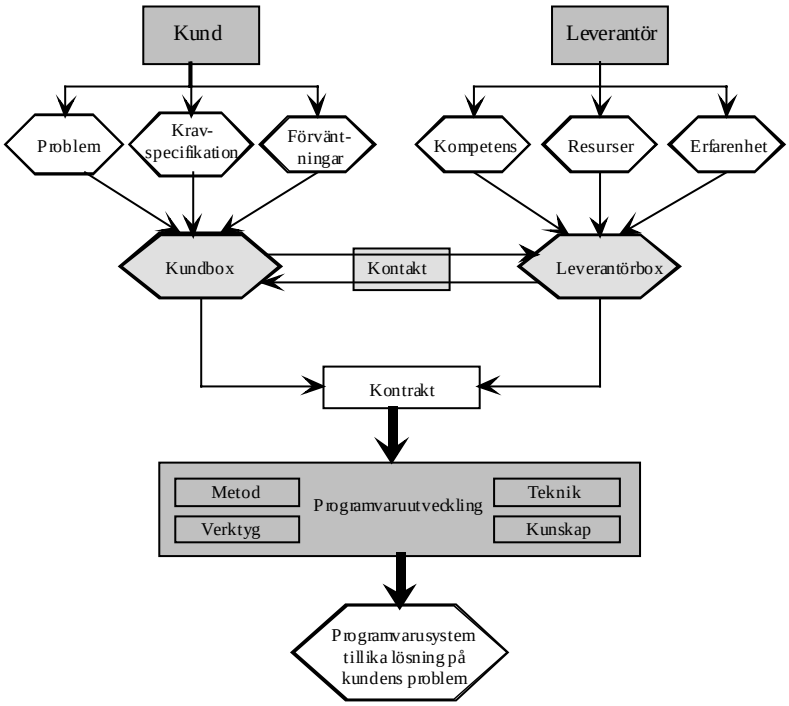
• Figur 2 Generaliseringsmetoden - en översikt

2.1. Introduktion

Programvarusystem har successivt utvecklats och byggts ut i takt med datorindustrins tillväxt. I början var all programvara enkel och okomplicerad och kunde utvecklas av en person. Sedan datorer antagit en central plats inom industrin har många företag blivit beroende av allt större och mer komplexa programvarusystem. Ett programvarusystem är en sammansättning av mindre program som arbetar ihop. Ju större ett system är desto fler mantimmar krävs det för att bygga systemet, vilket leder till att fler personer involveras i utveckling av det.

Komplexa programvarusystem används inom både den interna och den externa verksamheten som exempelvis ekonomi, produktionsstyrning, processövervakning, kundregister, produktutveckling, analyser, kundkontakt, försäljning, marknadsföring, mm. Det finns alltså många användningsområden för programvarusystem och behovet av dessa ökar ständigt.

Utveckling av ett programvarusystem bygger på en interaktion mellan kunden och användaren av systemet. Det är viktigt att ett programvarusystem håller hög kvalitet och är anpassat efter verksamhetens behov. För att uppfylla dessa krav hos en programvara måste utvecklarna och kunderna ha varit i kontakt med varandra och enas om vad som förväntas av systemet. I Figur 3 visas exempel på hur ett flödesdiagram över interaktionen mellan kunden och leverantören kan se ut. Detta exempel är skapat av författaren till rapporten med hjälp av kunskaper som inhämtats från [Lekvall & Wahlin 1993], [Ncube 2000], [Royce 1970], [Eklund & Fernlund 1998].



• Figur 3 Interaktion mellan kunden och leverantören av programvarusystem

Kunden har ett eller flera problem som skall lösas. Kunden har höga förväntningar på lösningen till problemet och ställer upp vissa krav. I kundboxen finns en sammanslagning av kundens krav och önskemål. Leverantören vill lösa kundens problem med hjälp av sin kompetens, sina resurser och sin erfarenhet. I leverantörsboxen finns de egenskaper som leverantören kan erbjuda kunden. Genom överrenskommelse mellan kunden och leverantören kan uppgiften som skall lösas, konkretiseras, priset bestämmas och ett kontrakt upprättas. Detta leder till programvaruutveckling och slutligen till en färdig produkt.

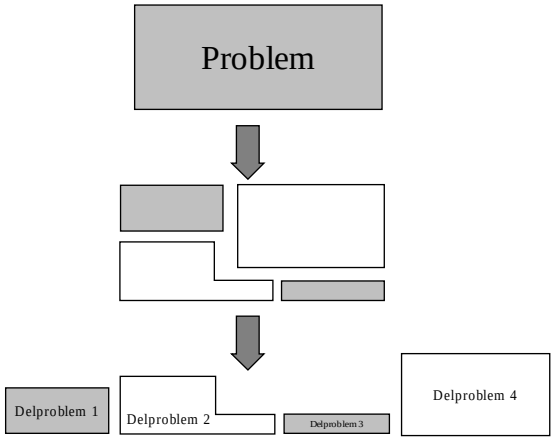
Programvaruteknik är en teknik för hur man kan gå tillväga vid utveckling av stora programvarusystem. Ju större och mer komplext ett programvarusystem är desto fler rader programmeringskod måste skrivas vilket kan leda till stor risk för fel.

Målet med programvaruutveckling är att skapa en tillförlitlig slutprodukt som kunden är nöjd med. För att uppnå detta mål måste arbetsmetoder konkretiseras och riktlinjer för hur man undviker programvarufel antas. Ju färre programvarufel desto högre kvalitet och tillförlitlighet på systemet. Programvarutekniken lägger upp metodiken för hur man kan gå tillväga för att på ett bra sätt vidareutveckla ett befintligt programvarusystem eller bygga ett nytt.

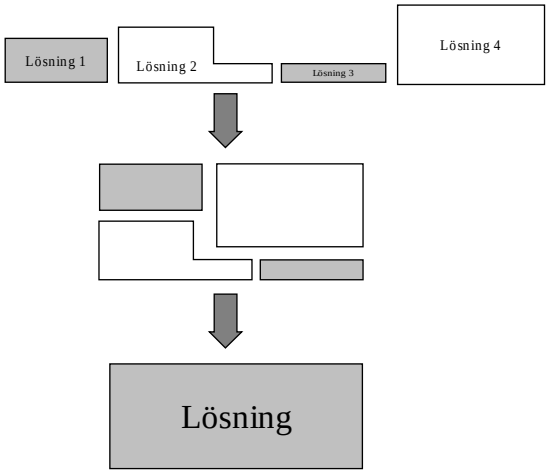
2.2. Problemlösning

Syftet med programvaruteknik är att lösa problem med hjälp av datorteknik. Tillvägagångssättet vid problemlösning kan beskrivas som att finna vägen mellan ett problem och lösningen till problemet. Filosofin bakom program inom programvaruutveckling beskrivs generellt i [Pfleeger & Shari 1998] med två processer ”The process of analysis” och ”The process of synthesis”. ”The process of analysis” är en Top-down strategi vilket innebär att problemet delas upp i mindre bitar, delproblem.

”The process of synthesis” är en omvänd process dvs. stödjer Top-up strategi genom att lösa varje delproblem för sig och sedan sätta samman dessa till en enda lösning, se Figur 4 och Figur 5.



• Figur 4 The process of analysis



• Figur 5 The process of synthesis

En generell metod för problemlösning som bygger på filosofin ovan har framtagits av Polyas [Eklund & Fernlund 1998]. Problemlösning med hjälp av programvara måste utföras metodiskt och noggrant steg för steg. Detta kan åstadkommas genom att följa de fyra faserna i Polyas metod:

1. Definitionsfasen: att förstå problemet
2. Fasen för design: att skapa en plan beskriven i designspecifikationen för lösning av problemet
3. Programmeringsfasen: att implementera alla lösningar, dvs. att utföra planen
4. Testfasen: att utvärdera resultatet

Polyas problemlösningsmetod beskriver arbetsgången för hur problem löses av mindre program. När det gäller komplexa programvarusystem[5] som oftast består av flera program som måste integreras med varandra krävs en mer specificerad arbetsgång.

2.3. Utvecklingsprocess

När programvarusystem blev så komplexa att de inte längre kunde hanteras av en utvecklare och den gamla metoden för utveckling av ett program (”skriv”, ”kör”, ”kompilera”, ”hitta buggar”) inte längre kunde tillämpas hittades nya vägar att tillgå.

Utvecklingsprocessen är samlingsbegreppet på de aktiviteter som utförs under utveckling av ett programvarusystem. Den kan indelas i flera faser där varje fas består av en eller flera aktiviteter. I [Pfleeger & Shari 1998] presenteras en grundläggande metodik för utvecklingsprocessen med hjälp av fasindelning enligt följande:

Kravhantering: att på ett kontrollerat sätt samla in och specificera kundens krav och önskemål.

Systemdesign: att med utgångspunkt från kravspecifikationen generera systemets struktur för att kunna beskriva vad systemet skall göra och inom vilka ramar (användargränssnitt).

Programdesign: att strukturera problemet till en nivå där fristående programdelar, moduler, kan avskiljas. Därefter specificeras ett externt snitt för dessa program, vad de skall beräkna eller utföra, vilka parametrar som skall tas med, specificera programmeringsparadigm[6].

Kodning (program implementering): att utifrån programdesignen för varje modul välja algoritm, intern datastruktur, logisk struktur och generera programmeringskoden.

Enhetstestning: att testa varje programmodul för sig.

Integrationstestning: att verifiera gränssnittet mellan modulerna genom att kontrollera om produkten uppfyller dess designspecifikation.

Systemtestning: att validera eller verifiera att systemet uppfyller de uppställda målsättningarna. Validering görs genom granskning och acceptanstest[7] av kunden medan verifiering görs genom att kontrollera att systemet uppfyller dess specifikation i systemdesignen.

Underhåll: Att rätta till fel som upptäckts efter leveransen av systemet samt att ändra delar i systemet efter nya krav och behov från kunden.

Utvecklingsprocessen resulterar i ett programvarusystem som tillgodoser kundens krav.

2.4. Kravhantering

Kravhantering definieras i [Regnell, Wohlin 1999] som ”metoder och tekniker för att på ett kontrollerat sätt samla in, specificera och dokumentera krav samt att säkerställa att de överensstämmer med uppdragsgivarens behov och önskemål”. Kravhantering fokuserar på *vad* ett programvarusystem skall göra. Svårigheter med kravhantering ligger i att hitta en för kund och utvecklare gemensam tolkning av kraven.

[5] Ett större system som bildas av ett antal mindre program som arbetar ihop.
[6] Beskriver olika sätt att programmera på: funktionell, objektorienterad, procedurell, logisk. Se Appendix A
[7] Validering av systemet gentemot kundens krav.

2.4.1. Krav

Inom programvaruteknik definieras ordet ”krav” som kundens krav och önskemål på produkten samt kundens behov som skall tillgodoses. Nedan följer de typer av krav som finns enligt [Ncube 2000].

Fysisk miljö, tex. lokal, miljöfaktorer som har någon inverkan på programutvecklingen.

Interface eller gränssnitt, tex. föreskrivet dataformat.

Användar och mänskliga faktorer, tex. vem och vilken typ av person skall använda systemet.

Funktionalitet, tex. vad systemet skall göra och när.

Dokumentation, tex. vilket format skall dokumentationen ha.

Data, tex. storlek på den inkommande och den utgående datamängden.

Resurser, tex. material som skall användas, personal som skall arbeta med projektet.

Säkerhet, tex. vem kommer att ha access till systemet, har systemet någon säkerhetskopia.

Kvalitetsgaranti, tex. systemets tillförlitlighet, tillgänglighet, underhåll.

2.4.2. Klassificering av krav

Klassificering av kraven på ett programvarusystem underlättar beskrivningen av dessa krav [Pfleeger & Shari 1998]. Kraven beskriver ett systems beteende och respons på det som kommer in i systemet. Klassificering av krav kan göras genom att dela in kraven i två distinkta klasser: funktionella krav och icke-funktionella dvs. egenskapskrav.

De funktionella kraven beskriver en interaktion mellan systemet och dess omgivning samt de tjänster som systemet skall tillhandahålla. Funktionella krav beskriver alltså sambandet mellan indata och korresponderande utdata. Exempel på detta är att om användaren av systemet vill ta ut pengar från en bankautomat genom att trycka in sin kod och önskat belopp skall den begärda summan komma ut. Systemet skall svara på användarens anrop.

Egenskapskraven beskriver villkor och restriktioner som de funktionella kraven måste uppfyllas. Dessa krav beskriver vad som begränsar lösningen av problemet som ligger till grund för att systemets skulle byggas. Exempel på icke-funktionella krav kan vara tillförlitlighet, användbarhet, effektivitet, kapacitet.

2.4.3. Kravhanteringsprocessen

För att tillgodose de krav som ställs på ett programvarusystem att leverera en korrekt utformad produkt måste riktlinjer för tillvägagångssätt bestämmas och utarbetas. Kravhanteringsprocessen innebär att specificera och hantera dessa kundkrav. Denna process kan ses utifrån två olika perspektiv, kundens och leverantörens.

Utifrån kundens perspektiv är det väsentligt att tillgodose befintliga behov samt att vidga sina vyer på vad produkten kan göra genom att med hjälp av leverantörens kunskaper och erfarenheter finna nya innovativa lösningar.

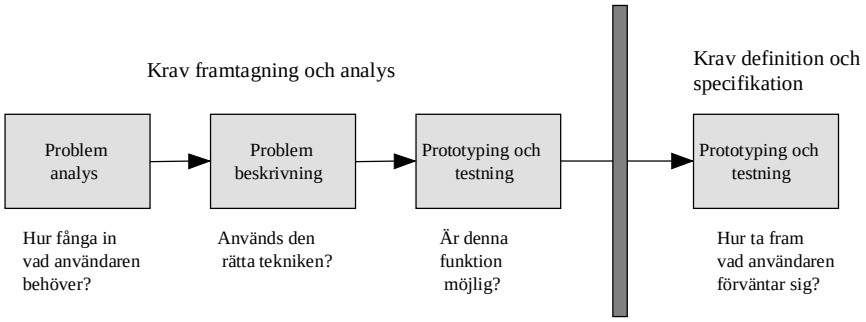
Utifrån leverantörens perspektiv är det väsentligt att på ett korrekt sätt definiera kundens krav samt att skapa nya behov hos kunden genom att erbjuda nya tekniska lösningar. I [Eklund & Fernlund 1998] har tagits fram riktlinjer för hur leverantören kan definiera kraven. En sammanfattning av dessa riktlinjer följer nedan:

- Identifiera kunden: ta reda på vem som egentligen kommer att använda systemet.
- Analysera befintligt system: om det finns ett befintligt system som skall förnyas eller ersättas, beskriv detta systemet tex. grafiskt och kontrollera att kunden tolkar systemet på samma sätt.
- Intervjua användare: att sätta sig in i användarens situation och försöka bilda sig en fullständig bild av den.
- Identifiera och samla in kraven från kunden.
- Kontrollera omgivningens krav: att kontrollera om omgivningen har några speciella krav på systemet som för kunden är så självklara att de utelämnats från kravspecifikationen.
- Göra en hierarkisk problembeskrivning: att beskriva systemet på detaljerade nivåer och tala om vad som skall lösas och vad som inte ingår i problemlösningen.
- Prioritera och välja bland kraven: Tre nivåer rekommenderas:

1. Krav som absolut måste tillgodoses.
2. Krav som är högst önskvärda men inte nödvändiga.
3. Krav som är möjliga men kan elimineras.

- Dokumentera kraven: att beskriva kraven på ett för kunden lätt begripligt språk.
- Validera kraven: kontrollera att det är rätt produkt som utvecklats.
- Låta kunden ta ansvar för och vara delaktig i kravspecifikationen: att låta kunden delta under hela utvecklingsprocessen genom att han stegvis godkänner kraven i kravspecifikationen.

När kundkraven identifierats måste utvecklaren besluta vilka av dessa krav som skall ligga till grund för programvaruutveckling. Beslutet grundas på hur problemet skall lösas, vilken är den rätta tekniken för lösning av problemet, om kunden är villig att betala kostnaden samt är det möjligt att genomföra. Modellen i Figur 6 belyser en krav beslutsprocess enligt [Pfleeger & Shari 1998].



• Figur 6 Framtagning och analys av krav

2.4.4. Intressenter

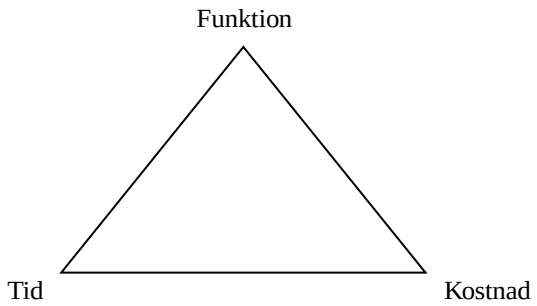
Följande intressenter kan vara involverade i kravhanteringen:

- Kunden: den som beställer och betalar för utvecklingen av systemet
- Kundens kunder: kunder till kunden enligt definition ovan.
- Användaren: den som använder systemet.
- Utvecklaren: den som utformar och konstruerar systemet.
- Leverantören: den som tillverkar och säljer systemet.
- Marknadsföraren: den som marknadsför systemet.

Dessa intressenter kan ha olika tolkningar av de krav som ställs på systemet och olika uppfattningar om vad som är möjligt att göra. Därför är det viktigt att de som är inblandade kommunicerar med varandra och tillsammans utformar för kravspecifikationen.

