

Handelshögskolan vid
Göteborgs Universitet
Institutionen för informatik

Acceptanstestning - en jämförelse mellan teori och praktik

Karolina Magnusson

Examensarbete II, IA7300, 10 poäng

Höstterminen 1999

Handledare:

Birgitta Ahlbom (Institutionen för informatik)
och Anders Söderberg (PreEra Konsult AB)

Abstrakt

Acceptanstestning är en viktig del i systemutvecklingens sista fas, testfasen. Det är denna testning, som utförs av kunden, som godkänner eller underkänner systemet. För att skapa en förståelse för acceptanstestning hos kunden har generella riktlinjer beskrivits. Utifrån dessa har en jämförelse gjorts med ett praktikfall, för att utreda om teorin efterföljs i praktiken. Resultatet, som grundar sig på litteraturstudier, en fallstudie och intervjuer, visar att praktiken följer teorin på många punkter, men att den även går sin egen väg, där det faller sig mer naturligt. Det viktigaste som kom fram, är att man inte kan följa de generella riktlinjer som finns till punkt och pricka, utan dessa måste anpassas till varje unikt projekt.

1 Inledning	3
1.1 Problemställning	3
1.2 Syfte	3
1.3 Avgränsningar.....	4
1.4 Metod.....	4
2 Acceptanstestning	5
2.1 Vad är acceptanstestning?	5
3 Genomförande av acceptanstestning	7
3.1 Testfacit byggs upp.....	7
3.1.1 Kravspecifikationen	7
3.2 Testplanen	8
3.2.1 Testbeskrivning	9
3.2.2 Testfall	10
- 3.2.2.1 Utformningen av testfall	10
3.2.3 Testprocedur	11
3.2.4 Testprotokoll och testrapport	12
3.3 Hur stor del av systemet skall man testa?.....	12
3.3.1 Dokumentation.....	12
3.3.2 Riskanalys	13
3.4 Hur länge skall man testa?	13
3.5 Vem skall testa?.....	14
3.5.1 Organisation	14
3.5.2 Utbildning	14
3.6 Uppsättning av testmiljö	14
3.7 Felrapportera	15
3.8 Slutrapportera.....	15
4 Bakgrund till praktikfall	17
4.1 Kund	17
4.2 Utvecklingsprojektet	17
4.2.1 KAS-A.....	17
4.2.2 Dukas.....	18
4.2.3 HIPNET	18
4.2.4 Konvertering	18
4.2.5 Leveransgodkännande.....	19
5 Genomförande av acceptanstestning av HIPNET-anpassningar	21
5.1 Testfacit byggs upp.....	21
5.1.1 Kravspecifikationen	21
- 5.1.1.1 Dokas	21
- 5.1.1.2 Dukas	21
- 5.1.1.3 KAS-F och KAS-A/KAS-F (gamla Prokas-registren)	22
- 5.1.1.4 KAS-A	23
5.2 Testplanen	24
5.2.1 Bakgrund.....	24
5.2.2 Övergripande tidplan	24

5.2.3 Förberedelser	24
5.2.4 Deltagare	24
5.2.5 Genomförande av tester	25
6 Jämförelse och resultat	27
6.1 Testfacit byggs upp.....	27
6.1.1 Kravspecifikationen	27
6.2 Testplanen	28
6.2.1 Testbeskrivning	28
6.2.2 Testfall	28
- 6.2.2.1 Utformningen av testfall	28
6.2.3 Testprocedur.....	29
6.2.4 Testprotokoll och testrapport.....	29
6.3 Hur stor del av systemet skall man testa?.....	29
6.3.1 Dokumentationen.....	29
6.3.2 Riskanalys	30
6.4 Hur länge skall man testa?.....	30
6.5 Vem skall testa?.....	30
6.5.1 Organisation	30
6.5.2 Utbildning	30
6.6 Uppsättning av testmiljö	31
6.7 Felrapportera	31
6.8 Slutrapportera.....	31
6.9 Sammanfattning av jämförelse	32
7 Diskussion.....	35
8 Källförteckning	37
8.1 Böcker.....	37
8.2 Elektroniska dokument	37
8.3 Opublicerat material.....	37
8.4 Enkätfrågor	38

Bilagor

Bilaga 1 – Exempel på mall för testplan

Bilaga 2 – Exempel på mall för testbeskrivning med testfall

Bilaga 3 – Exempel på mall för testrapport

Bilaga 4 – Exempel på mall för slutrapport

Bilaga 5 – Intervjufrågor

Bilaga 6a – Intervjusvar, Elisabeth Franklin

Bilaga 6b – Intervjusvar, Tommy Harström

Bilaga 7 – Funktionslista/Testprotokoll

Bilaga 8 – Felrapport

1 Inledning

Att sätta i drift ett system, som inte är testat, kan orsaka stora problem. Systemet har redan kostat mycket att utveckla och kommer att kosta ännu mer, om det driftsätts utan att kanske fungera som det skall. Det kan komma att innebära förlorad arbetstid och inkomst, då det kan hindra verksamheten att arbeta, som den brukar. Systemet kommer att stjälpa företaget istället för att hjälpa det. Genom att kunden/beställaren tänker igenom sina krav och acceptanstestar systemet gentemot dessa krav innan det sätts i drift, kan både tid och pengar sparas. Om fel och brister kan upptäckas i ett tidigt skede, kan kostnader för korrigeringar minskas betydligt.

Förhoppningen med detta arbete är, att läsaren skall få en grundläggande kunskap om acceptanstestning, så att vikten av dessa tester inte underskattas vid systemutveckling och för att underlätta förberedelsearbetet vid sådana tester. Arbetet innehåller även en utredning av hur acceptanstester ”bör” gå till enligt teorin, jämfört med hur de går till på ett företag i Göteborg.

1.1 Problemställning

Detta examensarbete beskriver vad acceptanstestning är och riktlinjer för hur dessa tester kan utföras. Arbetet behandlar också frågor som:

- Vad är en testplan?
- Hur är testplanen uppbyggd?
- Hur stor del av ett system bör man testa?
- Hur länge skall man testa?
- Vem skall testa?
- Hur bygger man upp en testmiljö?

En redogörelse görs av bakgrunden till praktikfallet med kund- och uppdragsbeskrivning. Jag ställer mig därefter frågan:

- Hur går acceptanstestningen till på företaget Telia InfoMedia TeleVision jämfört med de riktlinjer som tas upp?

1.2 Syfte

Syftet med detta arbete är att öka förståelsen för acceptanstester hos beställaren/kunden och slutanvändarna av ett system. Arbetet har också som syfte att jämföra hur sådana tester går till i verkligheten i förhållande till teorin.

1.3 Avgränsningar

Acceptanstester är en del av systemutvecklingens testfas. Detta arbete tar dock inte upp några andra tester och inte heller några andra systemutvecklingsfaser.

Vid acceptanstestning kan automatiserade verktyg användas för att genomföra testerna. Dessa verktyg kommer inte att beskrivas i detta arbete, då acceptanstesterna på Telia InfoMedia TeleVision inte använder sig av sådana.

Avsnitten som behandlar praktikfallet på Telia InfoMedia TeleVision, innehåller namn på system och andra fackuttryck som används på företaget. Dessa beskrivs inte, då de inte har någon betydelse för syftet med det här arbetet.

Syftet med detta arbete är heller inte att utreda hur testresultaten på Telia InfoMedia TeleVision utföll. Det kommer således inte att beskrivas, varför systemen i slutändan blev godkända eller underkända.

1.4 Metod

För att uppnå syftet med arbetet har diverse material studerats. Materialet har bestått av böcker, internetdokument och andra opublicerade dokument från PreEra Konsult AB och från Telia InfoMedia TeleVision AB, vilka handlar om acceptanstestning och systemutveckling i allmänhet.

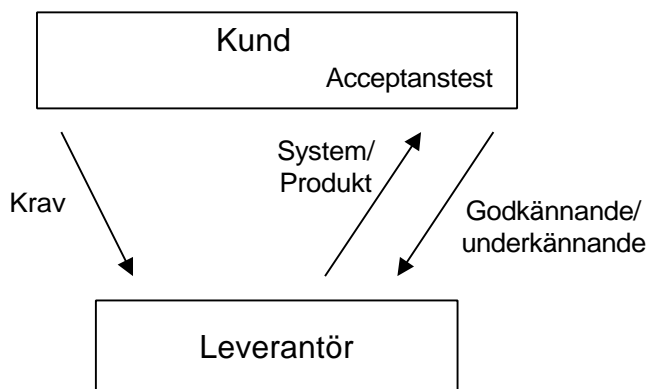
En undersökning har gjorts för att utreda om praktiken stämmer överens med teorin inom acceptanstestning. Denna undersökning gjordes genom en jämförelse mellan projektet på Telia InfoMedia TeleVision och teorin, tagen från studerad litteratur. En enkät skickades även ut till två medlemmar i testgruppen, för att få information om själva testerna.

2 Acceptanstestning

2.1 Vad är acceptanstestning?

Acceptanstestning är den del av testfasen i systemutvecklingen, där kunden godkänner eller underkänner det utvecklade systemet. Systemet skall testas av kunden med hjälp av leverantören, helst i den miljö där systemet sedan skall sättas i drift, för att se om det motsvarar de krav eller specifikationer som identifierar produkten. Även handböcker och dokumentation som ingår i systemet skall testas. För mer utförligt information om vem som testar systemet, se Kapitel 3.5 ”Vem skall testa?”.

