



Handelshögskolan
VID GÖTEBORGS UNIVERSITET
Institutionen för informatik
2003-06-27

TEKNISKA FÖRUTSÄTTNINGAR FÖR WEBBASERAD PROGRAMUTHYRNING

Abstrakt

Webbaserad programuthyrning innebär, till skillnad från traditionell försäljning och implementering av informationssystem, att kunder erbjuds hyra systems funktionalitet över Internet. Denna modell anses av många ha stor framtidspotential, men skeptiker ifrågasätter konceptets hållbarhet och vi upplevde kvalitetsrelaterade problem i samband med driftsättning av ett system som användes över Internet under förhållanden som ur ett tekniskt perspektiv liknar programuthyrning. Syftet med denna uppsats är att undersöka om tekniska förutsättningar, vad gäller prestanda, pålitlighet och säkerhet, är goda för att bedriva webbaserad programuthyrning och om dessa förbättrats under de senaste åren. För att besvara undersökningsfrågan genomfördes prövning av fem hypoteser med anknytning till problemområdet. Empiriskt underlag togs fram genom mjukvarutester genomförda 1999 och 2003 samt en enkätundersökning ställd till det testade systemets användare. Resultaten av mjukvarutesterna indikerade att tillräckligt god prestanda och pålitlighet kan uppnås, samt att prestanda signifikant förbättrats de senaste åren. Resultatet får ytterligare stöd av enkätundersökningen. Uppsatsens slutsats är att tekniska förutsättningar för webbaserad programuthyrning, inom uppsatt avgränsning, är goda.

Nyckelord: Programuthyrning, ASP, outsourcing, prestanda, pålitlighet, säkerhet

Författare: Stefan Häggkvist, Andreas Östberg
Handledare: Kjell Engberg
Magisteruppsats, 20 poäng

Innehållsförteckning

1	Inledning	3
1.1	Programuthyrning	3
1.2	Problembakgrund	4
1.2.1	Citat Solutions och tidredovisningssystemet	5
1.2.2	Kvalitetsrelaterade problem	7
1.3	Frågeställning	8
1.4	Val av hypoteser	8
1.4.1	Prestanda	9
1.4.2	Pålitlighet	10
1.4.3	Säkerhet	11
1.4.4	Systemutformning	11
1.4.5	Tiden	12
1.5	Avgränsning	12
1.5.1	Typ av system	13
1.5.2	Kund	13
2	Metod	14
2.1	Evaluering av hypoteserna	16
2.1.1	Prestanda	16
2.1.2	Pålitlighet	19
2.1.3	Säkerhet	20
2.1.4	Systemutformning	21
2.1.5	Tiden	21
2.2	Mjukvarutestning	22
2.3	Tester	24
2.3.1	Test A och B	26
2.3.2	Test C och D	26
2.3.3	Test E	28
2.4	Enkät	28
3	Resultat	30
3.1	Mjukvarutester	30
3.1.1	Justering av testdata	31
3.1.2	Test A och B, uppstartstider	33
3.1.3	Test C och D, svarstider	36
3.1.4	Test E, pålitlighet	38
3.1.5	Krypterat/okrypterat	39
3.1.6	Modemtester	40
3.2	Enkätundersökning	40
3.2.1	Prestanda	41
3.2.2	Pålitlighet	42
3.2.3	Säkerhet	43

3.2.4	Systemutformning	43
3.3	Övriga källor	43
3.3.1	Pålitlighet	44
3.3.2	Säkerhet	44
4	Diskussion	47
4.1	Prestanda	47
4.2	Pålitlighet	48
4.3	Säkerhet	49
4.4	Systemutformning	49
4.5	Tiden	50
4.6	Hypotesernas beviskraft	50
4.7	Slutsats	52
4.8	Framtida forskning	53
5	Referenser	54
5.1	Böcker	54
5.2	Artiklar/Rapporter	55

1 Inledning

"Webbased application outsourcing will ultimately become the dominant model for how applications are delivered" menade Phil Wainwright, utgivare av nyhetsbrevet ASP News Review 1998 [Wainwright 98]. Han var inte ensam om att höja ett nytt hett koncept som man kallade programuthyrning (på engelska ASP, Application Service Providing) till skyarna. Förväntningarna var extremt höga och stora såväl som små mjukvarutillverkare annonserade sitt intresse på området. Microsoft, IBM, Oracle, Netscape, PeopleSoft, Sun Microsystems, EDS och SAP var exempel på företag som satsade stort på konceptet. Framförallt var det dock i inledningsskedet mindre företag som erbjöd programuthyrning mot mindre och medelstora kunder [Caldwell 98].