2.4.5. Kostnadsuppskattning

Kostnadsuppskattning är en viktig del i kravhanteringen. Kostnaden för ett programvarusystem relateras till den tiden som projektet kräver fram till leveransen av en färdig produkt. Den tillåtna kostnaden kan vara en begränsande faktor för utvecklingen av ett system eftersom den styr hur lång tid utvecklingsprocessen får ta. I Figur 7 visas sambanden mellan tid, funktion och kostnad i Åtagandetriangeln enligt [Eklund & Fernlund 1998].



• Figur 7 Åtagandetriangel

Grundprincipen i Åtagandetriangeln bygger på att ett av de tre hörnen i triangeln måste hållas dynamiska under utvecklingsprocessen. Det är mycket svårt att korrekt bedöma tiden för att lösa ett nytt problem eftersom man vet inte vilka hinder man möter på vägen.

Följande faktorer har stor inverkan på projektets tidsuppskattning och därmed kostnaden:

- Gruppen. Hur fungerar gruppen tillsammans? Har gruppmedlemmarna samma tolkning av problemen, etc.?
- Arbetsplatsen. Var finns arbetsplatserna för samtliga involverade i projektet?
- Erfarenhet och kunskapsnivån hos gruppen. Har gruppen erfarenhet av liknande projekt? Besitter gruppen den nödvändiga kunskapen, etc.?
- Tidspressen. Är det gott eller ont om tid?

- Krav och funktionalitet. Är kraven på programvarans funktionalitet fastställda eller kan de ändras vid behov?
- Storleken på projektet. Hur stort är projektet i mantimmar räknat?

2.4.6. Dokumentation

Dokumenthantering kan beskrivas från två synvinklar: dels utifrån kundens perspektiv och dels utifrån utvecklarens perspektiv. När ett dokument skrivs utifrån kundens perspektiv måste det vara skrivet med lättbegripliga termer och på en nivå som kunden kan förstå.

Utifrån utvecklarens perspektiv måste ett dokument skrivas på ett sätt som gör att utvecklaren får en exakt och korrekt bild av vad som krävs av programmet.

Uppläggning av kravspecifikation rekommenderas i [Eklund & Fernlund 1998] och [Regnell, Wohlin 1999] enligt följande punkter:

1. Introduktion och bakgrund.
2. Översiktlig beskrivning av programvaran.
3. Systemkrav.
   - Funktionella krav.
   - Icke funktionella krav.
4. Krav på användargränssnitt.
5. Projektkrav
   - Leveransvillkor
   - Utvecklingsmiljö
   - Testkrav
   - mm..
6. Terminologi

2.5. Programvaruarkitektur

Programvaruarkitektur definieras i [Kazman *m.fl.* 1999] som ” The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them”.

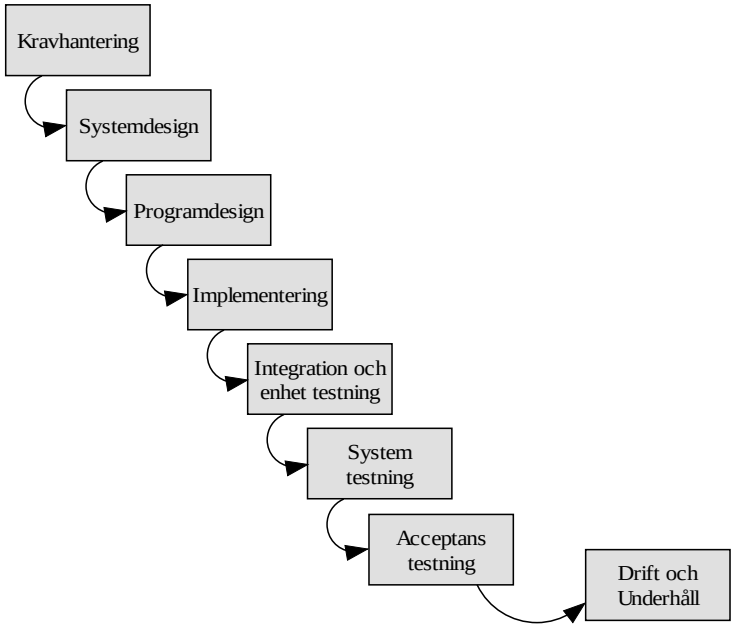
En programvaruarkitektur definierar komponenter såsom moduler, objekt, processer, undersystem, kompileringsenheter m.fl. samt de relevanta relationerna mellan dessa komponenter. Relationerna kan vara ”beroende relationer” samt av typen ”anrop”, ”skicka data till”, ”synkronisera med”, ”används som” m.fl.. I ett tidigt stadium av designprocessen måste vissa viktiga beslut tas. Dessa beslut är nödvändiga för att en grupp människor skall kunna samarbeta med varandra när de bygger ett programvarusystem. En programvaruarkitektur är ett resultat av dessa beslut.

2.6. Utvecklingsmodell

En utvecklingsmodell är en processmodell som beskriver ett processförlopp för att på ett organiserat sätt utveckla en programvara. ”En process är det som sker när ett arbete utförs” [Regnell, Wohlin 99]. Det finns olika processmodeller att tillämpa vid utveckling av ett programvarusystem. Valet av utvecklingsmodellen beror på flera faktorer som typ och storlek av systemet som skall utvecklas, behov av en prototyp, tid, kostnad mm. I följande avsnitt presenteras ett urval av utvecklingsmodeller för att ge en översikt.

2.6.1. **Vattenfallsmodellen**

Den mest klassiska och också en av de första utvecklingsmodellerna för programvarusystem är Vattenfallsmodellen i Figur 8 utvecklad av Royce [Royce 1970]. Vattenfallsmodellen beskriver en programvaras livscykel. Med livscykel menas det ”liv” som ett programvaruprodukt har från att produkten bara är ett begrepp till att den implementeras, levereras, används och underhålls. Vattenfallsmodellen kännetecknas av att en aktivitet måste vara färdig innan nästa aktivitet kan påbörjas. Denna utvecklingsmodell fungerar som ett vattenfall. Man går igenom aktiviteterna en efter en utan att återvända.



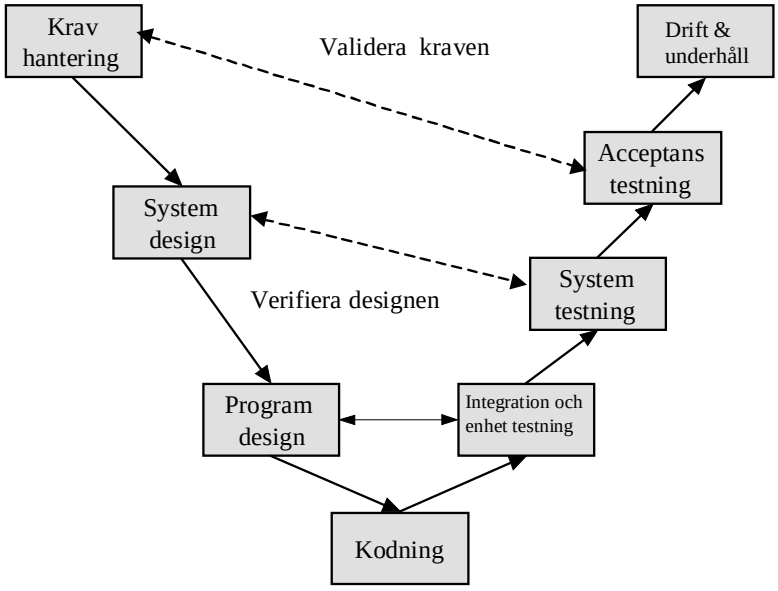
• Figur 8 Vattenfallsmodellen

Vattenfallsmodellen är från början utvecklad åt hårdvaruindustrin där produktionsförloppet bygger på utveckling av en produkt och därefter reproducering av denna produkt. När ett komplext programvarusystem utvecklas existerar relationssamband mellan de flesta komponenter och aktiviteter i modellen. Dessa aktiviteter följer inte någon exakt kronologisk ordning under processutvecklingen. Detta grundar sig bl.a. i att för många problem som skall lösas finns det ingen föreskriven lösning. Därför är det svårt att utveckla ett system utan att kunna göra ändringar när behov därav uppkommit i efterhand. En involvering av kunden under processens gång för att stämma av alla krav som ställs på systemet och minska risken för missförstånd mellan kunden och utvecklaren är inte heller möjlig i Vattenfallsmodellen. Modellen stödjer inte återanvändning av programdelar eller prototyputveckling.

Å andra sidan kan Vattenfallsmodellen hjälpa utvecklare av ett programvarusystem att förstå vad som behöver göras samt vilka parametrar, aktiviteter och uppgifter som är väsentliga vid systemutveckling. Modellen ger en bra och överskådlig bild av processförloppet. Vattenfallsmodellen är kostnadseffektiv vid utveckling av enstaka system men ej för grupper med många system.

2.6.2. **V modellen**

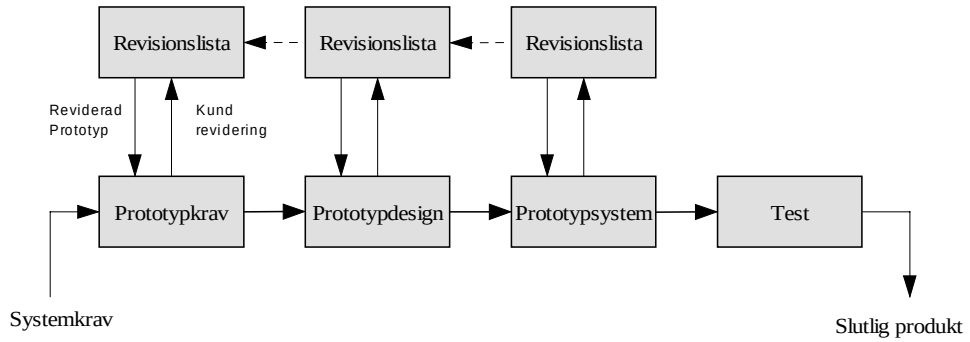
Vattenfallsmodellen skapade grunden för efterföljande modeller. En närbesläktad processmodell är V modellen [Pfleeger & Shari 1998] i Figur 9. I denna modell representeras relationssamband mellan testningsfasen, kravhanteringsfasen och designfasen. Genom dessa samband kan problem som uppstår under verifiering eller validering åtgärdas innan den slutliga testningen av programvaran.



• Figur 9 V modellen

2.6.3. **Prototypmodellen**

Idén med Prototypmodellen i [Pfleeger & Shari 1998] är att genom imitering av den förväntade produkten så snabbt som möjligt utveckla ett programvarusystem. Figur 10 visar hur Prototypmodellen kan vara uppbyggd. Denna modell kan tillämpas när man inte vet hur programvaran kommer att fungera. Med hjälp av denna modell kan utvecklaren upptäcka, särskilja och definiera de exakta kraven från kunden. I första processfasen utvecklas en prototyp som valideras och verifieras. I nästa steg, med kundens involvering, definieras de borttappade eller nya kraven och hela proceduren upprepas. Förbättring av produkten sker därmed stegvis. När kunden och utvecklaren är nöjda med resultatet anses produkten vara klar.



• Figur 10 Prototypmodellen

Denna typ av processutveckling är kostnads- och tidseffektiv. Det finns dock risk att kunden börjar använda en prototyp som produkt innan den är färdigutvecklad. Detta kan leda till större risk för buggar i programvaran samt en ofullständig dokumentering av hela systemet. Dålig dokumentering kan få konsekvenser vid vidareutvecklingen av produkten när inte alltid exakt samma grupp av utvecklare är involverade.

2.6.4. **Win-Win Spiralmodellen**

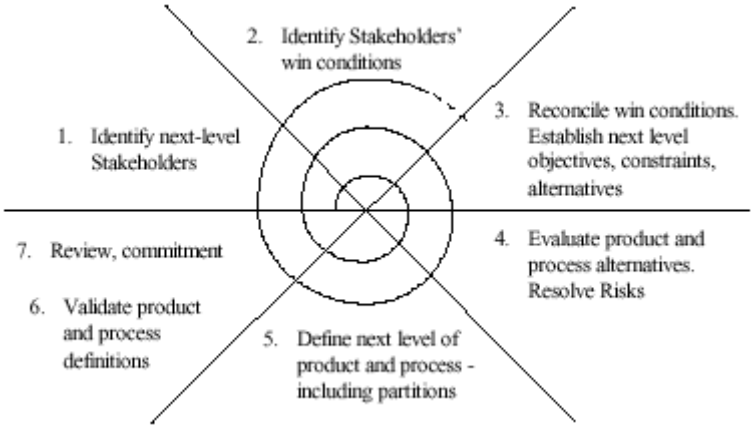
Win-Win Spiralmodellen [Boehm & Bose 1994] är en av de modernare processmodellerna som finns att tillgå. Win-Win modellen är en förfinig av ursprungsmodellen Spiralmodellen [Pfleeger & Shari 1998] och bygger på Win-Win teorin[8]. Win-Win modellen har fått ärva samtliga av Spiralmodellens egenskaper. Den har förbättrats genom att öka antalet faser som utvecklingsprocessen består av. Denna modell ger en bättre och mer sanningsenlig bild av den verkliga systemvärlden än andra utvecklingsmodeller.

Win-Win Spiralmodellen kännetecknas av metamodell med systematiskt tillvägagångssätt, omfattande och täckande processgång samt prototyptänkande. Den fokuserar på de mest risktagande delarna av programvaruutvecklingen samt inkorporerar andra processmodeller.

Win-Win Spiralmodellen är formad som en spiral. Idén med spiralformen är att följa spiralen från mitten utåt och genomgå de processfaser som beskrivs där. Grundprincipen med denna utvecklingsmodell är att gå flera varv i spiralen och på så sätt hinna med flera versioner av det önskade programvarusystemet innan slutprodukten levereras. Kunden och utvecklaren kan under hela utvecklingsprocessen följa programvaran och därmed förstå och reagera på de risker som finns på varje utvecklingsnivå. Slutprodukten har därmed större chanser att bli både stabil och kvalitativt bättre eftersom fel elimineras på ett tidigt stadium då validering av systemet sker kontinuerligt.

Win-Win Spiralmodellen är uppdelad i sju huvudfaser som alla passeras upprepade gånger. Dessa sju faser bildar en livscykel [Boehm *m.fl.* 1998]. Varje sådan cykel kan genomgås med hjälp av andra processmodeller beroende på vilken aktivitet i varje cykel som anses viktigast att genomföra. Denna flexibilitet underlättar utvecklingen av programvarusystem och gör användningen av denna modell mer kostnadseffektiv än av övriga processmodeller. Win-Win Spiralmodellen presenteras i Figur 11.

[8] en managementteori som behandlar problemet med många olika intressenter med motstridiga mål och ger riktlinjer för hur dessa mål löses på ett effektivt sätt.



• Figur 11 Win-Win Spiralmodell

De sju huvudfaserna är:

1. Identifiera systemets intressenter.
2. Identifiera intressenternas villkor för systemet eller underprogram.
3. Fastställa nästa nivås objekt, begränsningar och möjligheter.
4. Utvärdera produktens och processens möjligheter samt riskerna.
5. Definiera produkt och process. Planera nästa cykel och uppdatera livscykel plan.
6. Validera definitionen av produkt och process.
7. Utvärdera definitionen av produkt och process.

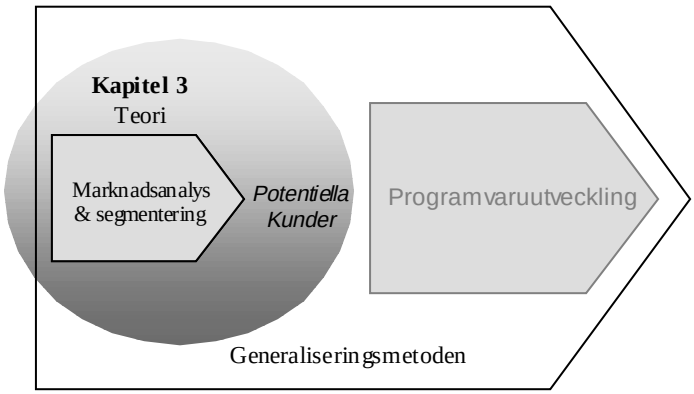
2.7. Slutord

I detta kapitel har många utvecklingsmodeller presenterats. En slutsats är att Kravhanteringen är en viktig del av en utvecklingsprocess eftersom den har en framträdande roll i alla de ovan beskrivna modellerna. För utveckling av Generaliseringsmetoden valdes Vattenfallsmodellen som arbetsexempel, eftersom den är både överskådlig och välkänd. Det är dock viktigt att poängtera att alla ovanstående processmodeller med fördel kan användas tillsammans med Generaliseringsmetoden.

3. Marknadsanalys och segmentering – en översikt

För att uppnå företagets mål måste behov och önskemål inom den potentiella marknaden bestämmas och det eftersökta levereras mer effektivt och skickligt än konkurrenterna. Detta kapitel behandlar marknadsföringens grunder samt hur man bl.a. kan segmentera och positionera en marknad.

Marknadsanalys och segmentering används som grund för Generaliseringsmetoden enligt Figur 12.



• Figur 12 Generaliseringsmetoden[9] - en översikt

3.1. Produkt

En produkt är det som kan erbjudas till en marknad för användning, förvärv, konsumtion, mm.. och som kan tillfredställa behov och önskemål. En produkt kan enligt [Kotler *m.fl.* 1998] analyseras som kärnprodukten, den verkliga produkten eller den utvidgade produkten.