Man kan säga att acceptanstestning är en slags leveranskontroll (se Figur 1). Om kunden godkänner systemet och fel uppkommer vid senare tillfälle, när systemet har satts i drift, får kunden själv stå för kostnader för att åtgärda felet.¹



Figur 1 Acceptanstestning – en leveranskontroll²

Trots att stor vikt läggs vid design och utveckling av systemet är det vanligt att fel ”slinker in” eller att krav inte blir uppfyllda. Syftet med acceptanstestningen är att exekvera systemet med mål att hitta dessa fel och upptäcka bortglömda krav. Eftersom människan omedvetet strävar efter att uppfylla ett fastställt mål, är det viktigt att ha som mål med testningen att hitta fel. Om målet hade varit att visa att programmet är korrekt, så hade människan visat att programmet var korrekt, trots att det innehöll fel. Ett antal ”snälla” testfall hade konstruerats som bara övergripande testade programmet. Men om vi däremot har som mål att hitta fel och att få programmet att spåra ur, så måste vi hitta på så många svåra testfall som möjligt, för att uppnå detta mål. Eftersom felaktigheter ändå kommer att uppträda någon gång, är det

¹ Cem Kaner, Jack Falk och Hung Quoc Nguyen, *Testing Computer Software* (United States of America: Van Nostrand Reinhold, 1993), 316

² Björn Birk, Patrik Olnäs och Magnus Penker, *A UML process – the Astrakan Approach* (Sverige: Astrakan, 1998), 169

bättre att hitta dem innan systemet implementeras och börjar användas hos kunden. Om systemet blir underkänt vid acceptanstestningen, d.v.s. om fel hittas, kan kunden godkänna det senare när felet/felen har åtgärdats och nya tester har genomförts.^{3 4}

³ Sven Eklund och Hans Fernlund, *Programkonstruktion med kvalitet – projekthantering och ISO 9000* (Lund: Studentlitteratur, 1998), 187

⁴ Birk, Olnäs och Penker, 169-170

3 Genomförande av acceptanstestning

För att testningen skall lyckas måste man följa vissa principer vid utformandet av testerna. Detta arbete underlättas genom att man följer de dokument, som har producerats under designfasen i systemutvecklingen och som också utgör facit för testerna.

3.1 Testfacit byggs upp

I den första fasen i systemutvecklingen, definitionsfasen, görs en analys av kundens problem. Ett kontrakt skrivs och kundens krav på produkten specificeras. I detta skede definieras även projektgruppens sammansättning och dessutom fastställs budget, tidsplan och andra rutiner.

Definitionsfasen omfattar därmed två huvudaktiviteter: problemanalys och projektplanering. I problemanalysen formuleras exakt vad som skall produceras och levereras och därmed kan man i projektplaneringen fastställa projektets resurser gällande tid, personal, budget mm. Dessa punkter fastställs i två grundstolpar: kravspecifikationen och projektplanen. Eftersom kravspecifikationen ligger till grund för acceptanstestningen kommer följande kapitel att behandla den djupare.

3.1.1 Kravspecifikationen⁵

Målet med kravspecifikationen är att få fram de krav, som systemet skall byggas efter. Dessa krav kan fastställas på tre olika sätt:

- *Definierade av kunden.* Renodlad är denna metod mycket ovanlig, då det ställer höga krav på kundens kunskap om systemutveckling. Den används dock då kunden har mycket höga krav på programvaran och då dessa inte går att förhandla fram.
- *Definierade av leverantören.* Denna princip används då man utvecklar generella programvaror, vilka kommer att användas av en stor mängd olika användare. Ofta kan leverantören dock göra en marknadsundersökning för att se vad användarna vill ha, men det är ändå leverantören som fastställer kraven i kravspecifikationen.
- *Framtagna i dialog.* Denna princip är den som bör eftersträvas, då en dialog förs mellan kund och leverantör. Att diskutera fram kraven gör att missförstånd upptäcks på ett tidigt stadium och kraven blir lättare att förstå för både kund och leverantör.

När alla krav är framtagna, delas de ofta in i olika nivåer, för att utreda vilka krav, som är viktiga och vilka som bara ”vore bra att ha”. De delas även in i olika grupper, beroende på vad för slags krav det är. Dessa grupper kan till exempel vara funktionella krav, som beskriver de tjänster som programvaran skall utföra för användaren, icke-funktionella krav, som beskriver de villkor och restriktioner under vilka de funktionella kraven måste uppfyllas samt krav på användargränssnittet, som beskriver hur programvaran skall se ut.

⁵ Eklund och Fernlund, 97-104

En kravspecifikation är det viktigaste dokumentet, som tas fram under ett projekt, eftersom det ligger till grund för hela projektet. Den kan kännetecknas av följande punkter:

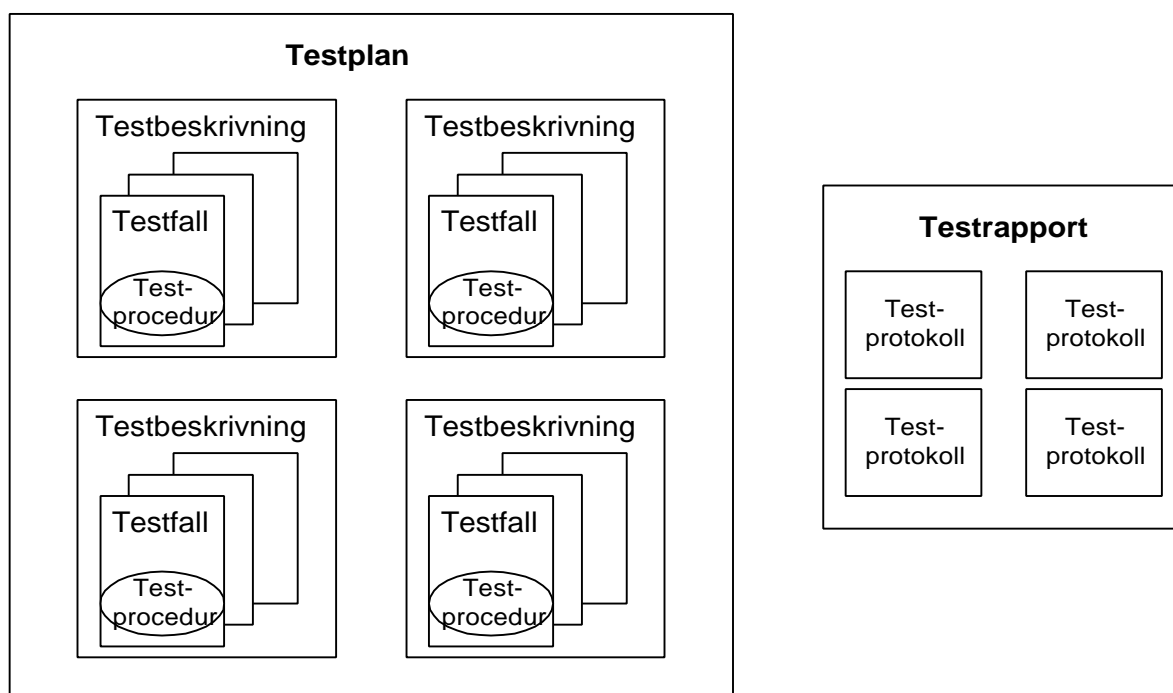
- Kravspecifikationen skall inte vara något designdokument. Den skall inte säga något om *hur* problemen skall lösas, utan bara *vad* det är som skall lösas.
- Allt som programvaran skall kunna utföra, skall specificeras i kravspecifikationen.
- Kraven skall vara skrivna på ett sådant sätt, att man kan jämföra dem med den färdiga produkten.
- Inga krav får stå i konflikt med varandra.

3.2 Testplanen ⁶

Då utvecklingen av systemet är färdig, är det dags att utvärdera, hur utgången blev jämfört med det planerade resultatet. Detta görs genom att testa mot de önskemål/krav, som ställts upp i kravspecifikationen. Kravspecifikationen används inte direkt som den är, för att utföra testerna, utan utifrån denna härleds istället en så kallad testplan, som skall ingå i projektplanen. Testplanen utformas av projektledaren i samarbete med systemutvecklare och testpersoner. Det är en övergripande plan, för hur all testning skall utföras. Här beskrivs vilka metoder som skall användas, hur lång tid testningen får ta, hur mycket pengar och andra resurser som är avsatta o.s.v. Denna övergripande plan räcker dock inte, utan den måste utvecklas och göras mer konkret angående det som skall testas. Detta görs vanligtvis av testpersonerna. Bilaga 1 innehåller ett exempel på hur en mall för en testplan kan se ut.

Testplanen för acceptanstestningen skrivs samtidigt som kravspecifikationen utformas. Detta innebär att testplanen skrivs långt innan kodningen av produkten påbörjas. Strukturen av en testplan visas i Figur 2.

⁶ Ibid., 192-193



Figur 2 Testplan och testrapport ⁷

Enligt Figur 2 består testplanen av ett antal testbeskrivningar, som var och en är orienterad kring ett speciellt objekt, t.ex. ett krav i kravspecifikationen. Ett krav kan också testas på flera olika sätt, d.v.s. i flera olika testfall. Under testningens gång kommer varje testbeskrivning att ge upphov till ett testprotokoll. Testprotokollet beskriver vad testfallen har för utfall. Alla testprotokollen sammanställs sedan till en testrapport.

Testplanen bör utformas som ett strukturerat och lätthanterligt dokument. Eklund och Fernlund föreslår i sin bok "Programkonstruktion med kvalitet – projekthantering och ISO 9000", att en testplan skall ha följande innehåll:

- Namn/identifierare på testplanen.
- Introduktion till systemet, bakomliggande behov och vad systemet skall göra.
- Testobjekten, d.v.s. vilka krav systemet skall uppfylla och vilka av dem som skall testas. Här skall även referenser till motsvarande testbeskrivningar ges.
- Tid- och resursplan, d.v.s. hur mycket projekttid, antal personer, datorer mm. som är avsatta för testningen, samt när testningen skall utföras.
- Ansvarig för testplanens genomförande.

3.2.1 Testbeskrivning ⁸

En testbeskrivning beskriver hur varje enskilt krav i kravspecifikationen skall testas. Testbeskrivningen skrivs av den testansvarige, för varje specifikt krav, och bör innehålla följande punkter:

⁷ Ibid., 192

⁸ Ibid., 193

- Identifierare av testbeskrivningen.
- Beskrivning av vilket objekt/krav som testbeskrivningen avser.
- Tillvägagångssätt, d.v.s. hur testet skall gå till och vilka resurser som finns tillgängliga för testet.
- Översikt över alla testfall och referenser till dessa.
- Kriterium för att objektet/kravet är färdigtestat.
Det är mycket svårt att specificera när testningen av ett krav är klar. Eftersom målet med testerna är att hitta fel, kan man inte säga att testningen är klar när alla tester utförts, utan att man har hittat något fel. Därför är det mycket bättre att ha ett negativt kriterium, som t.ex. att man är klar, när minst 2 fel per 100 rader källkod har hittats. På så sätt får man verkligen anstränga sig att skriva svåra testfall. Att veta när testningen är klar behandlas vidare i Kapitel 3.4 ”Hur länge skall man testa?”.