Sedan millenieskiftet, då förväntningarna på konceptet var som störst, har många hundra ASP-företag startats, och de flesta av dessa har också hunnit försvinna. Hittills har de höga förväntningarna inte infriats. Samtidigt som entusiasmen inför möjligheterna med programuthyrning ändå lever kvar så finns det många som ifrågasätter konceptets hållbarhet. De skeptiska rösterna menar att det finns ett antal problem knutna till programuthyrning, problem som kanske blir för svåra att handskas med inom den närmaste framtiden. Bland annat gäller farhågorna säkerhet, pålitlighet, prestanda och kundens kontroll över systemen samt problem med utformningen av avtal och affärsmodeller [Wainwright 98, Boyd 00, Eisner 01, Larsson 01, Allen 02, Shepherd 02 etc.].

Vilka är det då som har rätt? Är det skeptikerna eller de som fortfarande tror på programuthyrning? I den här uppsatsen kommer vi att titta på hur ett webbaserat system fungerar dels som outsourcat, d.v.s. i extern drift, och dels i intern drift. Detta gör vi inte för att kunna uttala oss om system i intern drift utan snarare för att kunna dra slutsatser om den externa driften utifrån en jämförelse med den interna. Syftet är att titta på de tekniska förutsättningar för programuthyrning mot mindre företag.

1.1 Programuthyrning

Låt oss titta lite närmare på begreppen programuthyrning och ASP.

Programuthyrning innebär till skillnad från traditionell försäljning och implementering av informationssystem att kunden istället erbjuds tillgång till ett systems funktionalitet. Med detta menar vi att leverantören behåller systemet hos sig och står för underhåll av såväl mjuk- som hårdvara. I normalfallet handlar det snarare om en prenumeration av en eller flera applikationer än ett köp av mjuk- och hårdvara. Outsourcing av system är inget nytt koncept, den stora nyheten är att Internets utbredning och webbens grafiska gränssnitt har

givit helt nya möjligheter att nå många kunder utan större investeringar i nätverk och övrig hårdvara för kunden.

Den mest etablerade benämningen för företag som erbjuder denna typ av tjänster är ASP (Application Service Provider). *"An application service provider (ASP) is a company that offers individuals or enterprises access over the Internet to applications and related services that would otherwise have to be located in their own personal or enterprise computers."* [Verzion 03]

Det finns ingen enhetlig definition av begreppet programuthyrning, men de som försökt sig på brukar nämna följande egenskaper [Wainewright 98, Heinzl 98, Larsson 01, Susarla 01]:

Definition av programuthyrning

- Leverantören äger och förvaltar all hård- och mjukvara.
- Leverantören erbjuder applikationer som en tjänst åt sina kunder.
- Samma system erbjuds oftast till många kunder.
- En allmänt tillgänglig infrastruktur används (Internet).
- Betalning sker vanligt vis löpande, likt en prenumeration, d.v.s. man hyr systemet snarare än köper det.

Detta arbete studerar webbaserad programuthyrning, d.v.s. funktionalitet som erbjuds över Internet via en webbläsare. När vi i fortsättningen talar om programuthyrning så menar vi webbaserad sådan.

1.2 Problembakgrund

Konceptet med uthyrning av webbaserade system över Internet har av många lyfts fram som en modell med stor framtidspotential. Om man letar efter information om webbaserad programuthyrning upptäcker man mycket av det som skrivs främst behandlar förutsättningar ur ett affärsmässigt perspektiv snarare än ett tekniskt. Vi har dock under utvecklingen av webbaserade system själva stött på kvalitetsrelaterade problem av teknisk natur, framförallt i försök att göra system geografiskt oberoende. Problem som ställer upp en del frågetecken om konceptets hållbarhet.

Arbetet med uppsatsen har genomförts i samarbete med en företag som heter Citat Solutions. Bakgrunden till detta samarbete är en projektanställning som vi båda fick på företaget våren 1998. Under våren och sommaren konstruerade vi ett webbaserat tidredovisnings- och projekthanteringssystem som sedan implementerades i organisationen under hösten.