Kärnprodukten är den egentliga fördel kunden köper. Den verkliga produkten byggs upp runt kärnprodukten tex. speciell utrustning, formgivning, kvalitet, märkesnamn, förpackning, etc..

Den utvidgade produkten är den service som byggs upp runt kärnprodukten och den verkliga produkten.

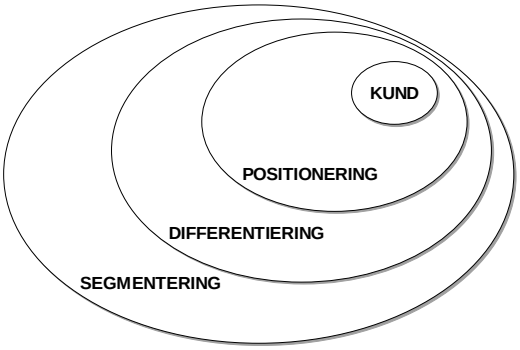
3.2. Marknadsföring och marknadsstrategi

Marknadsföring handlar om företagets samspel med marknaden – dvs. i huvudsak samspelet med nuvarande och potentiella kunder [Olsson, Skärvad 93]. Marknadsföring kombinerar många aktiviteter exempelvis marknadsundersökningar, produktutveckling, distribution, prissättning, reklam, försäljning, mm.. Utarbetning för marknadsföring består av följande delar:

[9] Generaliseringsmetoden beskrivs i kapitel 4

Marknadssegmentering	Ett marknadssegment är en grupp av kunder som reagerar lika på en given marknadsföringsstrategi. Företaget måste dela upp marknaden i sådana segment.
Marknadsinriktning	Företaget väljer ut ett eller flera segment som det skall inrikta sig på.
Marknadspositionering	Att arrangera så att produkten intar en klar och distinkt plats relativt sina konkurrenter hos målgruppen.
Differentiering	Att utveckla produkten så att denna skiljer sig från konkurrenternas motsvarigheter.

Figur 13 nedan visar en modell [Anderson, 1996] av sambanden mellan kunden och de tre strategierna positionering, differentiering och segmentering, vars syfte är att få kunden att fastna för företagets produkt.



• Figur 13 De tre ringarna

Ett företags framgång och tillväxt kräver en strategisk planering. Marknadsföring är ett hjälpmedel i den strategiska planeringen, och den övergripande strategiska planen definierar marknadsföringens roll i företaget. [Mintzberg 1998] skiljer mellan tre strategiska perspektiv:

- Strategi som plan.
- Strategi som handlingsmönster.
- Strategi som position och förändring av position.

”Strategi som plan” innebär att strategin uppfattas som en medveten handlingslinje för hur en given situation skall hanteras medan ”Strategi som handlingsmönster” avser det sätt på vilket strategier i en organisation växer fram och tar form, [Bruzelius & Skärvad 1989].

”Strategi som position” handlar om på vilka affärsområden strategin skall gälla samt hur företaget skall agera för att bli konkurrenskraftigt. Tabell 1 beskriver strategi som förändring av position i en produktmarknad enligt Ansoffs teorier i [Ansoff 1965]. De fyra möjliga sätten för tillväxt är följande:

1. Marknadspenetration: att öka existerande produkts andel på en existerande marknad utan produktförändringar.
2. Produktutveckling: att erbjuda en modifierad eller ny produkt till en gammal marknad.
3. Marknadsutveckling: att identifiera och utveckla nya marknader för existerande produkt.

4. Diversifiering: att utveckla nya produkter utanför tidigare produkter och marknader.

Marknader / Produkter	Nuvarande	Nya
Nuvarande	Marknads-penetration	Marknads-utveckling
Nya	Produkt-utveckling	Diversifiering

• Tabell 1 Produktmarknadstabell

Vidare nämns i [Bruzelius & Skärvad 1989] att ett företag bör sträva efter att verka inom affärsområden med god lönsamhetspotential och inom områden där företaget kan uppnå konkurrensfördelar. Ett affärsområdes lönsamhetspotential påverkas av antalet konkurrenter inom branschen, risken för nykommande aktörer på marknaden, nya teknologier som kan ersätta tjänsten/produkten, kundernas och leverantörernas förhandlingsposition och makt.

Ett företag kan konkurrera genom tre grundstrategier enligt Porter i [Porter 1985] som visas i Tabell 2 Grundstrategier.

- Lågkostnadsstrategi: att hålla de lägsta kostnaderna i produktion och distribution.
- Differentieringsstrategi: att utveckla en tjänst/produkt med unika egenskaper.
- Fokuseringsstrategi: att rikta sig till en specifik målgrupp, att hålla en särskild produktlinje, att koncentrera sig på en viss geografisk marknad, etc.

		Konkurrensfördelar	
		Lägre kostnad	Differentiering
Betjänad marknad	Hela marknaden	Kostnadsledarskap	Differentiering
		Fokuseringsstrategier	
	Del av marknaden	Kostnadsbaserad fokusering	Differentieringsbaserad fokusering

• Tabell 2 Grundstrategier

3.2.1. Internationell marknadsföring

När hemmamarknaden blir mättad eller när man koncentrerat sig på en viss kundkrets som är liten på hemmamarknaden uppstår ofta viljan att marknadsföra sina produkter internationellt. Många företag är globala idag och det är viktigt att inte komma efter sina konkurrenter. Internationalisering

är en komplicerad process och ett företag behöver ett systematiskt sätt att fatta sina internationella marknadsföringsbeslut.

Det är viktigt att förstå den internationella marknadens miljö och handelssystem. Man måste ta hänsyn till det ekonomiska, politiska och kulturella klimatet.

Vidare bör ett företag definiera sina marknadsföringsmål dvs. försäljningsvolymen, vilka typer av länder och hur många länder man är intresserad av att expandera i.

Ett företag måste veta vilka marknader som det vill satsa på. Utländska marknader kan rangordnas beroende på marknadsstorlek, marknadstillväxt, kostnaderna att göra affärer, konkurrenssituationen och risknivån.

Det är viktigt för ett företag att veta hur företaget skall gå in på marknaden, tex. genom joint venture eller direkt investering. Joint venture kan vara licensiering, kontraktstillverkning, ledningskontrakt eller delat ägande.

Slutligen måste företaget ta ett beslut om marknadsföringsprogram och om marknadsföringsorganisation. Att utveckla en effektiv organisation för att bearbeta den utländska marknaden kan vara en svår process. De flesta tjänsteföretag börjar med en internationell division för att senare skapa en global organisation.

3.2.2. Marknadsföring av tjänst

En tjänst är en abstrakt produkt som erbjuds kunden och som kunden har fördel av. Tjänster är enligt [Kotler *m.fl.* 1998] abstrakta, svåra att skilja från leverantören, varierande med omständigheterna och flyktiga. Kvalitén på tjänster kan variera kraftigt beroende på vem som tillhandahåller dem och hur de tillhandahålles. Marknadsförare måste finna sätt att göra tjänsterna mer konkreta, att öka produktiviteten för kunden och att förbättra utbudet trots tjänsternas flyktighet. Det krävs andra strategier för marknadsföring av en tjänst än de fyra traditionella som är rätt produkt, rätt plats, rätt påverkan och rätt pris. Dessa tillkommande krav är följande:

- Intern marknadsföring vars syfte är att effektivt utbilda och motivera de anställda som har kundkontakt samt alla serviceanställda till att arbeta i lag som tillhandahåller kundtillfredsställelse.
- Samspelsmarknadsföring vars syfte är att utveckla ett samspel mellan det tjänstlevererande företaget och kunden.

En vanlig typ av marknad är business-to-business[10] marknad, där både leverantören och kunden är tjänsteföretag. Det är viktigt att samspelet mellan företaget och kunden fungerar bra och att krav och önskemål från kunden tillfredställs noggrant. Detta eftersom slutkundens önskemål har blivit tolkade flera gånger och missuppfattningar kan lätt förekomma.

Ett företag kan bli konkurrenskraftigt om det skiljer sig från konkurrenterna. Differentieringen kan exempelvis göras med hjälp av ett lägre pris på tjänsten eller genom att utveckla tjänster med nya karaktärsdrag, etc.. Att hålla hög kvalitet på tjänsten och därmed överträffa kundens förväntningar är ett annat sätt att skilja sig från konkurrenterna. En annan viktig faktor är att upprätthålla och öka produktiviteten samt att hålla tidsplanen för leverans av produkten.

[10] Organisationsmarknad, se kapitel 3.3

3.3. Marknadstyper

En marknad utgörs av alla potentiella och verkliga kunder. Marknad kan definieras som de personer eller organisationer som köper eller kan tänkas vilja köpa varor och tjänster för att tillfredställa sina behov, [Olsson, Skärvad 1993].

Det finns två olika marknadstyper: konsumentmarknad och organisationsmarknad. Det finns väsentliga skillnader mellan dessa två marknader. Organisationsmarknaden består av färre men större kunder som ofta är geografiskt koncentrerade. Denna marknads behov beror på behovet hos konsumentmarknaden. Fler än en person är involverade i köpbesluten och dessa personer är oftast utbildade i konsten att förhandla. Vidare är denna marknad relativt okänslig för prisändringar.

Organisationsmarknad kan enligt [Kotler *m.fl.* 1998] indelas i tre huvudklasser:

1. Industrimarknaden. Köpbesluten varierar med vilken av tre köpsituationerna köparen befinner sig i, återkommande standardköp, återkommande modifierat köp, eller nytt köp. Det är viktigt för marknadsföraren att ta reda på vilka personer hos den tilltänkta kunden är beslutsfattande, vilka befogenheter och inflytande dessa personer har samt vilka kriterier för urval de använder sig av. Köpprocessen kan också påverkas av miljöfaktorer tex. ekonomi, organisationsfaktorer tex. struktur, internpolitik, mm..
2. Återförsäljarmarknaden. Denna marknad består av företag och individer som köper varor för att sälja vidare. Köpbesluten kan fattas av en person och beror på produkternas pris och sortiment, leverantörer, leveransvillkor, mm..
3. Statliga marknaden. Denna marknad karaktäriseras av att många människor är inblandade i köpbesluten tex. en centralenhet och bevakas oftast av allmänheten. Orderna är oftast mycket stora och ett öppet anbudsförfarande är ett vanligt sätt att välja leverantör.

3.4. Marknadssegmentering

Varje företag måste hitta rätt marknad för att lyckas sälja sina produkter. Ett företag kan leverera sin produkt eller sina produkter på följande sätt:

- Massmarknadsföring: att massproducera, massmarknadsföra och massdistribuera en produkt till alla kunder.
- Produktvarierad marknadsföring: att utveckla en eller flera produkter för en marknad som skall ge kunderna omväxling snarare än attrahera olika marknadssegment.
- Målinriktad marknadsföring: att identifiera marknadssegment och sedan utveckla produkter som anpassas till varje specifikt segment.

3.4.1. Målinriktad marknadsföring

Målinriktad marknadsföring har enligt [Kotler *m.fl.* 1998] tre nyckelelement: segmentering, målinriktning och positionering. Marknadssegmentering innebär att marknaden indelas i grupper efter följande variabler:

- Geografiskt: innebär att marknaden indelas efter vissa geografiska områden.
- Demografiskt: innebär att marknaden delas in i grupper beroende på faktorer som tex. kön, ålder, yrke, utbildning, etc..

- Psykografiskt: innebär att marknaden indelas i grupper beroende på social klass, livsstil, personliga karaktärsdrag, etc..
- Beteendemönstermässigt: innebär att indelningen av marknaden sker beroende på kundens kunskap, attityd och respons på en produkt. Kundens lojalitet till ett företag eller ett märke kan spela en betydande roll vid valet av produkt.
- Behovsmässigt: innebär att marknaden indelas efter kundens behov.

Marknadspositionering innebär den plats som kunden tycker att en produkt intar jämfört med konkurrerande produkter. Positionering består av följande steg:

- Identifiera möjliga konkurrensfördelar som kan utgöra grund för positioneringen.
- Välja rätt konkurrensfördelar.
- Kommunicera och leverera den valda positionen.

3.5. Marknadsföringens informationssystem

När man genomför marknadsföringsåtgärder måste beslut fattas som kan få stora ekonomiska konsekvenser. Genom att visa att dessa beslut är välgrundade måste företaget tillhandahålla ett fungerande informationssystem. Marknadsföringens informationssystem är uppbyggt av människor, verktyg, och procedurer som samlar in, sorterar, analyserar och utvärderar information.

Informationsinsamling kan ske på följande sätt:

- Interninformation: att använda information från tidigare projekt medför att ofullständiga fakta från de tidigare projekten inte är identiska med de nya.
- Allmäninformation: att ta tillvara information som finns hos företagets egen personal, leverantörer samt kunder.
- Marknadsundersökning: att undersöka marknaden.

3.5.1. Marknadsundersökning

Marknadsundersökningar har flera användningsområden. De kan användas som ett kontrollinstrument för att hålla reda på hur marknaden är uppdelad mellan konkurrenterna. De kan användas i samband med lansering av nya produkter för att följa en produkts utveckling. Ett annat viktigt användningsområde är vid diversifiering eller produktutveckling, se ovan.

En marknadsundersökning genomföres i fyra steg:

1. Definiera problem och undersökningsmål. Undersökningsmålet kan vara av tre typer:
   - Undersökande: att samla information för att bättre kunna definiera problemet.
   - Beskrivande: att beskriva kundens attityd vid köp av en produkt.
   - Orsak/Verkan: att undersöka hypoteser mellan orsak och verkan.
2. Undersökningsplan. Undersökningsplan går ut på att planera insamlandet av data och begränsa den insamlade informationsmängden. Insamling av data kan ske genom observationer, enkäter och experiment.

3. Implementering av undersökningsplanen. Samla in och analysera informationen.
4. Slutsatser och tolkning av resultat.

3.6. Slutord

Inblicken i, och förståelsen för marknadsföringens grunder har kunnat utnyttjas vid fallstudien av Generaliseringsmetoden. Marknadsföringens olika delar ses som modeller. Kunskaperna uttryckta som dessa modeller har varit verktyg för fallstudien i Kapitel 5.

4. Generalisering av programvarusystem

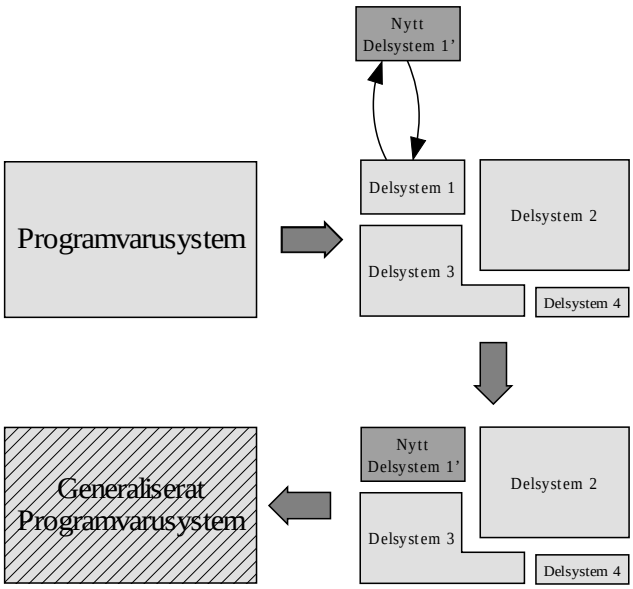
I detta kapitel beskrivs en Generaliseringsmetod som förväntas göra det möjligt att anpassa ett programvarusystem byggt åt en specifik kund till andra kunder.

4.1. Introduktion

Utveckling av en Generaliseringsmetod kräver en arbetsmetodik som följer riktlinjerna för konstruktionen av ett programvarusystem. Dessa riktlinjer definieras med hjälp av programvaruteknikens grunder i kapitel 2.

Dessutom behöver en marknadssegmentering och marknadsanalys göras för att ta fram kundunderlag till produktutvecklingen. Detta kan göras med hjälp av grunderna i Marknadsanalys och segmentering i kapitel 3.

Generalisering av programvarusystem bör göras på enklaste möjliga sätt. Ju mindre som behöver ändras i ett befintligt system desto lättare är det att behålla systemets ursprungliga form. Stora ändringar i ett programvarusystem kan leda till efterverkningar hos andra systemdelar och förändring av hela systemstrukturen. Önskvärt vore att generalisera ett befintligt system utan att behöva ändra systemstrukturen (och påverka eller förändra de andra delarna av systemet). Detta förtydligas med en enkel modell i Figur 14.