För ett exempel på hur en mall för en testbeskrivning kan se ut, se Bilaga 2.

3.2.2 Testfall^{9 10 11}

Varje testfall skall helst vara skrivet på ett eget dokument med in- och utdata specificerat så, att man kan återvända till testet och utföra det flera gånger. Detta är särskilt viktigt, då man återtestar efter att ett fel har åtgärdats. Man vill ju då testa med samma data som hittade felet, för att se om felet verkligen är åtgärdat. Ett testfall skall innehålla följande punkter:

- Identifierare av testfallet.
- Testprocedur, som förklarar vad som skall hända och vad man skall få ut av testningen.
- Vilket krav från kravspecifikationen som testas.
- Förlopp, som beskriver vilka sekvenser som skall utföras.
- Indata, som man skall använda och en beskrivning av vad man vill testa.
- Andra förutsättningar, som måste uppfyllas innan man kan köra testfallet, t.ex. att ett annat program måste köras först, eller att man måste trycka på en viss knapp först. Om det är många andra villkor som måste uppfyllas, kan dessa skrivas ner i en separat testprocedur.
- Förväntad utdata eller den effekt som indatan skall ha.
- Skapare av testfallet.
- Datum.

- 3.2.2.1 Utformningen av testfall

När man utformar ett testfall kan man använda sig av olika principer. Dessa principer ligger alla mellan två ytterlighetsprinciper: Black Box (testa enligt målsättningarna) och White Box (inspektera koden). Eftersom acceptanstestning i huvudsak utförs av kunden är det en självklarhet, att det är Black Box-principen som man utgår från. Kunden har ju oftast ingen kunskap om hur koden skall se ut, utan testar systemet efter de krav, som har ställts på systemet i kravspecifikationen. För kunden spelar det ingen roll hur leverantören har löst problemen, huvudsaken är att de är lösta. Målet med Black Box-principen är att testa de funktioner, som är specificerade av kunden och att testa i det sammanhang, som systemet

⁹ Ibid., 198-201

¹⁰ Birk, Olnäs och Penker, 172, 175-178

¹¹ Kenneth Jacobsson, *Programkonstruktion och Projekthantering, 10p, Systemtest* (Högskolan Dalarna, 1999), 2-3

skall användas. Det är dock omöjligt att utföra denna metod helt uttömmande, då det är omöjligt att testa alla kombinationer av indatavärden inom en rimlig tid. Man får istället inrikta sig på vissa indatagrupper och sedan göra stickprov, för att få en någorlunda heltäckande överblick.

Att skriva bra testfall är en av de svåraste punkterna inom testningen. Det kräver både fantasi, skicklighet och konstruktivt tänkande. Syftet med testfallen skall ju vara att knäcka programmet. Testfallen hämtas ur de krav och specifikationer som finns och testfallen skall innehålla referenser till dessa krav, så att man vet vad det är man testar.

- 3.2.2.1.1 Vad bör man tänka på när man skriver ett testfall?

När man skriver ett testfall för ett visst krav, skall man noga tänka igenom vad funktionen, som uppfyller kravet, skall göra och sedan hitta på så många testfall som möjligt med denna funktion. Man skall välja testfall, som inte liknar varandra så mycket, att de inte tillför något nytt. Man bör testa all möjlig felinmatning, som t.ex. negativa längder, för stora tal, siffror där bokstäver förväntas o.s.v. Man bör också testa att stänga ner programmet utan att spara, missa att fylla i information mm. Om systemet skall användas i fleranvändarmiljöer är det viktigt att även testa detta. Man bör testa hur systemet fungerar under samtidig användning och under hård belastning.

Att konstruera testfall som täcker alla felinmatningar, knapptryckningar, menyval o.s.v. är praktiskt taget omöjligt. För det första skulle det vara så tidskrävande, att systemet aldrig skulle bli godkänt. För det andra skulle det kosta så mycket pengar, att kunden aldrig skulle köpt systemet. Man får helt enkelt göra en avvägning och testa det som användarna kommer att använda systemet till samt det, som kan missuppfattas och leda till felinmatning. Det bästa sättet är då självklart att låta användarna vara med och konstruera testfallen. Om det är möjligt kan man använda sig av redan existerande testfall för systemet och lägga till testfall, om det tillkommit ny funktionalitet.

Man måste dock komma ihåg att testfallen skall vara nedskrivna innan testerna utförs. Det är viktigt att ha en detaljerad beskrivning, så att testet kan utföras igen om man upptäcker ett fel. Man skall undvika testning, där man testar slumpvisa indata, utan att ha någon kontroll på, vad som verkligen matas in.

3.2.3 Testprocedur ¹²

Om testproceduren inte är alltför omfattande, skrivs den inne i testfallet. Är den dock mer omfattande, skall den lyftas ut som ett eget dokument och skall då innehålla följande punkter:

- Identifierare av testproceduren.
- Referens till testfallet.
- Procedursteg, d.v.s. i vilken ordning man skall ge kommandon.
- Verktyg, program, maskinvara eller drivrutiner som används.

¹² Eklund och Fernlund, 194

3.2.4 Testprotokoll och testrapport ¹³

När testningen är slutförd sammanställer man en rapport, testrapporten, som innehåller de testprotokoll som skrivs efter varje utförd testbeskrivning. Testprotokollet skall innehålla följande:

- Identifierare av testprotokollet.
- Vilka testfall som har utförts inom testbeskrivningen.
- Resultaten av testfallen.
- Tidpunkten när testfallen har utförts.
- Ansvarig för testfallen och testbeskrivningen, d.v.s. ansvarig för att testfallen har utförts och att de redovisade resultaten är korrekta.
- Sammanfattning av de fel, som har framkommit av testfallen.

En strategi för hur felrapportering och felkorrigering skall gå till, skall upprättas tillsammans med testledaren.

Testrapporten är slutredovisningen av hela testplanen. Förutom att samla alla testprotokoll, bör man också sammanfatta alla större fel och problem som uppkommit under testerna. För ett exempel på hur en mall för en testrapport kan se ut, se Bilaga 3.

3.3 Hur stor del av systemet skall man testa?

Det finns många olika principer för hur man skall testa ett system. Räcker det att testa små delar av systemet var för sig eller måste alla testas tillsammans? När det gäller acceptanstestning är det viktigt att hela systemet testas precis så, som det skall användas. Det skall installeras i en skyddad testmiljö, som är så lik den miljö hos kunden där det skall användas som möjligt. Det kan utföras i en så kallad parallellinstallation, där det nya systemet är i drift parallellt med det gamla. På så sätt kan man också köra vissa jämförande tester, för att se om resultatet blir likadant i båda systemen.¹⁴

3.3.1 Dokumentation

Det är inte bara själva systemet som skall testas utan även användardokumentationen och driftsdokumentationen. Dessa dokument skall granskas och kontrolleras om de är riktiga mot systemet och mot de krav som är uppställda. Man kan använda systemet vid sidan om för att kontrollera. Kontroll skall också göras för att se om dokumentationen hänger ihop och att innehållet inte är motsägelsefullt. Fel och otydligheter skall felrapporteras och alla kommentarer skall dokumenteras i testrapporten.¹⁵

¹³ Ibid., 194-195

¹⁴ Ibid., 202-209

¹⁵ Birk, Olnäs och Penker, 179

3.3.2 Riskanalys ¹⁶

Eftersom det är praktiskt taget omöjligt att testa alla delar av ett system lika mycket, är det viktigt att göra en riskanalys. Man analyserar, vilka delar som bör testas mest och försöker förutse, om det kommer att inträffa stora fel om man testar en del lite mindre o.s.v. Man utreder även om man verkligen tjänar tillräckligt med tid genom att hoppa över testningen av vissa delar. Om man inte tjänar någon tid så är det ju onödigt att hoppa över testningen. En riskanalys bör innehålla följande punkter:

- Vilka delar skall testas?
- Hur omfattande är testningen av varje del?
- Hur mycket tid tjänar projektet på att hoppa över testningen av en viss del?
- Hur stor risk medför det att hoppa över testningen av en viss del?

3.4 Hur länge skall man testa? ¹⁷

Eftersom det nästan är omöjligt att uppnå en helt felfri programvara, är det mycket svårt att avgöra när testningen är klar. Ofta är det budgeten som bestämmer hur lång/kort tid man har på sig att testa. Ett viktigt kriterium på att testningen är klar, är att alla krav är korrekt implementerade. För att kunna bedöma hur väl programvaran är testad, finns det sex metoder som man kan ta till hjälp:

- *Viss täckningsgrad är uppnådd.* Detta mäts genom att kontrollera hur stor del av alla indatakombinationer, som har testats.
- *Visst antal fel funna.* Testningen anses vara klar, när ett visst antal fel har hittats. En rimlig målsättning brukar vara 10-20 fel per 1000 rader.
- *Antal funna fel per dag under en viss nivå.* Testningen anses vara klar, då antalet funna fel per dag har sjunkit under en förutbestämd nivå.
- *Viss andel av de inducerade felen funna.* En utomstående person stoppar in påhittade fel i koden. Den andel av dessa påhittade fel, som sedan hittas i testerna kan användas för att uppskatta antalet riktiga fel, som fortfarande finns kvar i programmet. Eklund och Fernlund föreslår en formel i sin bok "Programkonstruktion med kvalitet – projekthantering och ISO 9000":

Totalt antal kvarstående riktiga fel = (antal funna riktiga fel * totalt antal instoppade fel) / antal funna instoppade fel

Om t.ex. 20 fel stoppas in i koden och våra tester hittar 10 av dessa samt 10 riktiga fel så, innehåller koden ytterligare 20 riktiga fel.

- *Viss tillförlitlighetsgrad är uppnådd.* Detta kan man uppskatta då testfallen är skapade så att de kommer att likna de riktiga körningarna av systemet. Man kan då mäta hur ofta det uppkommer fel vid vanliga körningar.
- *Tiden för testning är slut.* Detta är den vanligast förekommande metoden, men också den sämsta. Den testar inte programvarans tillförlitlighet och det kommer troligtvis att uppkomma fel, när programvaran tagits i drift.