När systemet varit i drift ett tag funderade man på att sälja vår lösning till andra företag av ungefär samma storlek och organisationstyp. Tanken var då inte att sälja en färdigförpackad produkt, och inte heller att anpassa och implementera systemet hos den eventuella kunden, utan snarare att erbjuda systemens funktionalitet över Internet. Systemet lämpar sig för detta eftersom det i princip är geografiskt oberoende.

Vid ungefär samma tid etablerades begreppet ASP för ett sätt att sälja informationssystem som mycket liknade våra planer för tidredovisningssystemet. Planerna på att sälja tidredovisningssystemet förverkligades aldrig, men när vi beslöt oss för att göra en studie med fokus på programuthyrning framstod tidredovisningssystemet som ett mycket bra system att basera undersökningen på.

Detta tidredovisningssystem användes alltså inte i programuthyrningssammanhang och är därför i flera avseenden inte att betrakta som ett ASP-system. Om denna uppsats studerat exempelvis juridiska, sociala eller ekonomiska aspekter på webbaserad programuthyrning hade detta system inte varit användbart för studien. Det är dock så att de undersökningar på systemet som presenteras i detta arbete endast angriper tekniska aspekter på webbaserad programuthyrning. Så som systemet är utformat och på det sätt det användes under den tid det var i drift kan det i allt väsentligt likställas med webbaserad programuthyrning utifrån de tekniska aspekter vi intresserar oss för.

Trots detta kan man tycka att studien borde baserats på ett ”verkligt” fall av programuthyrning. Anledningen till att detta inte gjorts är att det i början av 1999 innebar vissa svårigheter att hitta ett lämpligt case då ASP fortfarande var i sin linda. Dessutom erbjuder studier av ett system man helt och hållet förfogar över överlägsna möjligheter att utforma och genomföra kontrollerade experiment.

Systemet skapades inte i syfte att skriva en magisteruppsats. När beslutet senare fattades att skriva en uppsats om programuthyrning var det tydligt att systemet lämpade sig väl för denna studie. Samtliga tester samt enkätundersökningen som presenteras i detta arbete är utformade och genomförda endast som en del i uppsatsarbetet och alltså inte som en del i utvecklingen av systemet och inte på uppdrag av Citat Solutions.

1.2.1 Citat Solutions och tidredovisningssystemet

Här följer en bakgrundsbeskrivning av varför och hur det system som senare kom att ligga till grund för undersökningen en gång skapades.

Citat Solutions ingår i Citat-koncernen och är ett företag med sin huvudverksamhet i Göteborg och Stockholm med totalt ca 100 anställda. Huvudkontoret ligger numera i Stockholm, men verksamheten har sitt ursprung i Göteborg. Företaget gick under 90-talet över från att producera CD-rombaserad multimedia till att koncentrera sig på avancerade Internet- och intranetapplikationer för att underlätta kunders marknadskommunikation.

Citat Solutions består främst av unga högutbildade människor. Företagets låga medelåldern samt en flexibel organisation gav ett för branschen typiskt ”kreativt kaos”. Detta tillsammans med den snabba organiska tillväxten och en hög arbetsbelastning hade 1998 lett till att utveckling av stöd för den interna administrationen kommit på efterkälken.

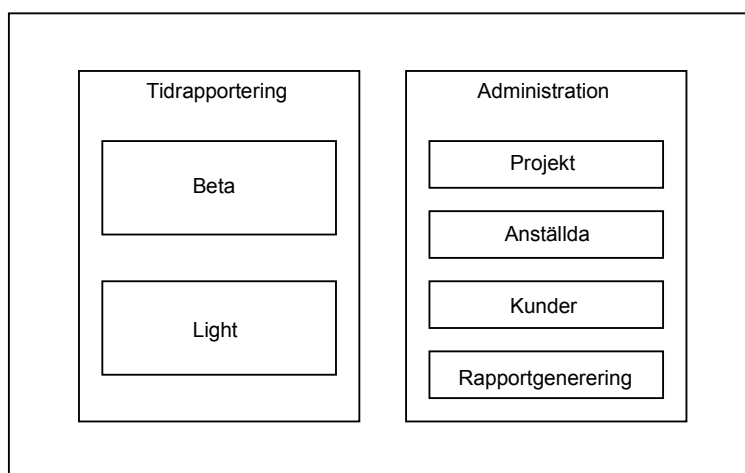
Ett av de mest påtagliga problemen hade med rapportering av arbetade timmar att göra. Metoderna för att rapportera sin tid upplevdes som besvärliga och detta innebar att man börjar slarva med sin rapportering, vilket ibland ledde till felaktig debitering med lägre vinst eller missnöjda kunder som följd. Man använde tidigare ett excelbaserat system där respektive anställds färdiga veckorapport skrevs ut på papper och lämnades in för bearbetning. Sammanställning för löneutbetalningar och faktureringar tog alltför stora resurser i anspråk.