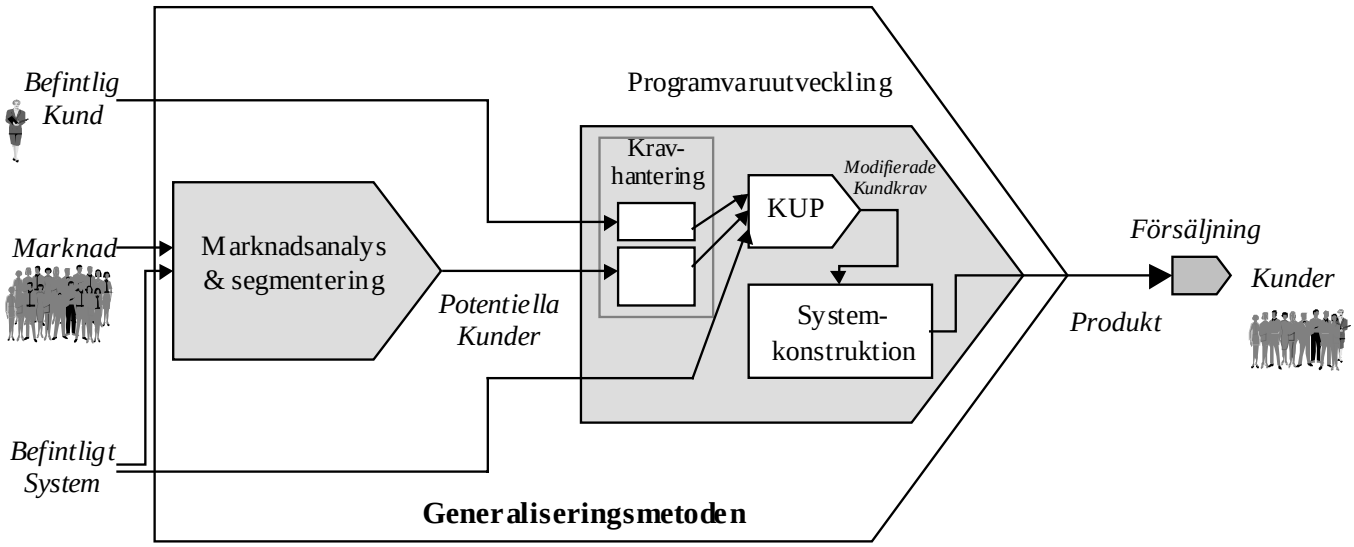


• Figur 14 Generalisering av ett programvarusystem

Generaliseringsmetoden som presenteras i denna rapport är ett sätt för hur man kan generalisera ett befintligt programvarusystem. Många krav ställs på slutprodukten som är resultatet av Generaliseringsmetoden och man kan säga att ”… val av rätt produkt är ofta en icke-trivial uppgift som kräver ett försiktigt balanserande mellan kundkrav, produkts funktionella förmåga och system arkitektur” i [Ncube 2000].

4.2. Grafisk presentation av Generaliseringsmetoden

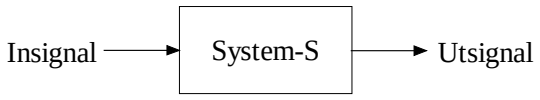
Generaliseringsmetoden presenteras i form av ett tankeschema för att lättare kunna överblicka och förstå principerna med denna metod, se Figur 15.



• Figur 15 Generaliseringsmetoden

4.3. Principen med Generaliseringsmetoden

Principen med Generaliseringsmetoden kan enklast beskrivas med fysikaliska termer som insignal-utsignal modell. Insignal är det som kommer in i system-S. System-S verkar på insignalen och skickar ut ett resultat som är en utsignal, se exempel i Figur 16.



• Figur 16 Insignal-utsignal modell

Generaliseringsmetoden inverkar på inparametrarna *Befintlig Kund*, *Marknad* samt *Befintligt System* vilket resulteras i utparametern *Produkt*.

In- och utparametrar

Befintlig Kund	är den ursprungliga kunden till det programvarusystem som skall generaliseras.
Marknad	är en odefinierad kundkrets som kan tänkas vilja köpa *Produkten*.
Befintligt System	är det befintliga systemet som skall generaliseras.

Produkt	är det generaliserade programvarusystem som är resultatet av *Generaliseringsmetodens* inverkan på *Befintlig Kund*, *Marknad* samt *Befintligt System.*
Kunder	är personer eller organisationer som är de verkliga köparna av *Produkten.*
Försäljning	innebär i detta sammanhang ett tillvägagångssätt för hur en produkt skall marknadsföras och säljas till en ny kundkrets, exempelvis vilka försäljningskanaler som skall användas, vilken försäljningsmetodik som skall följas, etc..

Produkten dvs. det generaliserade systemet kan sedan säljas med hjälp av *Försäljning* till *Kunder*. Kunderna kan kräva service i form av exempelvis support och underhåll. Detta behandlas som enskilda fall genom det kontrakt som skrivits mellan kunden och leverantören av systemet. *Försäljning* eller vad som händer sedan med *Produkten* diskuteras inte i denna rapport.

4.4. Generaliseringsmetodens beståndsdelar

Det är viktigt att undersöka vilka målgrupper ett företaget bör inrikta sig på vid lanseringen av ett generaliserat programvarusystem och hur systemet bör utformas för att bäst passa de potentiella kunderna. Generaliseringsmetoden har två huvuddelar med olika funktionalitet, *Marknadsanalys och segmentering* och *Programvaruutveckling.* De beskrivs nedan och fungerar enligt den i Kapitel 4.3 beskrivna modellen.

4.4.1. Marknadsanalys- och segmenteringsmetoden

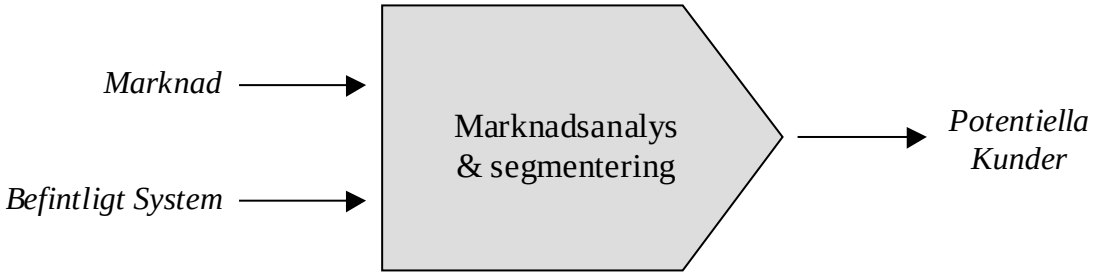
Marknadsanalys- och segmentering är en metod för analys och segmentering av en marknad. *Marknadsanalys och segmentering* används för att göra en marknadsundersökning av vilka personer eller organisationer som kan bli potentiella kunder till det generaliserade programvarusystemet. Marknadsundersökningen skall göras efter vissa förbestämda kriterier.

In- och utparametrar

Befintligt System	är det befintliga systemet som skall generaliseras.
Marknad	är en odefinierad kundkrets som skall analyseras och segmenteras.
Potentiella Kunder	är tänkbara kunder till det generaliserade systemet.

Inparametern till *Marknadsanalys och segmentering* är *Marknad* och *Befintligt System* medan utparametern är *Potentiella Kunder.* Marknaden skall avgränsas efter kriterier som beror på både det programvarusystemet som skall generaliseras och beställaren[11] av generaliseringen. Skulle det förhålla sig så att leverantören av det generaliserade systemet redan har flera kunder som är intresserade av att köpa systemet kommer dessa kunder att utgöra huvuddelen av det potentiella kundunderlaget. Figur 17 visar en modell av Marknadsanalys- och segmenteringsmetoden.

[11] Den som beställer generalisering av det befintliga systemet, oftast marknadsförare.



• Figur 17 Marknadsanalys- och segmentering

Det finns många beprövade sätt att segmentera en marknad varför det inte skulle vara meningsfullt att ägna sig åt detta i denna rapport. Metoden kan därför väljas fritt efter egna erfarenheter och kunskaper så länge som syftet med Marknadsanalys och segmentering kan uppfyllas.

4.4.2. Programvaruutvecklingsmetoden

Utvecklingsprocessen för ett generaliserat programvarusystem följer riktlinjerna för en modifierad processmetod som bygger på principerna för programvaruutveckling. Programvaruutvecklingsmetoden är en modifierad processmetod.

Programvaruutvecklingsmetoden har till syfte att ge en överskådlig bild av processförloppet för utveckling av ett generaliserat programvarusystem.

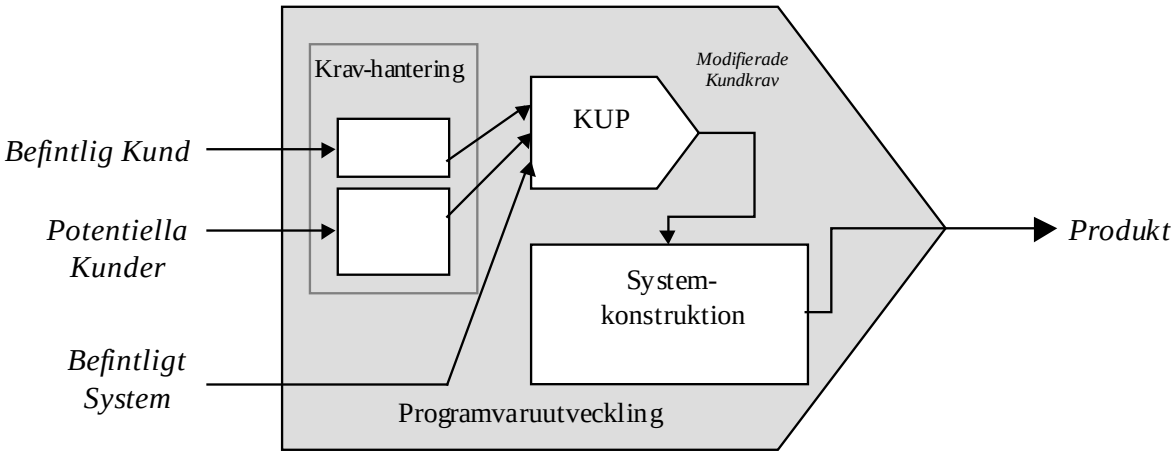
Programvaruutveckling kräver ett metodiskt och effektivt tillvägagångssätt. Genom att tillämpa rätt processmodell fås riktlinjer för hur man bygger en programvara. Processen för utveckling av en programvara kan bli mycket komplicerad och långdragen. Därför bör enkelhet eftersträvas. Enkelhet, förståelse och hanterlighet är nyckelorden.

Det finns flera användbara processmodeller som kan tillämpas vid uppbyggnad av det generaliserade systemet. Ett urval av dessa processer har åskådliggjorts i Kapitel 2. Det befintliga systemet har troligtvis utvecklats enligt en processmodell och det kan därför vara lämpligt att behålla denna modell ”för att inte behöva tillägnas en onödig tid på strukturförändringar” av det nya systemet. Det är naturligtvis fritt att välja processmodell.

In- och utparametrar

Potentiella Kunder	är de tänkbara kunder till det generaliserade systemet som Marknadsanalys och segmentering har frambringat.
Befintlig Kund	är den ursprungliga kunden till det programvarusystem som skall generaliseras.
Befintligt System	är det befintliga systemet som skall generaliseras.
Produkt	är det generaliserade befintliga programvarusystem.

Inparametrarna till *Programvaruutvecklingsmetoden* är *Potentiella Kunder* och *Befintligt System* medan utparametern är *Produkten*. Figur 18 illustrerar Programvaruutvecklingsmetoden grafiskt.



• Figur 18 Programvaruutvecklingsmetod

4.4.2.1. Kravhantering

Under första fasen av utvecklingsprocessen enligt *Programvaruutvecklingsmetoden* utförs *Kravhantering*. Kravhanteringsfasen är den grundläggande fasen i utvecklingsprocess för programvarusystem oavsett vilken processmodell som används. Genom att följa kravhanteringsprocessen, dvs. processen för identifiering av kundkraven, kan kraven som kunden ställer på programvaran identifieras. Enligt [Boehm 1980] kostar det upp till 100 gånger mer att korrigera fel i kraven i sena testfaser än att korrigera dem i kravhanteringsprocessen varför kravhanteringsprocess är en oerhört viktig del i utvecklingen av ett programvarusystem.

In- och utparametrar

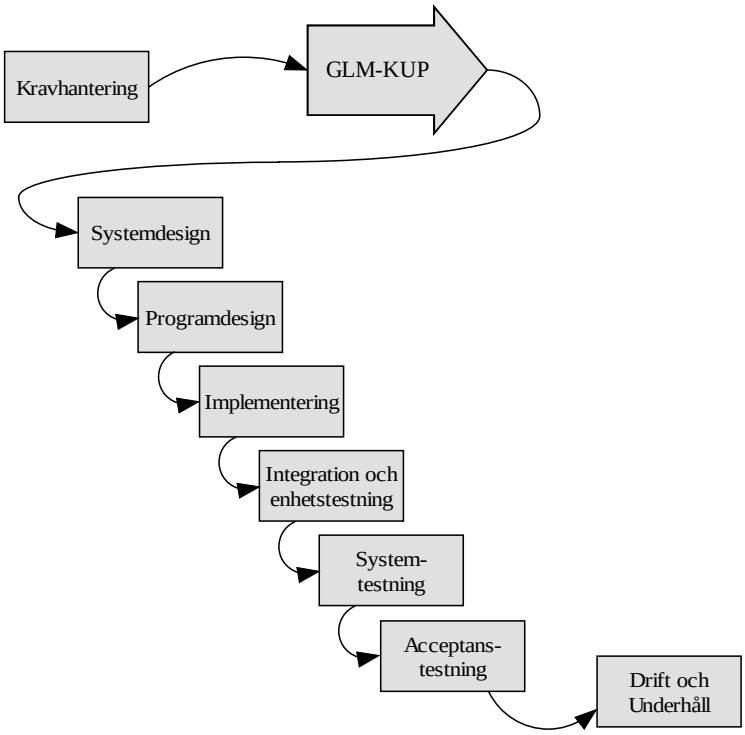
Potentiella Kunder	är de tänkbara kunder till det generaliserade systemet som Marknadsanalys och segmentering har frambringat.
Befintlig Kund	är den ursprungliga kunden till det programvarusystem som skall generaliseras.
Kundkrav	är de krav som de potentiella kunder och den befintliga kunden ställer på det generaliserade systemet.

Inparametrar i *Kravhanteringen* är *Potentiella Kunder* och *Befintlig Kund*. När en ny marknad skall beträdas måste denna marknads krav och önskemål tillgodoses. Om det är möjligt att ta del av den befintliga kundens krav kan dessa vara värdefulla vid utvecklingen av ett nytt system som bygger på det befintliga.

Kravhantering består av likadana moduler som alla utför kravhantering på var sin potentiell kund. Kravhanteringen kan följa de rutiner som rekommenderas för programvaruutvecklingen. Vid generaliseringen underlättas kravhanteringen kraftigt eftersom man har en erfarenhet av det befintliga systemet och alla rutiner därmed kan upprepas och återanvändas. Dessutom kan det finnas några processteg som är samma för alla moduler och därför behöver utföras bara en gång. Utparametrar i Kravhanteringen är kundkraven insamlade i paket, ett paket för varje potentiell kund.

4.4.2.2. Kravurvalsprocess KUP

Efter att kravhanteringen har utförts sker nästa aktivitet Kravurvalsprocess *KUP* som också är kärnan i *Programvaruutvecklingsmetoden. KUP* står för Generaliseringsmetodens kravurvalsprocess. *KUP* är ett mellansteg mellan Kravhanteringen och Systemdesignen eller andra delar av en utvecklingsprocess som kommer efter Kravhanteringen. För att lättare få en inblick i den modifierade utvecklingsprocessen appliceras *KUP* på den traditionella Vattenfallsmodellen[12] i Figur 19.



• Figur 19 KUP applicerad på Vattenfallsmodellen.

Utveckling av en programvara bygger på att tillfredsställa kundens behov och önskemål genom att uppfylla de krav som kunden ställer på systemet. Därför bör kravhanteringsfasen finnas med i alla utvecklingsprocesser. *KUP* länkas alltid till Kravhanteringen efter att Kravhanteringsfasen har utförts.

In- och utparametrar

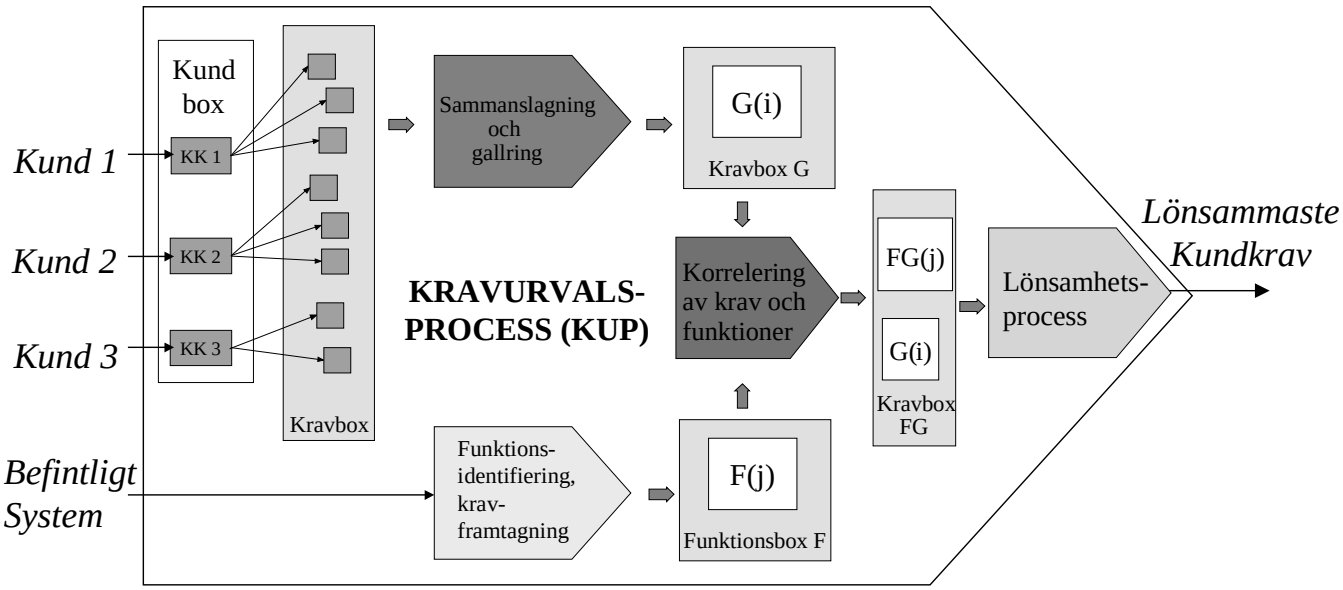
Kundkrav är de krav som de potentiella kunderna och den befintliga kunden ställer på det generaliserade systemet. Dessa kundkrav är fördelade i paket, ett paket för varje potentiell kund.