¹⁶ Ibid., 174

¹⁷ Eklund och Fernlund, 195-196

Vad och hur mycket som skall testas, beror också på kundens krav för acceptering av systemet. Om det är ett system som har höga krav på exempelvis säkerhet, är det viktigt att testerna blir uttömmande, medan det i andra fall kan räcka med att kontrollera att funktionerna, användargränssnittet, prestanda o.s.v. motsvarar de krav man har. Det är viktigt att systemet körs i skarp miljö vid testerna, d.v.s. i en miljö som är lik den miljö som systemet kommer att köras i senare.

3.5 Vem skall testa?

Eftersom det är kunden som skall godkänna systemet för leverans, är det också kunden som skall acceptanstesta systemet. Det är då viktigt, att det är de verkliga slutanvändarna som testar och inte bara beställaren av systemet som kanske inte alls kommer att använda systemet utan bara är ansvarig för beställandet av det samma. Testarna skall helst ha lite datorvana så att inte allting är nytt för dem när de sätter sig framför datorn. Testarna får ansvar för en viss del, vilket t.ex. kan innebära några testbeskrivningar och tillhörande testfall. De skall sedan utföra testerna och fylla i testprotokoll under tiden.

3.5.1 Organisation ¹⁸

En organisation/projektgrupp måste upprättas för att testningen skall kunna utföras på ett tillfredsställande sätt. Denna projektgrupp skall omfatta:

- Ansvarig för Test Configuration Management (testledning) och att skapa en testmiljö.
- Testansvariga för delprojekt (delar av funktionaliteten).
- Testare för framställning och exekvering av testfall och rapportering av resultat.
- Utvecklare från leverantören, tillgängliga under hela testfasen för att rätta till fel och svara på felrapporter.

3.5.2 Utbildning

Om systemet är helt nytt, måste alla testare utbildas i det nya systemet före test, så att de vet hur det fungerar. Alla testare måste få utbildning i systemet/funktionaliteten och testmiljöns verktyg och metoder. Detta är viktigt för att det inte skall bli tidsfördröjning och fel i testerna p.g.a. av okunskap om hur man använder systemet bland testpersonerna. ¹⁹

3.6 Uppsättning av testmiljö

En eller flera personer i projektgruppen skall avdelas för att sätta upp testmiljön. Testledaren skall innan testen startar ha försäkrat sig om att det finns resurser och tillgängliga testmiljöer för acceptanstestningen. Testmiljön skall kunna upprättas snabbt för att inte testningen skall bli försenad på grund av detta. Testledaren skall också planera eventuell beläggning på testmaskinerna, om dessa är en bristvara. Testmiljön skall vara den skarpa miljö eller en

¹⁸ Birk, Olnäs och Penker, 172

¹⁹ Ibid., 174

miljö, som är så lik de tänkta användarnas miljö som möjligt. När testerna körs igång skall testmiljön vara testad och alla testare skall ha åtkomst till systemet.²⁰

3.7 Felrapportera

Under testningen av programvaran kommer fel att hittas. När testarna hittat ett fel, skrivs en felrapport med en noggrann beskrivning av felet. Rapporten lämnas till programmeraren, som skall rätta felet. Programmeraren måste lokalisera felkällan för att undersöka, vad det är som orsakar felet. Denne måste exakt kunna peka ut var orsakerna till felet finns i programmet. När felkällan är identifierad kan felorsakerna elimineras och felet kan korrigeras. Då felet är rättat skickas en ny version av systemet till testarna, som skall testa med det gamla testfallet och gärna några nya testfall också för att se, att felet verkligen är rättat och att inga nya fel har uppkommit. Den som utfärdade felrapporten godkänner också att den är åtgärdad. Om felet kvarstår skickas en ny felrapport till programmerarna.^{21 22}

3.8 Slutrapportera

När testerna är genomförda och klara, samlas alla deltagare i projektet och sammanställer en slutrapport. Man utbyter erfarenheter och förbättringsförslag och redovisar detta i en rapport. Testledaren eller testansvarig skriver sedan ihop slutrapporten. Den viktigaste punkten i slutrapporten är självklart om systemet är godkänt, godkänt efter rättningar eller underkänt. För ett exempel på hur en mall för en slutrapport kan se ut, se Bilaga 4.²³

²⁰ Ibid., 172, 178

²¹ Eklund och Fernlund, 209-214

²² Birk, Olnäs och Penker, 179

²³ Ibid., 180

4 Bakgrund till praktikfall

4.1 Kund

Telia InfoMedia TeleVision AB ingår i Teliakoncernen och ansvarar för Telias underhållnings- och informationstjänster via TV:n. Företaget har ca. 170 anställda och omsatte 1998 drygt 550 Mkr.

Telia InfoMedia TeleVision paketerar, marknadsför och distribuerar nationell och internationell TV-kommunikation. De erbjuder ett brett sortiment av TV-tjänster, och genom att även erbjuda Internet, utnyttjas accessnätens höga kapacitet.

Telia startade sin kabel-TV verksamhet i Lund 1983 med 500 hushåll. 1988 hade de 500 000 hushåll anslutna och redan 1990 var 1 miljon anslutna.

I Sverige driver Telia InfoMedia TeleVision Europas näst största kabel-TV-nät med idag drygt 1,3 miljoner anslutna hushåll.

4.2 Utvecklingsprojektet

Projektet på Telia InfoMedia TeleVision omfattar utveckling av ny funktionalitet till existerande system, omskrivning av ett befintligt system till ny miljö och konvertering av databaser till ny miljö. För att ge en inblick i projektet kommer Kapitel 4.2.1 – 4.2.4 att beskriva de olika delavsnitten mer i detalj, trots att detta examensarbete senare bara kommer att beröra ett delavsnitt.

Projektet bemannas av personer, vilka arbetar i berörda system eller har en bred erfarenhet av organisationens arbetsförhållande och förutsättningar. För systemutveckling används i första hand Ingrid Wennerhag och Arne Wennerhag på företaget PC-syd och i andra hand Anders Ryberg på WM-data. Projektledare för projektet är Anders Söderberg, PreEra Konsult AB.^{24 25}

4.2.1 KAS-A

KAS-A är Telia InfoMedia TeleVisions nya produktionsplaneringssystem. Systemet är byggt i en treskiktslösning med Dataflex-databas för lagring av data, Microsoft Transaction

²⁴ Anders Söderberg, *Projektspecifikation HIPNET & Konvertering*, Revision PA2 (PreEra Konsult AB, 1999-06-27), 5

²⁵ Inga Samuelsson och Anders Söderberg, *Uppdragspecifikation Dukasanpassning*, Revision PA2 (Telia InfoMedia TeleVision AB/PreEra Konsult AB, 1999-05-26), 4

Server för affärslogik och gränssnitt mot databas, samt en tunn Visual Basic-klient för användargränssnitt. Systemet innehåller bl.a. funktionalitet för tidsrapportering samt resursplanering av tidsrapportering.

KAS-A-databasen skall i detta projekt konverteras till Oracle. Systemet skall vara byggt på samma sätt som befintligt system, d.v.s. med Microsoft Transaction Server och Visual Basic som klientutvecklingsverktyg.²⁶

4.2.2 Dukas

Dukas är ett system som används för felmottagning, dirigering av tekniker, samt återrapportering av åtgärdat fel. Systemet är byggt i Dataflex. Telia InfoMedia TeleVision vill kunna öka samordningsmöjligheterna och kvaliteten i tidsredovisningen genom aktivitetsbaserad redovisning. Telia InfoMedia TeleVision har utvärderat olika alternativ för anpassning av Dukas och kommit fram till att en total omskrivning av befintliga Dukas ger den långsiktigt mest effektiva lösningen.

Utvecklingsprojektet skall leverera ett system som i princip motsvarar den funktionalitet, som finns i befintliga Dukas, inklusive vissa tillägg och borttag av funktioner. Givetvis skall de möjligheter, som ett modernt Windows-gränssnitt medger tas till vara. Det åligger uppdragsgivaren att föreslå sådana lösningar, som effektiviserar slutanvändarnas arbete. I uppdraget ingår bl.a. att konvertera befintlig data i Dukas Dataflex-databas till den nya Oracle-databasen och handledning vid användarledd acceptanstestning.²⁷

4.2.3 HIPNET

Telia InfoMedia TeleVision kommer under 1999 starta tjänsten Internet Cable. En förutsättning för Internet Cable-försäljning är att fastighetsägaren anslutit sig till tjänsten HIPNET. Då Telia InfoMedia TeleVision förväntar sig volymförsäljning av HIPNET-anlutningar, krävs ett utökat systemstöd i systemet KAS-F.

HIPNET och framför allt Internet Cable ställer krav på dokumentation av basnät. Detta innebär att även de tekniska dokumentationssystemen måste anpassas.

HIPNET-projektet avser att utveckla applikationerna KAS-F, Dukas, KAS-A och Dokas så, att de kan tillgodose verksamhetens krav på affärsstödande funktionalitet för produkterna HIPNET och Internet Cable.²⁸

4.2.4 Konvertering

Telia InfoMedia TeleVision har beslutat att byta teknisk miljö för samtliga befintliga Dataflex-applikationer. Den valda målmiljön är Oracle. Bytet sker främst utifrån driftmässiga aspekter. De system som skall konverteras är KAS-F och Dokas.²⁹

²⁶ Ibid., 3-4

²⁷ Ibid., 3

²⁸ Anders Söderberg, *Projektspecifikation HIPNET & Konvertering*, 3

²⁹ Ibid., 3

4.2.5 Leveransgodkännande

Telia InfoMedia TeleVision leveransgodkänner systemen genom att utföra acceptanstestning. Testerna skall utföras av en utsedd användargrupp, där systemens samtliga funktioner kontrolleras. Testerna skall koncentreras på användarens syn på funktionaliteten.

Acceptanstesterna kommer att förberedas genom framtagning av testmaterial och facit, och genomföras av testpersonerna med teknisk assistans av WM-data och PC-syd.

Om det under testperioden uppkommer sådana fel, att testerna inte kan fortsätta, utför WM-data omgående erforderliga justeringar. Telia InfoMedia TeleVision kommer under hela testarbetet att föra testprotokoll, vilket kommer att överlämnas till WM-data efter avslutad test. Möjlighet skall finnas för WM-data att kontinuerligt följa testarbetet, så att vissa fel kan korrigeras parallellt med acceptanstesterna. Efter en felkorrigering testas de funktioner som i testprotokollet rapporterats som felaktiga, men inga övriga funktioner. När samtliga fel i testprotokollet är åtgärdade, anses Telia InfoMedia TeleVision ha godkänt systemet.³⁰

Då det kan bli rörigt och alltför omfattande att beskriva tester av alla system, som ingår i utvecklings-projektet på Telia InfoMedia TeleVision har jag valt att i denna rapport endast inrikta mig på testerna som rör HIPNET-anpassningarna.

³⁰ Anders Ryberg, *Allmänna villkor*, Version 1 (WM-data, 1999-05-27), Sid. 5

5 Genomförande av acceptanstestning av HIPNET-anpassningar

Detta kapitel beskriver testningen av HIPNET-projektet med utgångspunkt från dokument och information från Telia InfoMedia TeleVision AB och PreEra Konsult AB.

5.1 Testfacit byggs upp

En del förändringar har redan genomförts för att anpassa befintliga system till HIPNET- och Internet Cable-försäljning. Denna utveckling skall nu fortsätta och har specificerats i en kravspecifikation kallad "Analys av stödsystemen för HIPNET". Syftet med denna kravspecifikation är att utreda vilka förändringar som behöver göras i stödsystemen, samt att tidsuppskatta dessa förändringar. Analysen gäller enbart HIPNET, då förändringar gällande Internet Cable redan är genomförda.

5.1.1 Kravspecifikationen³¹

Kravspecifikationen innehåller många fackuttryck och förkortningar från Telia InfoMedia TeleVision. Då dessa inte har någon betydelse för syftet med det här arbetet, kommer de inte att beskrivas närmare. Varje system, som skall genomgå förändringar specificeras nedan var för sig.

- 5.1.1.1 Dokas

- Idag registreras det på varje nod, när noden är beställd och levererad. Nytt önskemål/krav är att det dessutom måste gå att se när noden är planerad för leverans. Detta innebär ett nytt fält.
- En offert som gäller HIPNET och som beställs i KAS-F, genererar ett projekt i KAS-A. I KAS-F skall det registreras, vem som är projektör för projektet. Här finns det ett önskemål/krav att förslag på projektör, för projekten som gäller HIPNET, skall lämnas som standard. För att detta skall fungera måste HIPNET-projektör anges för varje HC i HC-registret. Denna information kopieras över till HC-registret i KAS-F och används sedan för att peka ut projektör när projekt "Basnät bredbandsnät nytt 2" skapas utifrån KAS-F. Detta kräver ett nytt fält i HC-registret.

- 5.1.1.2 Dukas

- Det skall, i Dukas, gå att se det nya fält, som önskas i Dokas, d.v.s. när en nod är planerad för leverans.

³¹ Tommy Harström, *Analys av stödsystemen för HIPNET*, Revision B (Telia InfoMedia TeleVision AB, 1999-04-26), 7-12, 14

- **5.1.1.3 KAS-F och KAS-A/KAS-F (gamla Prokas-registren)**

- 5.1.1.3.1 Bygginfo

- För avtalen av typ ”HIPNET nytt” krävs två bygginfo. Ett som gäller det vanliga basnätet och ett som gäller byggandet av nod eller ombyggnad av basnätet. Detta kräver fler möjliga värden i fältet TYP i register BINFO och program BYQ, samt nya regler för registrering.
- För bygginfo, som gäller det vanliga basnätet, skall projektör väljas från en lista och det som skall registreras, skall både vara signatur och namn (idag är det bara namn).
- För bygginfo som gäller nod och ombyggnad av basnät, skall projektör hämtas med automatik från HC-registret. Detta för att innesäljarna inte vet vem som är projektör på respektive HC. Projektören läggs som förslag, men skall gå att ändra.
- I PROJ i KAS-A/KAS-F (gamla Prokas registren) måste det anges vilken bygginfo som gäller för projektet, eftersom händelsen i KAS-F som skapade projektet kan ha flera bygginfo.

- 5.1.1.3.2 Adressregistret (ADRR)

- Idag går det att se vilken nod som betjänar en adress och när noden är beställd och levererad. Det skall dessutom gå att se när en nod är planerad för leverans, d.v.s. samma fält, som önskas i Dokas.
- Det måste gå att radera/ändra fältet ”Fastighetsnät ombyggt” för en adress på samma sätt, som det nyregistreras att det är ombyggt, d.v.s. via basnätshändelsen.
- Fältet ”Fastighetsnät Internet” skall ej uppdateras med automatik via UPPA, när det inte är TIMTV, som byggt fastighetsnätet. När fastighetsnätet är byggt av TIMTV, skall fältet uppdateras med automatik och informationen skall hämtas från KAS-A.

- 5.1.1.3.3 Arbetsflöde för hantering av fältet ”Fastighetsnät Internet” då Telia InfoMedia TeleVision ej byggt fastighetsnätet.

Fastighetsägaren bygger nätet själv:

1. Fastighetsägaren rapporterar in till säljaren att fastighetsnätet är klart.
2. Säljaren markerar, att HIPNET i fastighetsnät är byggt. Registreras i datumfält ”Fast nät bygg”. Det datum som anges är klardatum.
3. KAS-F kvalificerar fastighetsnätet för driftsättning genom följande kontroller:
 - Kunden har ingen FA-H-händelse på adressen.
 - ”Fast nät bygg” skall vara ifyllt.
 - Aktiv BA-H-händelse på adressen.
 - Internet klar = 2 (basnät och HC tekniskt klara).När ovanstående är uppfyllt, skapas en arbetsorder för driftsättning i KAS-A.
4. Tekniker erhåller arbetsorder ur KAS-A.
5. Återrapport av arbetsorder sker. Om driftsättning är OK, uppdaterar KAS-A ”Fast.nät.internet” med ett J på aktuella adresser under ÖP:n.

- 5.1.1.3.4 Produktionsdata

Några fält, som presenteras under produktionsdata, skall plockas bort, eftersom de inte används längre och KAS-A inte heller uppdaterar dem.

- Alla fält som rör ”Teknikberedning”, alla fält som rör ”Planering” och fältet ”Ansvarig för klarskrivning” (inte datum) skall plockas bort.
- Dessutom skall fälten för ”Inmätning” byta plats med fälten för ”Utförande”.

- 5.1.1.3.5 SALJ

SALJ skall kunna hantera:

- Förtidsomförhandling och aktiv omförhandling av avtal från basnätsavtal till HIPNET-avtal. Aktiv omförhandling är, när det gamla avtalet stoppas innan avtalet gått ut och det nya börjar istället. Detta måste markeras så, att statistiken kan redovisa detta under en egen rubrik.
- Makulera en framtida omförhandling till ett basnätsavtal och ersätta den med ett HIPNET-avtal. Det finns kunder, där omförhandling redan gjorts till ett basnätsavtal och där avtalet ännu inte trätt i kraft, som vill ha ett HIPNET-avtal. För dessa kunder måste det framtida avtalet makuleras och ersättas av ett HIPNET-avtal. Hänsyn måste tas till statistiken. Eventuellt skall detta lösas genom att man omförhandlar den framtida händelsen istället för att makulera den. Elisabeth testat om det går.

- 5.1.1.3.6 Statistik

Fyra nya rubriker tillkommer i statistiken och skall heta:

- Omförhandlade aktivt Bas-Hipnet FIN-KOLL.
- Omförhandlade aktivt Bas-Hipnet KOLL-KOLL.
- Omförhandlade aktivt Bas-Hipnet FIN-KOLL utan tidigare mån.avg.
- Omförhandlade aktivt Bas-Hipnet KOLL-KOLL utan tidigare mån.avg.

Under dessa rubriker skall samma information redovisas, som under de andra omförhandlade-blocken.

Dessutom skall rubriker skapas för varje produkt för redovisning av uppsagda avtal. Antal avtal som sagts upp under perioden skall redovisas. Dessutom skall uppges hur många hushåll avtalen gäller och intäkter totalt för dessa hushåll per månad.

- 5.1.1.3.7 Ägarbyte 1:1 FIN-KOLL

Det går inte att göra ett ägarbyte av en individuell fastighetsägare till en kollektiv fastighetsägare. Vid försök att göra det meddelas, att det finns flera kunder på denna adress, d.v.s. de individuella kunderna och därefter avbryts ägarbytet. Till att börja med måste detta ändras så, att det går att göra ägarbytet. Nästa steg är att utveckla detta så, att de individuella kunderna stoppas med automatik och att de får ett brev med automatik.

- 5.1.1.3.8 Ägarbyte 1:1 HIPNET-HIPNET

När en fastighet, där det finns en beställning på HIPNET, men där det ej hunnit byggas ännu, byter ägare, skall projekten, som skapats från den första kundens beställning, avslutas och nya projekt skall skapas från den nya kundens beställning.

- **5.1.1.4 KAS-A**

- Vid sökning av projekt skall det gå att söka på projekt av typen "Basnät bredbandsnät nytt 1" och "Basnät bredbandsnät nytt 2" med hjälp av fältet "kas_extra" som finns i PROJ i KAS-A/KAS-F. Detta fält kan ha värdena 1 eller 2.
- Vid konvertering (kopiering) av projekten "Basnät bredbandsnät nytt 2" från KAS-F, skall projektören för HIPNET hämtas in till projektet istället för projektören för nybyggnationen.
- Skapa arbetsorder driftsättning - KAS-F skall ges möjlighet att skapa upp ett projekt, som avser "driftsättning" av en ÖP för Internet Cable.

- Skapa återrapport driftsättning - I KAS-A skall en speciell åtgärdsrapport för ”driftsättnings”-arbetsorder skapas. Skapa en flagga ÖP OK J/N. Om J skall samtliga adresser under aktuell ÖP uppdateras i KAS-F med att de är klara för Internet (Fast.nät.internet= J).

5.2 Testplanen ³²

Testplanen för HIPNET-projektet är sammanslagen med testplanen för de andra projekten. Kapitel 5.2 berör dock bara de delar som rör HIPNET.