Befintligt System är det befintliga systemet som skall generaliseras.

De mest lönsamma krav är de mest lönsamma kraven som *KUP* har framtagit.

[12] en processmodell, beskrivs i Kapitel 2.6.

Inparametrar i *KUP* är *Kundkraven* och *Befintligt System*. Dessa inparametrar hanteras först var och en för sig för att sedan sammanslås till nya enheter. Figur 20 illustrerar processförloppet för *KUP*.



• Figur 20 Kravurvalsprocessen KUP

Kundbox

I Kundboxen finns de insamlade kundkraven sorterade i mindre kundkravsboxar per kund. För att markera vilka krav kommer från vilken kund, betecknas varje kundkravsbox med KK(i) där i = 1, 2, …, n. KK(i) kan beteckna en kundkravsbox som innehåller ett eller flera kundkrav från någon specifik potentiell kund eller från den befintliga kunden.

Den första potentiella kunden tex. Företag1 AB kan betecknas som KK(1), Företag2 AB som KK(2) osv.. Varje kund kan ha flera krav på systemet. Kunden Företag1 AB har 8 krav som läggs in i Företag1 AB:s kundkravsboxen KK(1). På samma sätt behandlas de andra kundernas krav.

Kravbox A

Kundkravspaketet ”öppnas” sedan i *Kravboxen A* där alla kundkrav samlas. Kundkraven betecknas med g(i), där i = 1, 2, 3, …

Sammanslagning och gallring

Vidare går *Kravboxen A* genom processen *Sammanslagning och gallring*. Processens uppgift är att sortera och gallra alla kundkraven g(i). Gallringen sker efter följande kriterier i nedan en angiven turordning.

1. Krav från olika potentiella kunder kan vara likartade. Exempelvis kan en kund vilja ha ett kundregister som sorteras efter namn medan en annan kund vill ha ett kundregister med exakt samma parametrar men sorterade efter kundnummer. Genom en sammanslagning av dessa två kundkrav fås ett nytt kundkrav där man efterfrågar ett kundregister sorterat efter både namn och kundnummer. Det nya kundkravet får samma beteckning g(i), där ”i” är det lägsta numret av de två kraven.

2. Kundkrav g(i) med låg efterfrågan elimineras efter en viss värdering. Förutsatt att det potentiella kundunderlaget är tillräckligt stort och att tillräckligt många kundkrav har samlats in. I annat fall måste alla kundkrav tas med i denna fasen.

   Kriteriet för en elimination är att efterfrågepoängen för ett kundkrav understiger en femtedel av maxpoängen per krav. Varje krav som skall elimineras måste först värderas efter hur den tänkta produkten förväntas fungera. Det kan förekomma att de potentiella kunderna inte uppmärksammat behovet av ett visst krav och därmed inte insett dess betydelse för systemet. Om ett krav bedöms utgöra en viktig del i systemet, måste det trots låg efterfrågan få vara kvar.

Kravbox G

Resultatet av processen *Sammanslagning och gallring* placeras i *Kravboxen G*.

Varje krav får en beteckning G(i) där i = 1, 2, 3, …

Funktionsidentifiering

I nästa steg sker funktionsidentifiering av *Det Befintliga Systemet.* Inparametern *Befintligt System* kan bearbetas parallellt med händelseförloppet mellan Kravbox A och Kravbox G.

Funktionerna i *Det Befintliga Systemet* är lösningar som utvecklats för att uppfylla den befintliga kundens ursprungliga krav. Har kundkraven modifierats och ändrats efter det att den ursprungliga versionen av systemet har utvecklats överensstämmer inte den gamla kravdokumentationen med de grunder vilka det nuvarande programvarusystemet är uppbyggt på. Det är funktionerna ur de befintliga systemet som man vill återanvända vilket motiverar till varför de skall identifieras.

Slutresultatet i processen *Funktionsidentifiering* är de funktioner som identifierats i det befintliga systemet.

Funktionsbox

De identifierade funktionerna i det befintliga systemet samlas i *Funktionsboxen*.

Varje funktion får en beteckning F(j) där j = 1, 2, .. , m, där m är maximala antalet funktioner i det befintliga systemet.

Korrelering av krav och funktioner

Korreleringsprocess av krav och funktioner görs på följande sätt:

1. Ta fram de kundkrav G(i) som kan uppfyllas av funktionerna F(j) och benämn dessa kundkrav i tur och ordning med FG(j).

2. Eliminera de övriga funktionerna F(j) som inte tillfredsställer några kundkrav G(i). Funktionerna som anses nödvändiga för att systemet skall fungera som förväntat måste vara kvar i systemet. Dessa funktioner behöver inte gå igenom lönsamhetsprocessen utan utgör också ett resultat av kravurvalsprocessen.
3. De övriga kundkraven står kvar som G(i).

Kravbox FG

Kundkraven FG(j) samt återstoden av kundkraven G(i) samlas i Kravbox *FG*.

Lönsamhetsprocess

Lönsamhetsprocess är en strategi för hur man beräknar de mest lönsamma kundkraven av de kundkrav som finns samlade i *Kravboxen FG*. Lönsamhetsberäkningen grundas på efterfrågan och kostnaden för att bygga funktioner som uppfyller kundkraven G(i) och FG(j). Kundkraven FG(j) är mycket billigare att uppfylla eftersom det är möjligt att återanvända funktioner från det befintliga systemet. Är dessutom efterfrågan stor på kundkraven FG(j) kan utveckling av ett system som uppfyller dessa vara mycket lönsamt.

Det finns olika tekniker för att utifrån en kravspecifikation eller en funktionsbeskrivning göra en lönsamhetsberäkning och det finns flera olika metoder att tillgå. Under fallstudien som gjorts i samband med generaliseringen har författaren utvecklat en enkel metod för lönsamhetsberäkning. Denna beräkningsmetod beskrivs i Kapitel 5.

4.5. Slutord

Vid tillämpning av Generaliseringsmetoden identifieras den potentiella marknaden för ett generaliserat programvarusystem och kundkraven samlas in. Kravhanteringen i en generell utvecklingsprocess modifieras med ett tillskott av kravurvalsprocessen KUP, vilken beräknar de lönsammaste kundkraven. Dessa kundkrav uppfylls sedan av funktioner som har utvecklats för det befintliga systemet eller kommer att nyutvecklas i utvecklingsprocessens efterkommande faser. Resultatet av utvecklingen blir ett generaliserat programvarusystem samt de potentiella kunderna, vilka utgör underlag för en ny marknad. För att påvisa detta tillämpas Generaliseringsmetoden i fallstudien i Kapitel 5.

5. Tillämpning av Generaliseringsmetoden

I detta kapitel verifieras Generaliseringsmetoden genom en fallstudie av ett kundvårdssystem för mobila telekommunikationsbranschen. Dessutom beskrivs vilken marknadspotential som kundvårdssystemet har internationellt.

5.1. Inledning

Det är viktigt att undersöka vilka målgrupper ett företaget bör inrikta sig på vid lanseringen av ett programvarusystem och hur systemets funktioner bör utformas för att bäst uppfylla kundernas krav.

Programvarusystemet som studeras här är ett kundvårdssystem KVS utvecklat av företaget Leverantör AB åt kunden Kund AB. Leverantör AB är ett företag som levererar tjänster och lösningar för utveckling av avancerade tekniska system medan Kund AB är ett tjänsteföretag som säljer telekommunikationstjänster åt privatpersoner och organisationer.

Målet med fallstudien är att identifiera de kundkrav som bör utgöra underlag för ett generaliserat kundvårdssystem. Kundkraven utgör förutsättningen för generalisering av kundvårdssystemet KVS. Det behövs enligt Generaliseringsmetoden kundkrav från uppskattningsvis 5 till 10 kunder. För att inhämta kundkrav har en enkätundersökning gjorts. Av Generaliseringsmetoden följer att minst 20 potentiella kunder måste identifieras. De potentiella kunderna har identifierats genom marknadssegmentering och analys.

En förutsättning för att ett kundvårdssystem skall vara intressant för en marknad är att marknaden redan är etablerad och inte under nyetablering.

5.2. Kundvårdssystem – översikt

Kundvårdssystem används för att hantera och upprätthålla kundrelationer samt för att öka servicen till kunderna. Kundvårdssystem har utvecklats för att öka samspelet mellan leverantören av en tjänst och kunden enligt Samspelsmarknadsföring, se Kapitel 3.2.2.

Kundvård är ett tillvägagångssätt för ett företag att förstå kundernas individuella köpbeteenden och att omsätta denna kunskap i en kontinuerlig dialog med var och en av dess kunder. Marknadsföringen fokuseras kring enskilda kunder med utgångspunkt i kundernas specifika behov. All relevant information om den omsättning som genereras per enskild kund lagras i en databas och används sedan för kundanalys och kundinteraktion.

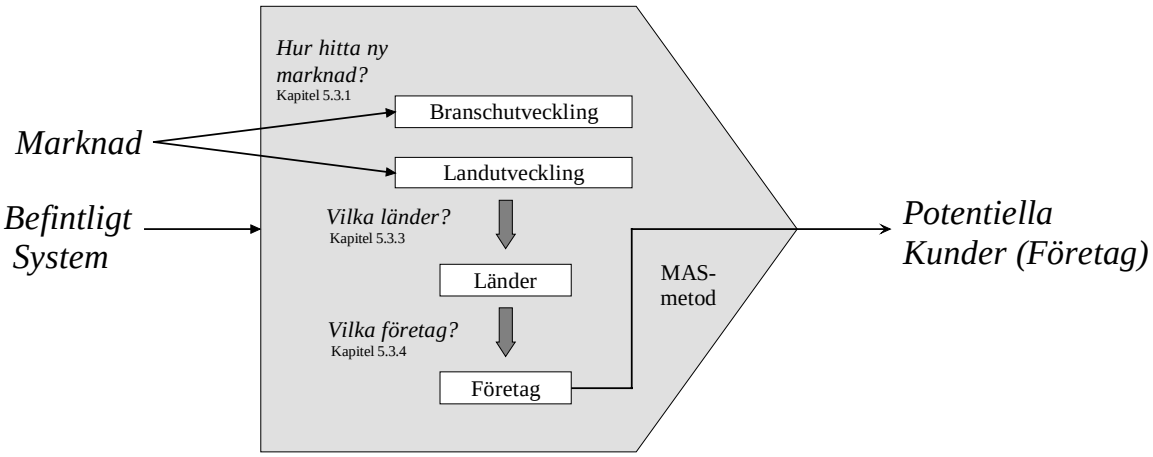
Företag som hellre skapar ett långsiktigt och ömsesidigt samarbete med befintliga kunder än skaffar nya kunder använder sig av kundvård. Genom att få inblick i kundens köpbeteende och kunskap om kundens individuella preferenser kan företaget skräddarsy erbjudanden till den enskilda kunden. Kunden blir förhoppningsvis nöjd och eftersom företaget blir bättre på att tillgodose kundens behov ökar kundens lojalitet till företaget. Detta leder i sin tur till att kunden blir mindre motiverad till att byta till en konkurrerande leverantör som saknar detaljkunskap om kundens behov och önskningar.

5.3. Segmentering och analys av marknaden

I detta delkapitel har Marknadsanalys och segmentering i Kapitel 4.4.1 tillämpats på kundvårdssystemet KVS.

För att en generalisering av kundvårdssystemet KVS skall bli möjlig måste en ny marknad hittas. För att hitta en ny marknad används ”marknadsutveckling” enligt Ansoffs teorier i Kapitel 3.2, dvs. nya marknader hittas för existerande produkt.

Målet med marknadssegmenteringen och analysen är att identifiera ungefär 20 till 30 potentiella kunder att hämta krav ifrån. Dessa kundkrav skall ligga till grund för vidare generalisering av kundvårdssystemet KVS. Tillvägagångssättet för marknadssegmenteringen följer tankeschemat i Figur 21.



• Figur 21 Tankeschema för segmentering av marknaden för KVS

In- och utparametrar

Kundvårdssystem KVS	är det befintliga kundvårdssystemet som skall generaliseras.
Marknad	är en odefinierad kundkrets som skall analyseras och segmenteras.
Potentiella Kunder	är tänkbara kunder till det generaliserade kundvårdssystemet KVS.

Följande delkapitel beskriver steg för steg hur och efter vilka kriterier som marknadssegmentering och marknadsanalysen har gått till.

5.3.1. Hur hitta ny marknad?

Det befintliga kundvårdssystemet KVS är byggt för telekommunikationsbranschen och den svenska marknaden. För att hitta lämpliga kunder till kundvårdssystemet KVS kan marknaden segmenteras med hänsyn till följande två variabler:

- Bransch
- Land

Tabell 3 visar kombinationerna av marknadssegmenteringen för kundvårdssystemet KVS.

Länder / Branscher	Nuvarande →	Nya →
Nuvarande ↓	X	Lands-utveckling
Nya ↓	Bransch-utveckling	Land och bransch utveckling

• Tabell 3 Marknadssegmentering för kundvårdssystemet KVS

I enlighet med tabellen ovan finns det tre möjliga sätt att göra marknadsutveckling på med hänsyn till variablerna land och bransch:

1. Branschutveckling, dvs. nya branscher och nuvarande land.
2. Landsutveckling, dvs. nya länder och nuvarande bransch.
3. Lands- och branschutveckling, dvs. nya branscher och nya länder

Den fjärde kombinationen, ”X” i tabellen, är inte relevant eftersom Leverantören AB vill fortsätta att samarbeta med kunden som kundvårdssystemet byggdes åt istället för att hjälpa kundens närmaste konkurrenter och därmed riskera att tappa en bra kund.

Att vid strävan att finna ny marknad bearbeta ny bransch och nytt land, Lands- och branschutveckling, är mycket krävande och rekommenderas inte. Genom att välja marknadsindelningen med en av variablerna konstant, bransch eller land, kan nytta dras av de kunskaper och erfarenheter som införskaffats under utvecklingen och försäljningen av det befintliga systemet. De kända variablerna är som nämndes tidigare den svenska marknaden och telekommunikationsbranschen.

5.3.2. Branschutveckling - Vilka branscher?

Nya branscher i Sverige som bedöms kan ha behov av ett generaliserat kundvårdssystem KVS är identifierade nedan:

1. Finansbolag
   - Banker
   - Försäkringsbolag
   - Posten
2. Media
   - Kabel TV-bolag
   - TV via Internet
   - Förlag och tidningar
3. Transport
   - Resebolag
   - Flygbolag
   - Tågbolag
   - Bussbolag
   - Taxibolag
4. Handels- och industriföretag
   - Elbolag
   - Livsmedels- och klädeskedjor
   - Bensinbolag
   - Postorderföretag
     - Bok/Musik/Video klubbar
     - Kläder och kosmetika
     - Internethandel
5. Andra företag med kundklubbar
6. Hotellkedjor

Det finns väldigt många leverantörer av kundvårdssystem i Sverige som kan användas inom de ovan beskrivna branscherna. Däremot finns det endast ett fåtal företag som utvecklar kundvårdssystem för användning inom telekommunikationsbranschen på den internationella marknaden. Därför valdes Landsutveckling, dvs. nya länder och nuvarande bransch som segmenteringsgrund vid första marknadsindelningen.

5.3.3. Landsutveckling - Vilka länder?

Detta avsnitt beskriver hur den internationella marknaden har segmenterats samt vilka kriterier som har varit avgörande för valet av länderna.

Kontinenter

Några avgörande faktorer för att företag i ett land skall vara lämpliga kunder för kundvårdssystem är att landet har en etablerad mobiltelefonimarknad och är redo för innovationer. Om nyetablering av mobiltelefoni krävs, faller idéen med kundvård eftersom det i så fall inte finns några gamla kunder att upprätthålla kundrelationer med. Därmed kunde flera länder uteslutas, t.ex. länderna i Afrika och i större delar av Asien.

En annan avgörande faktor är mobiltelefonkulturen i landet. Exempelvis karakteriseras USA av ett annorlunda användarbeteende än i Sverige; personen som tar emot samtalet betalar, vilket har lett till att de flesta istället har en personsökare. Det invanda beteendet bedöms svårt att förändra, varför USA valts bort.

Även Japan har valts bort då det inom loppet av examensarbetets löptid bedömts mycket svårt att få tillräckligt bra svar från japanska operatörer. Japan verkar dock vara en intressant marknad.

Vissa länder i Europa verkar väldigt lämpliga då de karakteriseras av snabb industriell tillväxt, en etablerad och mogen mobiltelefonimarknad samt hög penetration av Internet. Växande antal abonnenter, öppenhet för innovationer och teknisk utveckling, nya teletjänster och inte minst informationstillgänglighet är andra positiva faktorer.