5.2.1 Bakgrund

Telia InfoMedia TeleVision AB kommer under 1999 starta tjänsten Internet Cable. En förutsättning för Internet Cable-försäljning är att fastighetsskötaren anslutit sig till tjänsten HIPNET. Då Telia InfoMedia TeleVision förväntar sig volymförsäljning av HIPNET-anslutningar, krävs ett utökat systemstöd i KAS-F. HIPNET och framförallt Internet Cable ställer krav på dokumentation av basnät. Detta innebär, att även de tekniska dokumentationssystemen måste anpassas.

Innan systemen införs, skall tester genomföras för att säkerställa funktionalitet, prestanda och kvalitet.

5.2.2 Övergripande tidplan

- Vecka 32, 1999: HIPNET tester startar.
- Vecka 33, 1999: HIPNET tester fullföljs.
- Vecka 34-35, 1999: HIPNET tester avslutas.

Driftsättning sker under helgen vecka 36, 1999.

5.2.3 Förberedelser

Underlag för testerna är kravspecifikation ”Analys av stödsystem för HIPNET”.

- Testfall genomförs för aktuella funktioner.
- Förväntat resultat dokumenteras i samband med test.
- För provningsprotokoll genomförs mer omfattande testfall och förväntat utfall.

5.2.4 Deltagare

Elisabeth Franklin och Tommy Harström.

³² Anders Söderberg, *Testplan HIPNET, Konvertering och Dukas*, Revision PA2 (PreEra Konsult AB, 1999-08-06), 3-4

5.2.5 Genomförande av tester

- Testerna genomförs vecka 32-35, 1999.
- Vecka 32 testas den funktionalitet, som levererats t.o.m. vecka 31, 1999.
- Vecka 32, 1999, genomförs kompletterande specifikationer och utveckling av funktioner i Dokas och KAS-F, som gör att testerna kan fullföljas.

6 Jämförelse och resultat

Detta kapitel innehåller en jämförelse mellan teorikapitlet (Kapitel 3 "Genomförande av acceptanstestning") och acceptanstestningen på Telia InfoMedia TeleVision. Kapitlet följer samma rubriker som Kapitel 3, för att det skall bli lätt att följa. I de fall där praktikfallet inte följer de riktlinjer som togs upp i Kapitel 3, kan det förekomma nya rubriker eller avsaknad av rubriker.

6.1 Testfacit byggs upp

Projektet på Telia InfoMedia TeleVision har varit uppdelat i två delar. En del utvecklades under sommaren/hösten 1998 och är redan testad och klar, medan den andra delen, som omfattar nya önskemål och krav, är under utveckling och test under augusti/september 1999. I "Analys av stödsystemen för HIPNET", kravspecifikationen för HIPNET-anpassningarna, görs en beskrivning av vad som gjordes 1998 och av vilka nya krav som uppkommit. De nya kraven bildar kravspecifikationen för del två.

6.1.1 Kravspecifikationen

Kraven i "Analys av stödsystemen för HIPNET" är framtagna av användarna i diskussion med systemutvecklarna. Detta innebär, att båda parter är överens om vad som skall göras och eventuella missförstånd undviks. Då systemutvecklarna är väl insatta i de olika systemen och i vad som skall förändras, är kraven inte så detaljerade i sin beskrivning.

Under arbetets gång har ytterligare krav tillkommit till den ursprungliga kravspecifikationen och vissa krav har utgått, då de efter en tid bedömts som mindre viktiga eller alltför tids- och kostnadskrävande. Kraven har delats in i olika grupper beroende på vilket av stödsystemen som de tillhör. De är däremot inte indelade i grupper med avseende på typ av krav, d.v.s. om de är funktionella krav, eller om de är krav på användargränssnitt o.s.v.

Kraven specificerar vad de färdiga programvarorna skall kunna utföra och är utformade så, att de kan användas för att kontrollera att den färdiga produkten uppfyller dem. De är skrivna på enkel svenska och är lätta att förstå för både användare och systemutvecklare, som är väl insatta i Telia InfoMedia TeleVisions terminologi.

Vissa krav som t.ex. skulle göra hela systemet instabilt, eller som skulle behöva förankras i hela organisationen (Telia InfoMedia TeleVision), har bedömts vara för riskfyllda eller omfattande för att genomföras i detta skede och har därmed strukits från kravspecifikationen.

6.2 Testplanen

Testplanen för HIPNET-anpassningarna, ”Testplan HIPNET, Konvertering och Dukas”, är väldigt övergripande och tar bara i stora drag upp vad som skall göras. Testpersonerna förväntas göra en stor del av arbetet med att analysera kravspecifikationen och skapa testfall.

Enligt Eklund och Fernlund bör en testplan innehålla fem fasta punkter. Testplanen för HIPNET följer dessa punkter nästan till fullo. Den innehåller en *identifierare på testplanen*, d.v.s. ett namn, ”Testplan HIPNET, Konvertering och Dukas”. Dessutom finns det ett revisionsnummer, som identifierar versionen av planen. Den första delen i planen är en *introduktion till systemen*, där information ges om kunden och annan bakomliggande information. Efter introduktionen följer en *tidsplan* över när testerna skall genomföras. Även deltagare i projektet namnges. Testmiljö eller antal datorer avsatta för testerna tas däremot inte upp. Därefter beskrivs underlaget för testerna, d.v.s. kravspecifikationen. Däremot tas inga krav, *testobjekt*, upp i testplanen, utan här finns bara en hänvisning till kravspecifikationen. Testplanen tar heller inte upp vilka testbeskrivningar, som kommer att testa de olika kraven. *Ansvarig för testplanens genomförande* tas inte heller upp, men antas vara densamme som utformade testplanen, d.v.s. Anders Söderberg.

6.2.1 Testbeskrivning

Testpersonerna har fått fritt ansvar över testningen och har fått utföra den, hur de vill. Eftersom testerna är väl insatta i alla systemen och även i kraven för anpassningarna, har testbeskrivningar inte bedömts vara nödvändiga och därmed inte använts.

6.2.2 Testfall

I intervjuerna med testpersonerna framgår att även testfallen har legat på testpersonernas ansvar. Det framgår också att det inte har funnits något krav, från projektledningens sida, på att använda testfall. För detaljer angående intervjufrågor och svar, se Bilaga 5 och Bilaga 6. Testarna har inte skrivit utförliga testfall, men har ändå använt sig av en typ av testfall, då de har skrivit ner alla funktioner de utfört. Dessa funktioner har samlats i en funktionslista, vilken beskrivs i Kapitel 6.2.4 ”Testprotokoll och testrapport”. Eftersom testarna vet hur systemen fungerar då de använder dem varje dag, har ett namn på en funktion räckt, för att testaren skall förstå, hur denna funktion skall testas. Det har inte funnits behov av att skriva en utförlig beskrivning.³³

- 6.2.2.1 Utformningen av testfall

Testfallen eller funktionerna har utformats enligt Black Box-principen med kravspecifikationen som grund, för att testa, att kraven är uppfyllda och fungerar. Eftersom det är användarna som utformat testfallen, är dessa utformade enligt deras sätt att se på systemet. Detta är en fördel eftersom det bara är användarna, som vet hur systemet skall användas och vad för slags indata, som skall matas in i det. Användarna vet ju själva vilka inmatningar de kan göra och testat systemet efter dessa.

³³ Enkätfrågor besvarade av Elisabeth Franklin och Tommy Harström, 1999-09-08

Systemen kring HIPNET har genomgått omfattande tester inför år 2000 och dessa testfall har, med viss omskrivning använts, för att testa den nya funktionaliteten hos HIPNET-anpassningarna.³⁴

6.2.3 Testprocedur

Testproceduren ingår i testfallet och är då inte utformat som ett eget dokument.

6.2.4 Testprotokoll och testrapport

Testprotokollen i HIPNET-projektet är utformade som en funktionslista över alla testfall/funktioner (se Bilaga 7). Det finns således inte ett testprotokoll över varje testfall. Testfallen är inte heller indelade i testbeskrivningar, utan står som en egen funktion på listan.

En funktion är identifierad genom sitt namn. På listan finns även ett statusfält, som bestämmer vilken status funktionen har. D.v.s. om den är färdigtestad och godkänd, eller om den är under testning eller under utveckling o.s.v. Även datum och ansvarig för testet av funktionen finns med. Det finns även ett fält för kommentarer, där testpersonen kan skriva vilka fel som upptäckts och beskriva hur dessa uppkom.

Denna lista av testfall kommer även att utgöra testrapporten, eventuellt med en komplettering, innehållande beskrivning av de största felen och problemen som uppkommit.

6.3 Hur stor del av systemet skall man testa?

Eftersom systemen redan finns i drift hos Telia InfoMedia TeleVision och det bara är anpassningar till dessa system, som har gjorts i HIPNET-projektet, räcker det i detta fall att testa de funktioner, där HIPNET är inblandat. De andra funktionerna är ju redan testade, då systemen utvecklades från början. De funktioner där HIPNET-anpassningarna är inblandade, bör testas så noggrant som möjligt.

6.3.1 Dokumentation

Eftersom det inte är så stora förändringar i strukturen av systemen, då HIPNET-anpassningarna är genomförda, kommer dokumentationen inte att testas i detta projekt. HIPNET-anpassningarna innebär ju att befintliga system, som handhar kabel-tv-kunder och digital-tv-kunder m.m., även skall handha kunder och annan information, som har med HIPNET att göra. Grundstrukturen i systemen kommer inte att ändras, utan det kommer bara att ske vissa tillägg av information. Dessa tillägg gör det inte nödvändigt att testa ny dokumentation.

³⁴ Enkätfrågor besvarade av Elisabeth Franklin och Tommy Harström, 1999-09-08

6.3.2 Riskanalys

En riskanalys över hur mycket man skall testa, finns inte uppgjord i HIPNET-projektet. Däremot tas muntliga beslut, om hur mycket man skall testa och vilka funktioner, som är mer viktiga att testa o.s.v. Detta kan anses vara en riskanalys i sig, men den finns inte dokumenterad. Alla funktioner är i stort sett lika viktiga och skall testas lika mycket.

6.4 Hur länge skall man testa?

Enligt Kapitel 3.4 ”Hur länge skall man testa?” finns det sex metoder, vilka man kan använda sig av för att bedöma, hur väl programvaran är testad. HIPNET-projektet använder sig framför allt av en metod: *Viss tillförlitlighetsgrad är uppnådd*. Testfallen är skapade så, att de liknar de riktiga körningarna så mycket som möjligt och de skall köras i en miljö, som är så lik den verkliga som möjligt. De är uppbyggda kring vanliga arbetssituationer och systemen kommer att användas så, som de kommer användas i drift. När ett antal tester kring en situation är felfria, bedöms den funktionen vara godkänd.

6.5 Vem skall testa?

Det är viktigt, att det är de personer som skall använda systemet, som också godkänner det. Därför är det användarna av systemet, som också skall testa det. Testarna i HIPNET-projektet har alla tidigare datorvana och har arbetat med alla inblandade system, innan anpassningarna till HIPNET genomfördes. Testarna har blivit tilldelade olika delar av anpassningarnas krav, för att testa systemen mot dessa.

6.5.1 Organisation

Projektgruppen består av följande medlemmar:

- Anders Söderberg, projektledare från PreEra Konsult AB.
- Elisabeth Franklin, testare och testansvarig för delar av testerna, från Telia InfoMedia TeleVision.
- Tommy Harström, testare och testansvarig för delar av testerna, från Telia InfoMedia TeleVision.
- Anders Ryberg, systemutvecklare från WM data.
- Ingrid Wennerhag, systemutvecklare från PC-syd.
- Arne Wennerhag, systemutvecklare från PC-syd.

6.5.2 Utbildning

Eftersom testarna är väl bekanta med systemen och även varit med vid framtagningen av kraven för anpassningarna, är utbildning inte nödvändig.

6.6 Uppsättning av testmiljö

Testmiljön består av de datorer som testpersonerna använder i sitt dagliga arbete, vilket innebär att de alltid finns tillgängliga för testarna. För att miljön skall bli så lik den verkliga miljön som möjligt, har kopior tagits på de aktuella databaser, som systemen jobbar mot idag. Kopiorna togs dagen innan den första testdagen. Dessa kopior används sedan till att köra de nya systemen mot. På så sätt finns nästan dagsfärs information att testa med.

6.7 Felrapportera

Testarna bokför de fel som uppkommer i en felrapport, där en beskrivning av felet görs (se Bilaga 8). Felrapporten innehåller identifieringsnummer på felrapporten, funktion/testfall som felet upptäcktes med, beskrivning av felet, datum, status och signatur. Status är ett fält som talar om, var i processen som felrapporten befinner sig. Den kan t.ex. vara felanmäld, återtestad eller OK.

Felrapporten skrivs direkt när ett fel har hittats och lämnas till systemutvecklarna, som undersöker felet och åtgärdar det. En ny version av systemet skickas till testaren, som testar igen för att se, om felet är avhjälpt. Vid återtest används samma testfall, som hittade felet. Indatan behöver dock inte vara samma, så länge förutsättningarna är likadana, som när felet hittades. När felet är avhjälpt, sätts statusen på felrapporten till OK.

6.8 Slutrapportera

Eftersom HIPNET-projektet ingår i ett stort projekt, som omfattar både nyutveckling och konvertering av gamla system, kommer en omfattande slutrapport inte att göras i detta skede. Slutrapporten upprättas först, då alla delprojekt är klara. Detta kommer att ske utanför tidsramen för detta examensarbete.

En muntlig slutrapport har däremot avgivits då HIPNET-anpassningarna var färdigtestade. Den angav att de var godkända att sättas i drift. Anpassningarna sattes i drift 1999-09-07 och började direkt användas i verksamheten.

6.9 Sammanfattning av jämförelse

	Teori	Praktik	Skillnad
Planering			
<i>Testfacit byggs upp</i>	Kravspecifikationen skall användas som facit för testerna.	Kravspecifikationen användes som facit för testerna.	Ingen skillnad.
<i>Testplanen</i>	Testplanen skall vara övergripande, men bör innehålla fem punkter, enligt Eklund och Fernlund. Testbeskrivningar, testfall och testprocedurer bör användas för att definiera vad som skall testas och hur det ska testas. Testprotokoll och testrapport bör utformas så att testarbetet går att följa.	Testplanen är mycket övergripande och följer de fem punkterna nästan till fullo. Testbeskrivningar, testfall och testprocedurer har inte använts på Telia InfoMedia TeleVision. Däremot har funktionsnamn fungerat som ett slags testfall. Testprotokoll och testrapport är utformade som en funktionslista.	Telia InfoMedia TeleVision följer nästan exakt de riktlinjer som finns vid utformandet av testplanen. Däremot följs inte de riktlinjer som beskrivs då det gäller testbeskrivningar, testfall och testprocedurer. En typ av testfall används, då funktioner, som skall testas, namnges. Testprotokoll och testrapporter skrivs på Telia InfoMedia TeleVision, men de är inte utformade på samma sätt som riktlinjerna i teorin föreslår.

	Teori	Praktik	Skillnad
Genomförande			
<i>Hur stor del av systemet skall man testa?</i>	Systemet skall testas precis så som det skall användas. Även dokumentationen skall testas. En riskanalys bör utformas för att utreda vilka delar som bör testas mest noggrant.	Anpassningarna av systemet testades på det sätt som de skall användas. Dokumentationen har inte testats på Telia InfoMedia TeleVision. En riskanalys har inte heller upprättats, då alla delar skulle testas	Systemet är testat som det föreskrivs, men däremot är inte dokumentationen testad. Det finns inte heller en riskanalys uppgjord.

		lika noggrant.	
<i>Hur länge skall man testa?</i>	Hur länge man skall testa kan bestämmas utifrån sex olika metoder, enligt Eklund och Fernlund.	Telia InfoMedia TeleVision använde sig av en av de sex metoderna, <i>viss tillförlitlighetsgrad uppnådd</i> , för att avgöra hur länge de skulle testa.	Ingen skillnad.
<i>Vem skall testa?</i>	Eftersom kunden ska godkänna systemet är det också kunden som ska testa systemet.	Medarbetare på Telia InfoMedia TeleVision har acceptanstestat systemet.	Ingen skillnad.
<i>Uppsättning av testmiljö</i>	Testmiljön bör planeras och sättas upp i god tid före testerna.	Testmiljön består av användarnas egna datorer och kunde användas den första testdagen.	Ingen skillnad.

	Teori	Praktik	Skillnad
Rapportering			
<i>Felrapportera</i>	Då fel upptäcks skall en felrapport skrivas och skickas till programmeraren.	Testarna på Telia InfoMedia TeleVision har skrivit felrapporter som har skickats löpande till programmeraren.	Ingen skillnad.
<i>Slutrapportera</i>	En slutrapport bör utformas för att sammanfatta hela testarbetet.	En slutrapport har inte sammanställts i det här skedet.	En slutrapport finns inte upprättad ännu. Den kommer upprättas när alla projekt är klara.

Tabell 1 Sammanfattning av jämförelse mellan teori och praktik

7 Diskussion

Huvudsyftet med detta arbete var att jämföra hur acceptanstestning utförs i verkligheten i förhållande till teorin. Jag kan konstatera att projektet på Telia InfoMedia TeleVision i stora drag har följt de riktlinjer, som diskuterades i Kapitel 3 ”Genomförande av acceptanstestning”. Vissa avvikelser har dock gjorts, t.ex. vid testfallskonstruktionen, där Telia InfoMedia TeleVision har arbetat på ett sätt, som var mer naturligt för dem.

Vad som är rätt och fel när det gäller utförandet av acceptanstestning är svårt att säga. Varje projekt är unikt och det är omöjligt att generalisera. I vissa fall är det mycket viktigt att skriva utförliga testfall, för att underlätta testerna, medan det i andra fall inte alls behövs.

Beroende på förutsättningarna för projektet ställs olika krav på hur genomförandet av de tester som ska godkänna eller underkänna systemet, skall gå till. I de fall där leverantören utvecklar systemet inne på sitt kontor, är kraven på utförlig dokumentation och planering av testerna högre, än då leverantören genomför projektet hos kunden i nära samarbete med slutanvändarna.

Även själva systemet ställer olika krav på utförandet av testerna. I de fall där ett helt nytt system har utvecklats, krävs mycket omfattande och detaljerad dokumentation och planering av acceptanstesterna. Då ett existerande system anpassats med nya funktioner är inte detta lika viktigt, eftersom kunden och användarna känner till basystemet och vet hur det fungerar.

Projektet på Telia InfoMedia TeleVision har drivits i kundens lokaler. Kund och testpersoner har varit med från början vid fastställandet av kravspecifikationen och alla har varit delaktiga av och insatta i hela projektet. I detta fall tror jag inte, att det är nödvändigt att ”ödsla” tid på att skriva testbeskrivningar och testfall, då testerna har full kontroll på vad de skall göra utan dessa dokument. Vid uppdrag som liknar projektet på Telia InfoMedia TeleVision, d.v.s. där anpassningar av existerande system skall testas, är min uppfattning att de bästa testpersonerna är de, som arbetade med systemet innan anpassningarna genomfördes och de, som också kommer att använda systemet i framtiden. Under dessa omständigheter kan man genomföra acceptanstesterna på ett tillfredsställande sätt utan att dokumentera varje steg, vilket sparar både pengar och tid.

Testpersonerna på Telia InfoMedia TeleVision har varit noggranna vid testandet och testat det, som systemen skall användas till i praktiken. Eftersom de dagligen arbetar med systemen, var det lätt för dem att testa, då de bara behövde följa sina normala arbetsrutiner.

En problematik som kan dyka upp vid avsaknaden av utförliga testfallsbeskrivningar, är t.ex. då någon utomstående eller nyanställd vill ta del av testmaterialet vid ett senare tillfälle. Det finns då inget material, som beskriver testerna, utan bara namn på funktioner, som för en utomstående/nyanställd inte betyder någonting. Frågan är då, om man skall skriva testfall för ett eventuellt framtida behov eller om man skall skriva dem, för att de skall användas vid acceptanstestningen? Detta är givetvis också en fråga om hur mycket systemet får lov att

kosta och hur tidsplanen ser ut. Att skriva utförliga testfall, där de inte behövs, medför naturligtvis att systemet kommer att kosta mer.