Efter en sammanställning av den internationella marknaden beslutades att lägga fokuseringen på Europa och Australien där ett generaliserat kundvårdssystem KVS bedöms ha störst marknadspotential.

Kriterier på land

Urvalet av länder grundar sig på en marknadsanalys med hjälp av följande femton kriterier varav kriterier markerade med ”❑” har varit mest avgörande för valet av länder:

- ❑ Utvecklingsstadium
  1. Låg teknologisk utveckling
  2. Hög teknologisk utveckling
- ▪ Lagar och förordningar i landet
- ▪ Infrastruktur
- ▪ Demografi dvs. folkmängdens utsträckning
- ▪ Marknadsvillkor (tex. konkurrenskraften)
- ❑ IT-mognad
- ▪ Behov
- ▪ Företagsstruktur
  1. Statligt ägda företag
  2. Privat ägda företag
  3. Få dominerande stora företag

4. Många små företag
- Kultur och förmåga att anpassa sig
- Levnadsstandard
- Turismutveckling i landet
- Population
- Mobil penetration av populationen
- BNP-värde[13]
- Antal abonnenter av mobiltelekommunikation

Dessutom grundar sig valet på en helhetsbedömning av varje land. Utifrån denna bedömning valdes följande länder i Europa samt Australien:

Förstahandsval:

- Frankrike
- Italien
- Polen
- Spanien

Andrahandsval:

- Finland
- Norge
- Schweiz
- Storbritannien
- Tyskland
- Australien

5.3.4. Vilka företag?

Då det redan är bestämt att stanna i telekommunikationsbranschen återstår bara att välja företag. För att ta reda på vilka företag i varje land som kan bli potentiella kunder till det generaliserade kundvårdssystemet KVS gjordes en bedömning med utgångspunkt från följande kriterier:

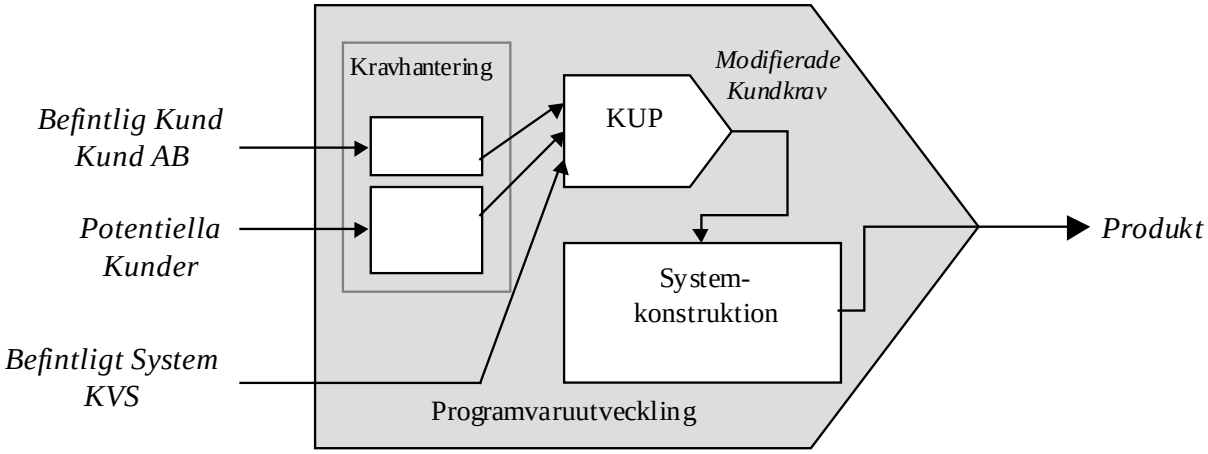
- Tillväxt och utveckling
- Etablering
- Kundkrets storleksmässigt
- Marknadsetablering
- Roamingavtal med andra
- Benägenhet att ändra och förbättra sina rutiner
- Huvudsysselsättning
- Marknadsetablering
- Struktur
- Resultat sista året
- Ägareförhållanden
- Är företagskulturen mogen för ett kundvårdssystem?

[13] Brutto National Produkt, är ett jämförande mått på länders utvecklingsgrad, *se Appendix A.*

I varje land studerades flera företag inom den mobila telekommunikationsbranschen. I första hand valdes ut de företag som hade tillräckligt stort och stabilt kundunderlag för att införandet av ett kundvårdssystem skulle anses vara behövligt. Nyetablerade företag hamnade utanför detta urvalskriteriet och valdes därmed bort. De utvalda företagen, totalt 60 stycken, analyserades. Därav valdes de 30 företag[14] som bäst uppfyllde ovannämnda kriterier ut. Dessa företag kan tänkas bli potentiella kunder till kundvårdssystemet KVS.

5.4. Programvaruutveckling tillämpad på kundvårdssystemet KVS

Utvecklingsprocessen för ett generaliserat kundvårdssystem följer riktlinjerna för en modifierad processmetod som bygger på principerna för programvaruutveckling. Detta avsnitt visar hur Programvaruutvecklingsmetoden i Kapitel 4.4.2 har tillämpats på kundvårdssystemet KVS.



• Figur 22 Tankeschema för Programvaruutvecklingsmetoden tillämpad på kundvårdssystemet KVS

In- och utparametrar

Potentiella Kunder	är de tänkbara kunder till det generaliserade kundvårdssystemet KVS som Valts med Marknadsanalys- och segmenteringsmetoden.
Befintligt System	är det befintliga kundvårdssystemet KVS som skall generaliseras.
Produkt	är det generaliserade kundvårdssystemet KVS.

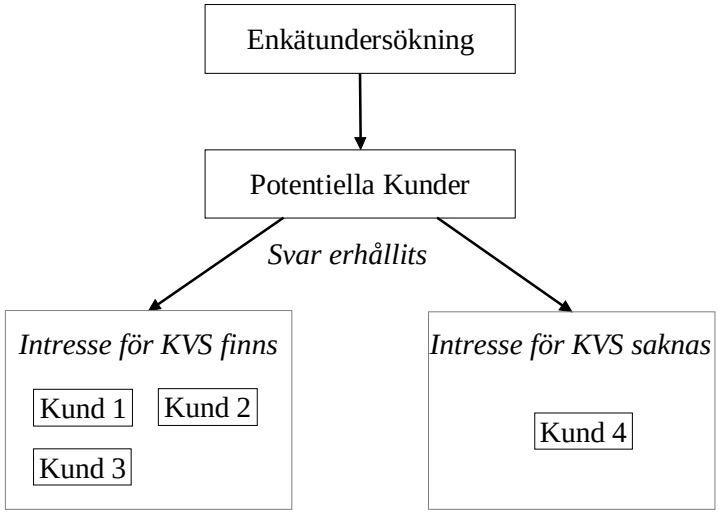
Följande delkapitel beskriver hur kravhanteringen och kravurvalsprocessen KUP har gått till och vilka resultat man fått.

5.4.1. Kravhantering

Generalisering av ett programvarusystem bygger på att nya kundkrav definieras. Med hjälp av riktlinjerna för kravhanteringsprocess i Kapitel 2.4.3 bestämdes kundkraven från de potentiella kunderna.

[14] se Appendix D Tabell D1

Insamling av kraven gjordes genom en enkätundersökning till de 30 företagen som bedömdes i Kapitel 5.3.4 vara bäst lämpade som potentiella kunder till kundvårdssystemet KVS. Utvecklingsgången följde tankeschemat i Figur 23.



• Figur 23 Tankeschema för enkätundersökning

Syftet med enkätundersökningen var att ta reda på vilket sätt det tillfrågade företaget hanterar kundrelationer idag samt om det fanns intresse för kundvårdssystem och i så fall vilka kundkrav som ställs på ett sådant system.

Genom att analysera det befintliga systemet KVS samt studera konkurrenternas tillvägagångssätt vad gäller kundvård kompletterades kunskaper om kundvårdssystem. Kundvårdssystemet KVS har analyserats genom att intervjua utvecklarna till systemet och genom att läsa all tillgängligt material och dokumentation om det. Dessutom, genom att försöka sätta sig in i användarens situation och därigenom testa systemet har en bättre bild av systemet kunnat bildas.

Enkätformuläret ”Kundkrav”[15] utformades med hänsyn till följande punkter:

- Kundorientering
- Försäljningskanaler
- Kundsegmentering
- Kundlojalitetsprogram
- Kundtjänst
- Online aktiviteter

Enkätformuläret ”Kundkrav” skickades till de utvalda företagen via email och brev. Dessutom togs telefonkontakt med flera företag för att öka svarsfrekvensen, förtydliga behovet av undersökningen och besvara eventuella frågor.

[15] Enkätformuläret ”Kundkrav” finns i Appendix C

Resultat och utvärdering av enkätundersökningen

De tillfrågade företagen var mycket ovilliga att lämna information om sitt företags tillvägagångssätt vad gäller kundhantering och kundrelationer, trots att de fick svara anonymt. Svarsfrekvensen blev därför mycket låg, endast 13% av de 30 kontaktade företagen svarade.

Ur de erhållna svaren framgick vilka kundkrav som de tillfrågade företagen var intresserade av. Resultatet av enkätundersökningen presenteras i Tabell 4.

Kundkraven g(i)	**Efterfrågepoäng per kund KK(j)**			**Efterfrågepoäng totalt**
	KK(1)	**KK(2)**	**KK(3)**	
g(1)	100	0	100	200
g(2)	70	80	100	250
g(3)	100	0	0	100
g(4)	100	0	100	200
g(5)	70	40	50	160
g(6)	70	80	100	250
g(7)	100	100	100	300
g(8)	100	100	100	300
g(9)	70	80	100	250
g(10)	0	0	100	100
g(11)	100	0	0	100
g(12)	0	100	0	100

• Tabell 4 Svarstal vid enkätundersökningen av kundkrav

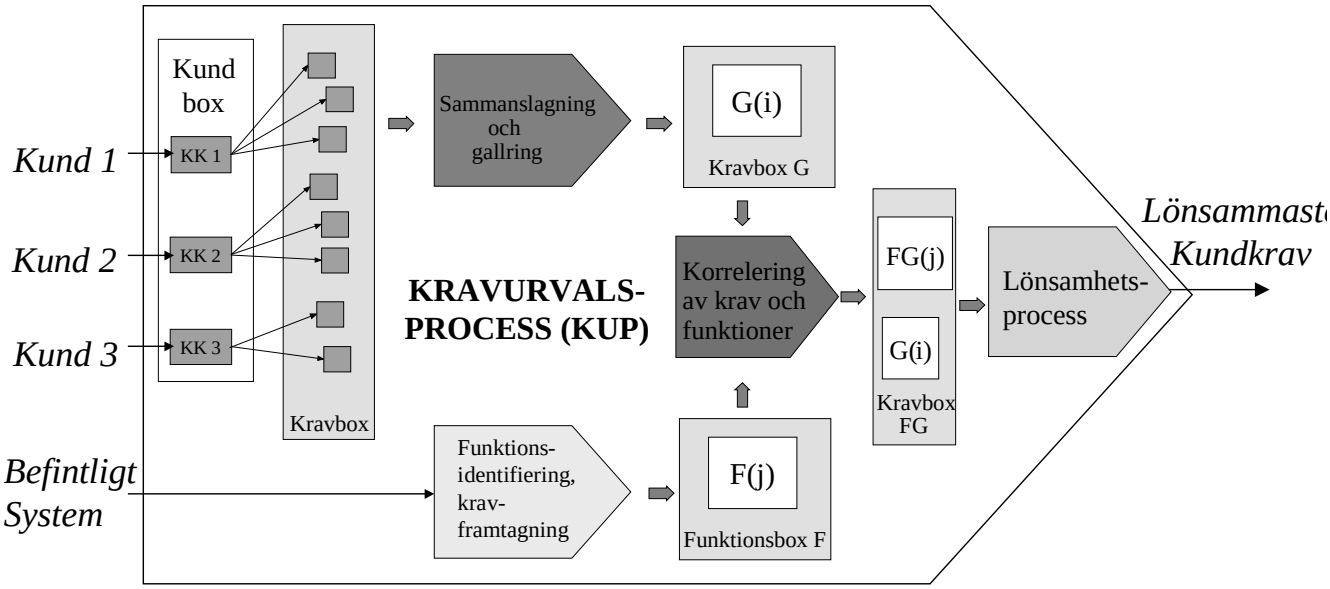
Förklaring av parametrarna i tabellen ovan:

- Kundkraven betecknas g(i) där i = 1, 2, .. , 12
- De potentiella kunderna betecknas KK(j) där j = 1, 2, 3
- De potentiella kundernas efterfrågan anges som poäng per varje kundkrav g(i). Poängen per krav ligger inom intervallet 0 – 100.
- Totalt: den sammanlagda efterfrågan i poäng per kundkrav g(i).

Resultatet av kravhanteringen är de kundkrav g(i) som ställs på ett kundvårdssystem av de potentiella kunderna KK(j).

5.5. Kravurvalsprocess KUP

Med hjälp av kravurvalsprocessen KUP har de lönsammaste kundkraven räknats ut. De lönsammaste kundkraven är ett urval av de kundkrav g(i) som tagits fram med hjälp av enkätformulären ”Kundkrav” i kapitel 5.4.1.



• Figur 24 KUP metoden

In- och utparametrar

Kundkrav	Kundkraven g(i)
Befintligt System	Kundvårdssystemet KVS
De lönsammaste kundkrav	är de mest lönsamma kraven enligt *KUP*

Kundkraven har bearbetats och resulterat i kundkraven G(i) som sedan samlats i Kravboxen G. Därefter har funktionerna F(j) ur det befintliga kundvårdssystemet KVS tagits fram och samlats i Funktionsboxen. Vidare har kundkraven G(i) och funktionerna F(j) bearbetats och resultatet har samlats i Kravboxen FG. Slutligen har de lönsammaste kundkraven beräknats.

Kravbox A

Kravbox A innehåller de kundkrav g(i) som varje potentiell kund KK(j) ställer på kundvårdssystemet KVS.

Sammanslagning och gallring

Sammanslagning av liknande funktioner har skett enligt nedanstående:

- g(4) och g(12) ger g'(4)
- g(10) och g(11) ger g'(10)

Resultat av sammanslagningen införs i Tabell 5.

Med hjälp av den totala efterfrågepoängen i Tabell 4 kan de kundkrav som haft lägst efterfrågan elimineras. Kriteriet för en elimination är att efterfrågepoängen för ett visst kundkrav inte överstiger 1/3 av de maximalt möjliga. Därmed har kundkravet g(3) eliminerats. Kundkraven med störst efterfrågan har urskiljts efter sammanslagningen och presenteras i sista kolumnen i Tabell 5.

Kundkraven g(i)	Efterfrågepoäng totalt	Kundkraven efter sammanslagning	Efterfrågepoäng totalt efter sammanslagning	Kundkraven med störst efterfrågan
g(1)	200	g(1)	200	G(1)
g(2)	250	g (2)	250	G(2)
g(3)	100	g (3)	100	
g(4)	200	g ’(4)	300	G(4)
g(5)	160	g (5)	160	G(5)
g(6)	250	g (6)	250	G(6)
g(7)	300	g (7)	300	G(7)
g(8)	300	g (8)	300	G(8)
g(9)	250	g (9)	250	G(9)
g(10)	100	g ’(10)	200	G(10)
g(11)	100			
g(12)	100			

• Tabell 5 Sammanslagnings- och gallringstabell

Kravbox G

Resultatet av processen *Sammanslagning och gallring* placeras i Tabell 6.

Kravbox G	Efterfrågepoäng totalt
G(1)	200
G(2)	250
G(4)	300
G(5)	160
G(6)	250
G(7)	300
G(8)	300
G(9)	250
G(10)	200

• Tabell 6 Kravbox G

Funktionsidentifiering

Kundvårdssystemet KVS består av sex huvudfunktioner som tillfredsställt den ursprungliga kundens krav. Varje huvudfunktion består av en eller flera mindre funktioner. Dessa huvudfunktioner har identifierats och betecknats med F(j) där j = 1, .. , 6.

Funktionsbox

De identifierade huvudfunktionerna F(j) i Kundvårdssystemet KVS har samlats och presenteras i Tabell 7.

Funktionsbox
F(1)
F(2)
F(3)
F(4)
F(5)
F(6)

• Tabell 7 Funktionsbox

Korrelering av krav och funktioner

Korreleringsprocess av krav och funktioner har genomförts och presenteras i Tabell 8.

Kundkrav G(i)		Funktioner F(j)	Kundkrav FG(j) (kan uppfyllas av funktioner)	Kundkrav G(i) (kan ej uppfyllas av funktioner)
G(1)				G(1)
G(2)				G(2)
G(4)				G(4)
G(5)				G(5)
		F(4)	FG(4)	
G(6)		F(5)	FG(5)	
G(7)		F(6)	FG(6)	G(7)
G(8)		F(2)	FG(2)	
G(9)		F(3)	FG(3)	
G(10)		F(1)	FG(1)	

• Tabell 8 Korreleringstabell

De heldragna linjerna betyder att en funktion F(j) kan uppfylla ett kundkrav G(i).

Den streckade linjen betyder att delar av en funktion F(j) kan uppfylla ett kundkrav G(i).

Kravbox FG

Kundkraven FG(j) samt kundkraven G(i) har samlats i Kravboxen FG och presenteras i Tabell 9.

FG(j)	Efterfrågepoäng totalt
FG(1)	200
FG(2)	300
FG(3)	250
FG(4)	250
FG(5)	250
FG(6)	300

G(i)	Efterfrågepoäng totalt
G(1)	200
G(2)	250
G(4)	300
G(5)	160
G(7)	300

• Tabell 9 Kravbox FG

Lönsamhetsprocess

För att beräkna vilka av kundkraven i Kravboxen FG som är de mest lönsamma gjordes en lönsamhetsberäkning utifrån en kravspecifikation och en funktionsbeskrivning. Lönsamhetsberäkningen har grundats på efterfrågan och en approximativ kostnad per kundkrav för att bygga funktioner som uppfyller kundkraven G(i) och FG(j). Dessutom har lönsamhetsberäkningen grundats på ett approximativt pris per kundkrav som kunden torde vara villig att betala.

Enkätformulär ”Kostnad och pris”

Kostnads- och prisberäkningar har gjorts med hjälp av expertuppskattningar. Ett enkätformulär ”Kostnad och pris” har tagits fram och lämnats till tre personer som arbetar med eller har arbetat med kundvårdssystemet KVS hos Leverantören AB. Denna enkäten är konfidentiell och enbart tillgänglig för AU-System AB. Samtliga respondenter[16] har besvarat enkätformuläret genom att göra en uppskattning av de efterfrågade uppgifterna. Respondenternas svar har baserats på deras tidigare erfarenheter och kunskaper om utveckling av funktioner i ett kundvårdssystem.

I enkätformuläret ”Kostnad och pris” har hänsyn tagits till följande faktorer:

- För varje kostnad/pris har en uppskattning på maxvärdet, minimivärdet samt ett uppskattat verkligt värde gjorts. Ett medeltal grundat på dessa värden har sedan beräknats.

- Grundkostnad är den kostnad som uppstår allra första gången man utvecklar en funktion som uppfyller ett visst kundkrav.

- Anpassningskostnad, är den kostnad som uppkommer när man bygger en funktion som uppfyller ett visst kundkrav för andra eller tredje gången. Anpassningskostnaden är oftast mindre än grundkostnaden då hela eller delar av den tidigare byggda funktionen kan återanvändas.

- TidK är tiden räknat i mantimmar för att tillverka en funktion som kan uppfylla ett visst kundkrav.

- MedelK är en medelkostnad per timme. Hänsyn har då tagits till att när kunskapen och/eller arbetskraften inom företaget Leverantör AB inte är tillräcklig, måste hjälp hämtas utifrån.

- FK är den fasta kostnad som utvecklingen av en funktion som uppfyller ett visst kundkrav medför. Exempelvis: verktyg, lokaler, förvaltning, Internet anslutning.

- PeriodK är den kostnad som krävs för att utveckla en funktion som uppfyller ett visst kundkrav per tidsperiod som täcker exempelvis: drift och underhåll, tillfälliga lokaler, utvecklingsmiljö.

De tre respondenternas svar sammanställdes och ett medelvärde av de ovanstående faktorer beräknades på varje kundkrav G(i) och FG(j).

Vidare beräknades det totala maxvärdet, minimivärdet samt verkligt värde av kostnad och pris för varje kundkrav med hänsyn till de potentiella kunderna som köpare av det generaliserade kundvårdssystemet KVS. Med hjälp av en enkel formel kunde lönsamhetsberäkningen göras.

[16] en uppgiftslämnare, dvs. den person som kan ge de önskade uppgifterna

I Tabell 10 exemplifieras lönsamhetsberäkningsmetoden på några fiktiva värden. Lönsamhetstabellen har prövats på många olika värden för att verifiera att beräkningsformlerna är rimliga och kan användas. Verkliga beräkningar och värden finns endast tillgängliga för AU-System AB.

Kund krav	Max Gki	Min Gki	Trolig Gki	Max SKi	Min SKi	Trolig SKi	Antal Ai	Max Tki	Min Tki	Trolig Tki	Max Fprisi	Min FPrisi	Trolig FPrisi	Max Rprisi	Min Rprisi	Trolig Rprisi
FG(101)	0	0	0	20	5	10	10	200	50	100	100	60	80	80	40	70
FG(102)	0	0	0	30	15	20	30	900	450	600	100	60	80	80	40	70
FG(103)	0	0	0	20	5	12	40	800	200	480	100	60	80	80	40	70
G(101)	50	30	45	30	15	20	30	920	465	625	100	60	80	80	40	70
G(102)	60	30	40	30	10	20	20	630	220	420	100	60	80	80	40	70
G(103)	40	15	25	15	5	10	10	175	60	115	100	60	80	80	40	70

Kund krav	Totalt Max Fprisi	Totalt Min Fprisi	Totalt Trolig Fprisi	Totalt Max Rprisi	Totalt Min Rprisi	Totalt Trolig Rprisi	Max TVFi	Min TVFi	Trolig TVFi	Max TVRi	Min TVRi	Trolig TVRi	Medel TV	Kund krav Rang		Kritiskt Fprisi
FG(101)	1000	600	800	800	400	700	950	400	700	750	200	600	600	**FG(103)**		18
FG(102)	3000	1800	2400	2400	1200	2100	2550	900	1800	1950	300	1500	1500	**FG(102)**		29
FG(103)	4000	2400	3200	3200	1600	2800	3800	1600	2720	3000	800	2320	2373	**G(101)**		20
G(101)	3000	1800	2400	2400	1200	2100	2535	880	1775	1935	280	1475	1480	**G(102)**		31
G(102)	2000	1200	1600	1600	800	1400	1780	570	1180	1380	170	980	1010	**FG(101)**		32
G(103)	1000	600	800	800	400	700	940	425	685	740	225	585	600	**G(103)**		18

• Tabell 10 Lönsamhetsberäkningstabell - exempel

Beteckningarna i Tabell 10 förklaras nedan.

Max — Det approximativa maxvärdet som kan antas.

Min — Det approximativa minimivärdet som kan antas.

Troligt — Det värde som förväntas ligga närmast det verkliga.

GK_i — Grundkostnad för utveckling av en funktion som uppfyller kundkravet ”i”.

SK_i — Anpassningskostnad för utveckling av en funktion som uppfyller kundkravet ”i”.

A_i — De potentiella kundernas totala behov av kundkravet ”i”, dvs. det antal kunder som förväntas köpa en funktion i det generaliserade kundvårdssystemet som uppfyller kundkrav ”i”.

TK_i — $TK_i = GK_i + (SK_i * (A_i - 1))$, dvs. total kostnad för utveckling av en funktion som uppfyller kundkravet ”i”.

$Fpris_i$ — Fast intäkt för utveckling av en funktion som uppfyller kundkravet ”i”. Intäkten följer konceptet ”Fast pris till fast tid”, dvs. ett fast pris och en fast tid då produkten skall levereras bestämmes.

$Rpris_i$ — Rörlig intäkt för utveckling av en funktion som uppfyller kundkravet ”i”. Intäkten följer konceptet ”kunden betalar allteftersom funktionen som uppfyller kundkravet ”i” utvecklas”.

Max TVF_i — Max TVF_i = Max $Fpris_i$ – Min TK_i , dvs. den största vinst som kan fås om antal A_i kunder köper ett system som kan uppfylla kundkravet ”i”.

Min TVF_i — Min TVF_i = Min $Fpris_i$ – Max TK_i , dvs. den minsta vinst som kan fås om antal A_i kunder köper ett system som kan uppfylla kundkravet ”i”.

Verklig TVF_i — Verklig TVF_i = Verkligt $Fpris_i$ – Verklig TK_i , dvs. den mest sannolika vinst som kan fås om antal A_i kunder köper ett system som kan uppfylla kundkravet ”i”.

Medel TV_i Medelvinsten.

Kundkrav Rang Rangordnade kundkraven.

Kritiskt Fpris = $(GK_i + (A_i - 1) * SK_i) / A_i$, ger ett nollresultat, dvs. det kritiska priset per kundkrav "i" och per kund som måste tas för att inte gå med förlust.

Resultatet av Lönsamhetstabellen enligt exemplet ovan är de sex kundkravens lönsamhet. De lönsammaste kundkraven rankas och sammanställs i Tabell 11.

Mest lönsamma					Minst lönsamma
1	2	3	4	5	6
FG(103)	FG(102)	G(101)	G(102)	FG(101)	G(103)

• Tabell 11 Rangordning av kundkraven i exemplet efter lönsamheten

Kundkraven FG(j) är mycket billigare att uppfylla eftersom det är möjligt att återanvända funktioner från det befintliga systemet som tillgodoser dessa krav. Är dessutom efterfrågan stor på kundkraven FG(j) kan utveckling av ett system som uppfyller dessa vara mycket lönsamt. Är däremot efterfrågan liten kommer som i exemplet ovan FG(101) inte vara ett lönsamt kundkrav.

Det är också viktigt att ta hänsyn till de funktioner som måste utvecklas för att ett kundvårdssystem skall fungera i sin helhet. Tex. måste ett användargränssnitt skapas även om kunden inte haft några direkta krav på dess funktionalitet. Detta innebär att i vissa fall måste även mindre lönsamma kundkrav uppfyllas av det generaliserade systemet för att den färdiga produkten skall fungera som tänkt. Leverantören måste göra en bedömning innan han utesluter några krav.

På samma sätt som i exemplet har en lönsamhetstabell gjorts för de riktiga Det riktiga resultatet av lönsamhetsberäkning i fallstudien är följande rangordning av kundkraven:

Mest lönsamma									Minst lönsamma
1	2	3	4	5	6	7	8	9	10
FG(1)	FG(5)	FG(6)	FG(2)	G(2)	G(1)	FG(4)	FG(3)	G(4)	G(5)

• Tabell 12 Rangordning av kundkraven i fallstudien efter lönsamheten, gråmarkerad ryms ej inom budgetramen.

Genom ett antagande att Leverantören AB avser tillsätta en budget på 70% av den totala utvecklingskostnaden för det befintliga kundvårdssystemet KVS till generaliseringen, faller de två minst lönsamma kundkraven utanför budgetramen. De mest lönsamma kundkraven som därmed erhållits genom lönsamhetsprocessen är: FG(1), FG(5), FG(6), FG(2), G(2), G(1), FG(4) och FG(3).

5.6. Resultat av fallstudien

Resultatet av fallstudien är potentiella kunder till kundvårdssystemet KVS samt lönsamma kundkrav. De potentiella kunderna är företag i länder där kundvårdssystemet KVS har störst marknadspotential. De valda länderna är följande: Finland, Frankrike, Italien, Nederländerna, Norge, Polen, Schweiz, Spanien, Storbritannien, Tyskland samt Australien. I varje land valdes ett till fyra företag inom telekommunikationsbranschen. Totalt valdes 30 företag som potentiella kunder till kundvårdssystemet KVS. Information om vilka företag som valdes finns endast tillgänglig för AU-System AB.

Insamling av kundkrav via enkäter gav ett dåligt resultat. Enbart fyra av de 30 tillfrågade potentiella kunder har varit villiga att lämna någon relevant information om behovet av ett kundvårdssystem samt vilka krav som de vill att ett sådant system skall tillgodose. Totalt har 12 kundkrav erhållits. Av dessa kundkrav kunde två uppfyllas av en och samma funktion och därmed sammanslåss till ett kundkrav. Av de tio kundkrav kvarvarande efter sammanslagningen kan fem uppfyllas genom återanvändning av funktioner i det befintliga kundvårdssystemet KVS. Resterande fem kräver nyutveckling av funktioner som specifikt kan tillgodose dessa kundkraven.

De lönsamma kundkraven valdes från de krav som efterfrågats mest av de potentiella kunderna. Hänsyn har dessutom tagits till utvecklingskostnaden för funktioner som kan tillgodose de utvalda kundkraven samt det pris som kunden kan tänkas vara villig att betala för dessa funktioner. Alla kundkraven har klassificerats som funktionella enligt följande undergrupper: kundadministration, statistiska beräkningar, avtalsbearbetning, prognosberäkning.

Resultatet i lönsamhetsprocessen är de åtta lönsammaste kundkraven rangordnade efter lönsamheten i Tabell 13.

Mest lönsamma							Minst lönsamma
1	2	3	4	5	6	7	8
FG(1)	FG(5)	FG(6)	FG(2)	G(2)	G(1)	FG(4)	FG(3)

• Tabell 13 De lönsammaste kundkraven i fallstudien

Faktiska värdena under fallstudien av kundvårdssystemet KVS byggt av företaget Leverantör AB åt kunden Kund AB finns endast tillgängliga för AU-System AB.

För att generalisera kundvårdssystemet KVS krävs nyutveckling av två funktioner som kan uppfylla de insamlade kundkraven G(2) och G(1) samt att återanvända funktionerna som F(1)-F(6) som kan tillgodose kundkraven FG(1) – FG(6) ur det befintliga systemet. Att realisera de åtta lönsammaste kundkraven bedöms kosta c:a 60% av utvecklingskostnaden som redan har lagts ner på det befintliga systemet. Utvecklingskostnaden uppskattades genom information från ägare till systemet.

Tidsåtgång

Den totala tiden att realisera ett generaliserat kundvårdssystem KVS kn grovt uppskattas till mellan 4000 till 7000 mantimmar. Insamling av kundkraven från de potentiella kunderna beräknas till c:a 1000 mantimmar. Tid att genomföra kravurvalsprocessen beräknas till c:a 80 mantimmar. Tidsåtgången beror på flera faktorer såsom kompetensen och resurserna hos Leverantören AB, möjliga budgeten, projektgruppens samarbete med mera. Därför är ovanstående tidsåtgång endast en grov uppskattning baserat på de data som fanns tillgängliga vid beräkningstillfället.

6. Slutsatser

I detta kapitel diskuteras resultatet av examensarbetet samt dras slutsatser. Det innehåller dessutom en utvärdering av rapporten och upplysningar som kan vara till praktisk nytta för en eventuell vidare forskning inom området.

Detta examensarbete har resulterat i en Generaliseringsmetod som utifrån ett programvarusystem, ursprungligen byggt åt en specifik kund, ger riktlinjer för hur detta system kan generaliseras till en bredare marknad. Ett av syftena med att generalisera ett befintligt programvarusystem är att vid utveckling av nya liknande system minska arbetsbördan samt öka kostnadseffektiviteten. Genom att följa Generaliseringsmetoden beskriven i denna rapport är det möjligt att med en modifierad kravhantering samt återanvändning av det befintliga programvarusystemets delar uppnå dessa syften.

Generaliseringsmetoden har verifierats genom en fallstudie av ett kundvårdssystem inom mobila telekommunikationsbranschen. Resultatet av fallstudien är dels en sammanställning av de lönsammaste kundkraven som kan ligga till grund för vidare utveckling av ett generaliserat kundvårdssystem och dels potentiella kunder till systemet.

Generaliseringsmetoden är testad enbart på ett kundvårdssystem inom mobila telekommunikationsbranschen. Det finns dock inget som utesluter att Generaliseringsmetoden skulle kunna tillämpas på andra programvarusystem eftersom den inverkar på enbart en liten del av utvecklingsprocessen nämligen kravhantering. Kravhantering handlar bland annat om att analysera kundens behov och konkretisera kundens krav och ingår i alla programvarusystem. Därför ändras bara en liten del av utvecklingsprocessen vilket medför att flera systemdelar ur det befintliga systemet kan återanvändas och att systemarkitekturen inte behöver förändras.

En genomgång av Generaliseringsmetoden ger även en kartläggning av om det överhuvudtaget är möjligt att generalisera ett befintligt system för att täcka behoven från en större marknad. Kan kundkraven från de potentiella kunderna inte alls eller till en mycket liten del uppfyllas av det befintliga systemet är en generalisering inte meningsfull.

Fallstudien har visat att det inte går att bygga ett generaliserat system och tro att det kommer att passa alla kunder, alltså precis som Murphys lag säger: ”Om det står: en storlek passar alla, så passar den inte någon”. Varje företag är unikt med sina egna behov och problem. Därför måste ett generaliserat system anpassas efter vart och ett av de nya kundernas behov och förutsättningar. Man skulle då kunna tro att allt arbete med utveckling av ett generaliserat system har gjorts förgäves men så är inte fallet.

När ett generaliserat system anpassas till en ny kund, har kunden enligt Generaliseringsmetoden redan varit med och påverkat vilka krav som kunden funnit relevanta att uppfylla och vilka behov som systemet skall tillfredsställa. Enkätundersökningen har visat en variation i de potentiella kundernas krav. En kund kan vara nöjd med det generaliserade systemet som erbjöds medan en annan kund kanske inte har lika stora behov och därför inte vill stå för kostnader som täcker fler funktioner i systemet än de han behöver. Företaget som levererar systemet måste därför ta reda på vad kunden förväntar sig av systemet och därefter undersöka om det går att reducera systemet.

Generaliseringsmetoden är som mest tidsbesparande och därmed kostnadseffektivast när det finns liten variation i kundernas krav och när många av dessa krav kan uppfyllas av det befintliga programvarusystemet. Tiden det tar att utveckla ett generaliserat system beror även på hur komplext

och omfattande det befintliga programvarusystemet är samt på hur bred marknad systemet skall anpassas till.

Marknadsanalys och segmentering i fallstudien bygger på insamling, analys och utvärdering av material från flera informationskällor. Dessa källor kan vara mer eller mindre tillförlitliga och därför påverkat valet av de potentiella kunderna till det kundvårdssystem som generaliseras.