Målet med acceptanstestningen är att testa, så att de krav som identifierar systemet är uppfyllda. Min slutsats är att detta kan genomföras på ett tillfredsställande sätt utan att strikt följa uppställda rutiner.

8 Källförteckning

8.1 Böcker

Beizer, Boris. *Black-Box Testing, Techniques for Functional Testing of Software and Systems*. United States of America: John Wiley & Sons, Inc., 1995

Birk, Björn, Patrik Olsnäs och Magnus Penker. *A UML process – the Astrakan Approach*. Sverige: Astrakan, 1998

Eklund, Sven och Hans Fernlund. *Programkonstruktion med kvalitet – projekthantering och ISO 9000*. Lund: Studentlitteratur, 1998

Kaner, Cem, Jack Falk och Hung Quoc Nguyen. *Testing Computer Software*. United States of America: Van Nostrand Reinhold, 1993

8.2 Elektroniska dokument

Jacobsson, Kenneth, 1999, *Programkonstruktion och Projekthantering, 10p, Systemtest*, Högskolan Dalarna, http://www.du.se/~d97kja/pdf_files/systemstestplan.pdf
Tillgänglig: 1999-08-09

NCP AB, *Utbildning: Beställning, acceptanstesting och godkännande av datasystem*, NCP AB, Stockholm, http://www.ncpab.se/Subpages/Top_o_Mid/utbilningmid.htm
Tillgänglig: 1999-08-09

8.3 Opublicerat material

Harström, Tommy, 1999-04-26, *Analys av stödsystemen för HIPNET*, Revision B, Telia InfoMedia TeleVision AB

Samuelsson, Inga och Anders Söderberg, 1999-05-26, *Uppdragsspecifikation Dukasanpassning*, Revision PA2, Telia InfoMedia TeleVision AB/PreEra Konsult AB

Söderberg, Anders, 1999-06-27, *Aktivitetsplan HIPNET och Konvertering*, Revision PA2, PreEra Konsult AB

Söderberg, Anders, 1999-06-27, *Projektspecifikation HIPNET & Konvertering*, Revision PA2, PreEra Konsult AB

Söderberg, Anders, 1999-08-06, *Testplan HIPNET, Konvertering och Dukas*, Revision PA2, PreEra Konsult AB

Ryberg, Anders, 1999-05-27, *Allmänna villkor*, Version 1, Sid. 5, VM-data

8.4 Enkätfrågor

Franklin, Elisabeth (1999-09-08), Telia InfoMedia TeleVision AB

Harström, Tommy (1999-09-08), Telia InfoMedia TeleVision AB

Bilagor

Bilaga 1 – Exempel på mall för testplan

Bilaga 2 – Exempel på mall för testbeskrivning med testfall

Bilaga 3 – Exempel på mall för testrapport

Bilaga 4 – Exempel på mall för slutrapport

Bilaga 5 – Enkätfrågor

Bilaga 6a – Enkät svar, Elisabeth Franklin

Bilaga 6b – Enkät svar, Tommy Harström

Bilaga 7 – Funktionslista/Testprotokoll

Bilaga 8 – Felrapport

Bilaga 1 - Exempel på mall för testplan ³⁵

Introduktion

- Översikt av systemet
- Syfte och omfattning

Testmiljö

- Innehåll
- Uppbyggnad

Test

Procedurer

- Målsättningar – syfte, typ och grad
- Riktlinjer och dokumentation
 - testkrav, manualer, procedur/metoder, checklistor, mallar, projektplan
- Utvärderingskriterier – kriterier för lyckat test, kategorisering av testresultat
- Felrapportering, felkorrigerande och omtestningsprocedurer

Tester

- Omgivningskrav – data och maskinutrustning
- Summering över vilka testfall som skall exekveras
- Kravmatris – karta över vilka krav som lett till vilka testfall (systemkrav och användarkrav)
- Testsvit 1
 - Namn
 - Syfte (vad som ska certifieras)
- Testsvit 2
- ...
- Testsvit N

Regressionstest

- Specifiera kriterier för vad som bör ingå som regressionstester och hur mycket som ska täckas in

Tid och Resursplan

- Kompetenser – vilka resurser behövs
- Organisation – ansvariga
- Utbildning/Information – vilken utbildning behöver genomföras
- Tidplan – för varje aktivitet eller fas

Risker

- Lista risker och eventuella åtgärder

Verktyg

- Lista de verktyg som skall användas, ev per miljö

Avslutskriterier

- Generellt
- Huvudprojektet
- Delprojekten

³⁵ Birk, Olnäs och Penker, 182

Bilaga 2 - Exempel på mall för testbeskrivning med testfall ³⁶

Identifikation

Beskrivning över vad som skall testas

Beskriv vilket/vilka krav, feature(s) eller egenskaper som skall testas

Syfte med testen

Indata till testen – Krav/Specifikationer

Testprinciper

Testmiljö

Generella förberedelser

Testkonfiguration

Testfall 1

Indata/Precondition

Procedur vid exekvering

Förväntat resultat/Postcondition

För featuren

För inblandade klasser enheter

Testfall 2

...

Testfall N

Regressionstestförslag

Referenser

³⁶ Birk, Olnäs och Penker, 183

Bilaga 3 - Exempel på mall för testrapport ³⁷

Identifikation

Allmän introduktion

Testmiljö

Utförda test

*Lista alla tester som genomförts och utfall
(testfall och regressionstest).*

Mätdata

³⁷ Birk, Olnäs och Penker, 183

Bilaga 4 - Exempel på mall för slutrapport ³⁸

Allmän introduktion
 Sammanfattning
 Omfattning
 Referenser
Projektet
 Utvärdering (bemanning, roller, samarbete, indata)
Testmiljö
 Utvärdering (miljö, verktyg, metod)
 Mätdata
Utförda test
 Uppfyllnad
 Utvärdering
Mätdata totalt
Systemkvalitet
Förslag på förbättringar
Acceptansresultat

³⁸ Birk, Olnäs och Penker, 183

Bilaga 5 - Enkätfrågor

Enkätfrågor angående test av HIPNET-anpassningarna

1. Har du skrivit testfall till testningen?

Om testfall inte har använts, fortsätt med fråga 7.

2. Skrev du testfallen innan testningen eller under tiden som testningen pågick?

3. Använde du någon mall för testfallsskrivning eller hur konstruerades testfallen?

4. Hade du kravspecifikationen för HIPNET-anpassningarna som hjälp vid testfallsskrivningen för att veta vad som skulle testas?

5. Hur noggranna är testfallen? Beskriver du steg för steg hur du ska utföra dem? Hur ser de ut? (Skicka gärna exempel)

6. Hur återtestar du ett tidigare fel? Används samma testfall som hittade felet? Testar du med samma indata?

Om testfall inte användes:

7. Hur har du genomfört testerna?

8. Hur återtestar du ett tidigare fel? Testar du med samma indata som hittade felet?

Bilaga 6a - Enkät svar, Elisabeth Franklin

Hej Karolina!

Jag har inte skrivit några testfall på det som jag testade när det gäller hipnetanpassningen. Jag hade kravspecifikationen som grund samt att Ingrid skickade mail kontinuerligt allt eftersom hon klarar med de olika funktionerna. När jag återtestade tidigare fel så använde jag en ny kund med nya indata fast med samma förutsättningar. Det som Tommy och jag testade tillsammans gjorde vi på samma sätt vi använde nya indata fast samma förutsättningar.

Det som jag testade själv är ju sådana saker som jag jobbar med varje dag och därför hade jag inget behov av att skriva testfall där beskrev steg för steg vad jag gjorde. Om det hade funnits tid så kan jag väl kanske tycka att man skulle ha skrivit testfall och beskrivit steg för steg hur man utförde de olika testerna. Jag tror inte att det var någon som sa att vi skulle skriva några testfall på det vi testade när det gäller Hipnetanpassningen .

Om det är något du undrar över efter min jätte långa utläggning så får du gärna ringa mig på 070/ 5112507

hälsningar

Elisabeth

Bilaga 6b - Enkät svar, Tommy Harström

Hej Karolina!

Här kommer svaren på frågorna. Svaren är under frågorna.

1. Har du skrivit testfall till testningen?

Om testfall inte har använts gå till fråga 7.

SVAR: Jag har inte skrivit testfall till testerna. Jag har endast skrivit upp testfallet i det gemensamma dokumentet. Jag fick uppfattningen att vi inte skulle skriva testfall så jag har testfallen i huvudet. Nu när jag sitter och funderar på det så kanske det där med att inte skriva testfall endast gällde konverteringen. Jag har i alla fall inte skrivit något testfall. Jag hade endast ett fåtal själv och de flesta gällde statistiken. Jag har däremot hjälpt Elisabeth med hennes testfall och där skrev vi inte heller några testfall utan endast rubriker.

Om testfall inte användes:

7. Hur har du genomfört testerna?

SVAR: Utifrån de testfall jag skrivit upp, d.v.s. rubrikerna har jag använt kravspecifikationen för att göra testet. Eftersom jag varit med och gjort kravspecifikationen så har jag vetat hur programmen skulle fungera.

8. Hur återtestar du ett tidigare fel? Testar du med samma indata som hittade felet?

SVAR: För statistiken så har jag använt samma indata för jag har inte haft något annat val. För övriga testfall så har vi använt ny indata.

Rent generellt så tycker jag att testfall skall skrivas om tid finnes.

Mvh

Tommy Harström
Telia Infomedia Partner AB
Mobile +46 705177058
Office +46 31773 78 70

Bilaga 7 - Funktionslista/Testprotokoll

Funktionslista

System: Dukas

Status:

UTV-Under

utveckling

LEV-Levererad för

test

TEST-Test

påbörjad

AN- Felanmäld

ÅTER- Återtest

OK - Klar

Funktion	Status	Datum	Signatur
Registrera Felanmälan	UTV	990830	MAON

Bilaga 8 - Felrapport

FELRAPPORTSLOGG

System: Dukas

Status:

RAPP - Rapporterad

AN - Felanmäld

ÅTER – Återtest

OK – Klar

Felrapp.nr	Funktion	Felbeskrivning	Datum	Status	Sign.
1	Anmälan	Fel i behörighet	990830	OK	MAON