Då resultatet som erhållits genom enkätundersökningar inte återspeglar verkligheten kan avvikande och felaktiga svar ha erhållits på den insamlade datamängden. Resultatet av enkätundersöknigen kan ha påverkats av vem som är uppgiftslämnare, vilken befattning han/hon har i företaget samt hur uppgiftslämnaren har tolkat frågorna i enkätformulären. På grund av den låga svarsfrekvensen kan enkätundersökningen anses utgöra ett dåligt statistiskt underlag. Dessutom finns det en risk att de insamlade kundkraven inte stämmer överens med de krav som de potentiella kunderna hade angett i en verklig upphandlingsprocess där deras intressen skulle ha gynnats bättre. Enkätundersökningen visar dock tydligt att det är möjligt att få in kundkraven från potentiella kunder och att det föreligger ett intresse för ett kundvårdsystem hos dessa kunder.

6.1. Rekommendationer till fortsatt utveckling av Generaliseringsmetoden

I detta avsnitt ges övergripande rekommendationer om Generaliseringsmetoden för fortsatt forskning inom området.

Marknadsanalys och segmentering

På grund av internationaliseringen av större företag som arbetar med tjänster inom telekommunikationsbranschen kan det i framtiden vara lämpligt att vid en marknadssegmentering utgå från företag istället för från länder. Det kan vara betydelselöst i vilket land som ett visst företag har grundats och/eller har sitt huvudkontor. Däremot kan det vara viktigt att veta i vilka länder som detta företag har sin verksamhet.

Kravhantering

Vid anpassning av ett generaliserat programvarusystem till en ny kund måste den nya kundens krav prioriteras. Ett generaliserat programvarusystem uppfyller kanske fler krav än vad kunden behöver eller är beredd att betala för. Därför måste det diskuteras vilka krav som skall uppfyllas för att kunna utesluta de krav som inte behöver uppfyllas. Om kraven är väldigt många kan man prioritera dem med olika hjälpmedel som t.ex. Focal Point's Requirements Planner.
Det är utvecklat av Focal Point AB och automatiserar prioriteringsprocessen. Det följer kravhanteringen från identifiering, via prioritering och beslut till frisläppning och validering.
Focal Point AB skriver ”As product lifecycles shrink and competition intensifies, products must be developed faster and continuously fulfil the key customer requirements. Requirements Planner enables you to develop better products, faster, by identifying and focusing the development process on delivering the most valuable customer requirements”.

Enkätundersökning

Resultatet av en enkätundersökning kan förbättras genom användning av professionella hjälpmedel. Det finns en metod kallad Delfi-tekniken enligt [Lekvall & Wahlin 1993] som sysslar med hur man skall utföra undersökningar och vilka risker man bör ta hänsyn till.

En annan aspekt som bör beaktas i framtiden är det stora bortfallet. Bortfallet beror till största delen på ovilja att lämna ifrån sig uppgifter om företaget som anses konfidentiella. Det skulle eventuellt kunna åtgärdas genom att skapa en personlig kontakt med de tillfrågade personerna i företagen.

Lönsamhetsberäkning

Det finns många tekniker för att uppskatta kostnaderna för utveckling av ett programvarusystem. Fallstudien beskriver en teknik som kan användas för att uppskatta kostnaderna för utveckling av systemets. En annan teknik som kan användas är Function Point Analysis. Det är en inom IT-industrin väletablerad teknik skapad av Allan Albrecht 1979. Tekniken bygger på att utifrån en kravspecifikation eller systemets funktionalitet räkna ut systemets storlek och därmed tid, kostnad och resurser som behövs för dess utveckling. En annan metod skapades 1981 av Dr. Barry Boehm, som bland annat utvecklat Win-Win Spiralmodellen. COCOMO II (Constructive Cost Model) är en algoritmisk modell för empirisk kostnadsestimering av programvaruutveckling.

Allmänt

Det är viktigt att utvärdera det befintliga systemet för att fastställa om det överhuvudtaget är lämpligt att genomföra en generalisering av det. När en ny teknik införes måste man också vara medveten om de effekter den nya tekniken kan föra med sig.

Referenser

Referenslista trycksaker

[Anderson, 1996]	Bengt Anderson, Positioning: The Art of Being First, Sandberg & CO, Göteborg, november 1996
[Ansoff 1965]	I. Ansoff, Corporate Strategy, McGraw-Hill 1965
[Boehm 2000]	Barry Boehm, Spiral Development: Experience, Principles, and Refinements, USC Center for Software Engineering, 2000
[Boehm 1980]	Barry Boehm, Software Engineering Economics, Prentice-Hall Interna-tional, Englewood Cliffs, New Jersey, 1980
[Boehm & Bose 1994]	Barry Boehm, P. Bose, A Collaborative Spiral Software Process Model Based on Theory W, Proceedings International Conference on Software Process: IEEE CS Press, Los Alamitos, USA, 1994
[Boehm m.fl. 1994]	B. Boehm, A. Egyed, J. Kwan, D. Port, A. Shah, R. Madachy, Using the WinWin Spiral Model: A Case Study, IEEE Computer, USA, Juli 1998
[Boehm *m.fl.* 1998]	Barry Boehm, Alexander Egyed, Dan Port, Archita Shah, Julie Kwan and Ray Madachy, Using the WinWin Spiral Model: A Case Study, USC Center for Software Engineering, 1998
[Brown 1999]	Stanley A. Brown, Strategic Customer Care, John Wiley & Sons, Canada Ltd, 1999
[Bruzelius & Skärvad 1989]	Lars H. Bruzelius & Per-Hugo Skärvad, Integrerad organisationslära, Studentlitteratur, Lund 1989
[Eklund & Fernlund 1998]	Sven Eklund & Hans Fernlund, Programkonstruktion med kvalitet, Studentlitteratur, Lund 1998
[Falonius & Hedberg 1997]	Tommy Falonius & Tomas Hedberg, Spelet på marknaden, IHM Förlag AB, Göteborg 1997
[Kazman *m.fl.* 1999]	R. Kazman, M. Klein, P. Clements, Evaluating Software Architectures for Real-Time Systems, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA 1999
[Kotler *m.fl.* 1998]	P Kotler, G Armstrong, J Saunders och V Wong, Principles of Marketing, The European Edition. Prentice Hall 1998
[Lekvall & Wahlin 1993]	Per Lekvall & Clas Wahlbin, Information för marknadsföringsbeslut, IHM Förlag AB, Göteborg 1993
[Mintzberg 1988]	H. Mintzberg, Opening up the Definition of Strategy, i Quinn, J – Mintzberg, H – James R M: The Strategy Process, Prentice Hall Inc., Int Ed 1988
[Ncube 2000]	Cornelius Ncube's, A Requirements Engineering Method for COTS-Based Systems Development, City University London, 1999
[Olsson, Skärvad 1993]	Jan Olsson, Per-Hugo Skärvad, Företags ekonomi 99, Liber-Hermods Förlag AB, Malmö 1993

[Porter 1980]	M. E. Porter, Competive Strategy, The Free Press 1980
[Regnell, Wohlin 1999]	Björn Regnell & Claes Wohlin ProgramvaruUtveckling för Stora System, Projekthandledning, Telekommunikationssystem, Lund augusti 1999
[Siebel & House 1999]	Thomas M. Siebel & Pat House, Cyber Rules, Doubleday, New York 1999
[Sommerville 1996]	I. Sommerwille, Software engineering, Addison-Wesley, Wokingham, UK, 1996
[Pfleeger & Shari 1998]	Pfleeger & Lawrence Shari, Software Engineering : theory and practice, Prentice Hall Inc., New Jersey USA 1998
[Royce 1970]	W. W. Royce, Managing the development of large software systems: Concept and techniques, WESCON, 1970
[Yeh & Zave]	R. T. Yeh, P. Zave, Specifying Software Requirements. Proceedings of the IEEE, 68(9):1077–1085, 1980

Referenslista Internet

Marknadsanalys - Statistik

Worldrover	http://www.worldrover.com
Riksbank	http://www.riksbank.se
Statistiska Central Byrån	http://www.scb.se
Geographic	http://www.geographic.org
Worldbank	http://www.worldbank.org
Utrikes politiska institutet	http://www.ui.se/index.htm
Makroekonomisk teori och politik	http://www.nek.lu.se/distansmakro
Sveriges tekniska attachéer	http://www.statt.se
Attaché	http://www.attachesoftware.com

Tidskrifter på nätet

Mobil	http://www.mkf.se/
iMobile	http://www.imobile.com.au
Public Network	http://www.pnewire.com/

Analys av mobila operatörer

GSM World	http://www.gsm.org/cgi-bin/search.cgi
CIT Publications	http://www.citpubs.com/
Gartner	http://gartner11.gartnerweb.com/
Imperial Communications Consultants	http://www.imcoco.com
Idg	http://www.idg.se
Idg	http://www.idg.net/
Ovum	http://www.ovum.com/
Memory	http://www.memory.se/
Total Tele	http://www.totaltele.com/
Spara Pengar	http://www.sparapengar.com/
EMC	http://www.emc-database.com/

Tillverkare av CRM-verktyget som finns på marknaden idag

Datametrix	http://www.datametrix.se
Memory	http://www.memory.se/CRM/crm.htm
Evolve	http://www.evolvesol.com/evolve/kontakt.asp
G2 solutions	http://www.g2solutions.com/
Bollplanket	http://www.bollplanket.m.se/produkter.htm
Crm_Europe	http://www.crm-europe.com/
MSC	http://www.msc.se/
Cybernetics	http://www.cybernetics.se/
QiTEL	http://www.qitel.se/
Evolve Solutions	http://www.evolvesol.com
CRM Ltd.	http://www.crm-europe.com/

Övrig information

Focal Point	http://www.focalpoint.se/
Broad Vision	http://www.broadvision.com
Com Tracker	http://www.comtracker.com/
Siebel	http://www.siebel.com
CRM Community	http://www.crmcommunity.com/
Wapetite	http://www.wapetite.com
Cybernetics	http://www.cybernetics.se/
The CDMA Development Group	http://www.cdg.org/

Figur- och tabellförteckning

Figurförteckning

- Figur 1 Generaliseringsmetoden - en översikt 15
- Figur 2 Generaliseringsmetoden - en översikt 17
- Figur 3 Interaktion mellan kunden och leverantören av programvarusystem 18
- Figur 4 The process of analysis 19
- Figur 5 The process of synthesis 19
- Figur 6 Framtagning och analys av krav.....22
- Figur 7 Åtagandetriangel 23
- Figur 8 Vattenfallsmodellen 25
- Figur 9 V modellen 26
- Figur 10 Prototypmodellen 27
- Figur 11 Win-Win Spiralmodell.....28
- Figur 12 Generaliseringsmetoden - en översikt 29
- Figur 13 De tre ringarna.....30
- Figur 14 Generalisering av ett programvarusystem.....37
- Figur 15 Generaliseringsmetoden.....38
- Figur 16 Insignal-utsignal modell.....38
- Figur 17 Marknadsanalys- och segmentering.....40
- Figur 18 Programvaruutvecklingsmetod.....41
- Figur 19 KUP applicerad på Vattenfallsmodellen.....42
- Figur 20 Kravurvalsprocessen KUP.....43
- Figur 21 Tankeschema för segmentering av marknaden för KVS…48
- Figur 22 Tankeschema för Programvaruutvecklingsmetoden.....52
- Figur 23 Tankeschema för enkätundersökning.....53

- Figur 24 KUP metoden 55

Tabellförteckning

- Tabell 1 Produktmarknadstabell 31
- Tabell 2 Grundstrategier 31
- Tabell 3 Marknadssegmentering för kundvårdssystemet KVS 49
- Tabell 4 Svarstal vid enkätundersökningen av kundkrav.....54
- Tabell 5 Sammanslagnings- och gallringstabell 56
- Tabell 6 Kravbox G.....56
- Tabell 7 Funktionsbox 57
- Tabell 8 Korreleringstabell 57
- Tabell 9 Kravbox FG 57
- Tabell 10 Lönsamhetsberäkningstabell - exempel 59
- Tabell 11 Rangordning av kundkraven i exemplet efter lönsamheten 60
- Tabell 12 Rangordning av kundkraven i fallstudien efter lönsamheten 60
- Tabell 13 De lönsammaste kundkraven i fallstudien.....61

Terminologi

I detta avsnitt förklaras några av de grundläggande beteckningar och begrepp som används i rapporten.

Begrepp	Beskrivning
Acceptanstest	Validering av systemet gentemot kundens krav.
Användare	En användare av ett programvarusystem är en person eller en grupp människor som använder systemet.
Egenskapskrav	Hur en produkt skall fungera.
Fallstudier	Enligt [Lekvall & Wahlin 1993] betecknar termen fallstudier ”studier där intresset är riktat mot detaljerade och ofta djupgående beskrivningar och analyser av enskilda fall”.
Funktionella krav	Vad produkten eller systemet skall kunna utföra.
Funktionell programmering	Ett programmeringssätt där man delar upp ett problem i mindre bitar, moduler, och varje funktion som arbetar med dessa moduler är fristående från övriga funktioner. Tex. Lisp.
Generaliseringsmetod	En metod för generalisering av ett programvarusystem byggt åt en specifik kund till en bredare marknad.
BNP	BNP är en förkortning av Brutto National Produkt och betecknar det sammanlagda värdet av produktion av de varor och tjänster för slutlig användning som produceras i ett land under en period, vanligen ett år utan hänsyn till fördelning av inrikes och utrikes fordran. Här används BNP-måttet som jämförande mått på länders utvecklingsgrad.
KUP	Generaliseringsmetodens kravurvalsprocess. Används för att ta fram lönsammaste kundkrav utifrån de potentiella kunders krav.
Komplexitet	En funktion av de ingående delsystemen och de samband som finns mellan dessa delsystem, [Eklund & Fernlund 1998].
Kravhantering	Att samla in, prioritera samt specificera och dokumentera de kundkrav som ett programvarusystem byggs för att uppfylla.
Kvalitetsegenskaper	Egenskaper hos en programvara såsom tillförlitlighet, effektivitet, ändringsbarhet
Kund	En kund är en organisation, företag eller person som betalar för utveckling och byggnad av systemet.
Kundkrav	Kundens krav och önskemål på produkten samt kundens behov som skall tillgodoses.
Kundvård	Kundvård är ett tillvägagångssätt för ett företag för hantering av kundrelationer genom att fokusera på kundens individuella köpbeteenden och speciella behov. All relevant information om den omsättning som genereras per enskild kund lagras i en databas och används sedan för kundanalys och kundinteraktion.
Kundvårdssystem	Ett programvarusystem där all information om kunderna samlas och skapar en korrekt, enhetlig bild av företagets kontakter och transaktioner med var och en av kunderna. All kunskap om kunden blir lätt tillgänglig för alla berörda

	informationsanvändare som exempelvis försäljare och marknadsförare. Databasen uppdateras regelbundet och företaget kan hålla reda på vem den enskilda kunden är, vad kunden vill ha eller hur individanpassa nya erbjudanden till kunden.
Kundvärde	Ett värde som beräknas på varje kund relaterat till hur mycket kunden konsumerar av företagets produkter eller tjänster. Ett högt kundvärde kan betyda att kunden köper många produkter och är därmed en bra kund som företaget vill behålla.
Kundvärdering	Ett tillvägagångssätt att värdera en kund utifrån hur mycket kunden konsumerar av företagets produkter eller tjänster.
Logisk programmering	Ett programmeringssätt som kan närmast jämföras med en databas och där man själv måste definiera allting. Tex. Prolog.
Marknadsföring	Allt som görs för att sälja en vara eller en tjänst.
Marknadsanalys- och segmenteringsmetoden	Metod för analys och segmentering av en marknad. Används här för att hitta potentiella kunder till det generaliserade programvarusystemet.
Metod	Planmässigt tillvägagångssätt för att uppnå visst resultat
Metodik	Uppsättning metoder inom visst område.
Modell	Ett förenklat, åskådligt tankeschema för beskrivning av komplicerad abstrakt företeelse.
Objektorienterad programmering	Ett programmeringssätt där data och funktioner integreras till enhetliga moduler som själva definierar sina attribut och egenskaper. Tex. Jawa, C++, Simula, Smalltalk.
Procedurell programmering	Ett programmeringssätt som utgår från att en procedur utgör en algoritm medan data kapslas in i moduler eller procedurer. Tex. C, Pascal.
Programmeringsparadigm	Beskriver olika sätt att programmera: funktionell, objektorienterad, procedurell, logisk. Se vidare beskrivningar.
Programvarusystem	Ett antal mindre program som arbetar ihop och därmed bildar ett större som blir ett system.
Programvaruteknik	Verktyg och metoder som studeras och sedan används vid konstruktion av program och system.
Programvaruutvecklingsmetoden	Ett processförlopp för utveckling av ett generaliserat programvarusystem.
Time-to-market	Hur lång tid det tar innan en produkt eller tjänst kan lanseras på marknaden.
Top-down design	Börjar med helheten och bryter ner den till mindre bitar.
Top-up design	Börjar med mindre bitar som sätts ihop för att bilda en helhet.
Utvecklingsprocess	Ett tillvägagångssätt eller strategi för att på ett kontrollerat och organiserat sätt utveckla ett programvarusystem.
Validering	Säkerställa att rätt produkt har utvecklats.
Win-Win teorin	En managementteori som behandlar problemet med många olika intressenter med motstridiga mål och ger riktlinjer för hur dessa mål löses på ett effektivt sätt.
Verifiering	Säkerställa att en produkt är korrekt utvecklad.