Uppsatsens författare arbetade tillsammans med Hans Larsson, dåvarande VD Dick Eriksson och projektledare Shahab Gavami fram ovan nämnda system för tidredovisning och projekthantering i syfte att lösa dessa problem. Man ville ha automatisk generering av ett antal rapporter som tidigare krävde omfattande manuellt arbete att ta fram, t.ex. faktureringsunderlag, beläggning, lönerapporter o.s.v. Dessutom önskade man ett mer lättanvänt system för själva tidrapporteringen samt ett webbaserat sådant för geografiskt oberoende.

Systemet baseras på Active Server Pages (ASP, ej att förväxla med ASP i betydelsen Application Service Provider), en metod för webbprogrammering där html kombineras med scriptspråk som exekveras på servern, SQL-server (Microsofts relationsdatabas) samt Crystal Reports (rapportgenerering).

En förutsättning för ett fungerande tidredovisningssystem av den typ som önskades var att man även formaliserade hantering av projekt, kunder, anställda m.m. Därför utvecklades parallellt med tidredovisningen administrativa delsystem för detta. Dessa gav möjligheter att skapa projekt och underprojekt, samt att knyta anställda till specifika uppgifter, kunder till projekt o.s.v. Det var även här som rapportgenereringsverkygen fanns. Dessa funktioner vände sig dels till ekonomiavdelningen (rapporter som rör löner och fakturering) och dels till projektledare och företagsledning (planering och uppföljning av projekt). Den första versionen av systemen sattes i drift i augusti 1998. Systemen användes under drygt 2 år innan de ersattes av ekonomisystemet Maconomy.

Systemen är webbaserade och geografiskt oberoende. Detta gäller alltså såväl tidrapportering som projekthantering och rapportgenerering. De som använde systemen från göteborgskontoret gjorde detta via Citat Solutions intranät. Interaktion med systemen utifrån (t.ex. från stockholmskontoret och konsulter hos kund) skedde via företagets extranät, krypterat över Internet. Det faktum att systemen användes både internt över del lokala nätverket samt externt över Internet gav oss goda möjligheter att studera aspekter på programuthyrning eftersom de tekniska omständigheterna för systemens externa använd-



Figur 1.1 – Systemets olika delar

are i allt väsentligt var desamma som för en tänkt programuthyrningskund. Samtliga delsystem lämpar sig för programuthyrning, men de tester som genomförts i denna undersökning är gjorda på tidredovisningsdelen.

1.2.2 Kvalitetsrelaterade problem

Från början fanns endast en version av tidrapporteringsklienten. Internt fungerade denna bra, men när även stockholmarna började använda tidredovisningen uppenbarades tydliga kvalitetsproblem. Flera användare klagade på långa väntetider och olika fel som uppstod vid användning.

Systemet sades vara trögt att starta och använda. Vi kom fram till att detta bland annat berodde på ett expanderbart projekträd i stil med det som finns i Windows utforskare. Detta navigationsträd fungerade mycket bra i Göteborg där trafiken mellan användarnas datorer och serverna endast gick över det lokala nätverket, men när trafiken även gick över det publika Internet blev datamängderna för stora. Därför utvecklades snart även en "light-version" av tidredovisningen. Denna var enklare i sin uppbyggnad och grafiska framställning, men inkluderade ändå samtlig funktionalitet. När vi framöver i denna uppsats hänvisar till dessa två tidrapporteringsklienter kallar vi den ursprungliga för Beta (en tidig benämning som hängt med) och den senare för Light.

De svårigheter och problem som uppstod i samband med att systemet började användas utifrån fick oss att börja fundera kring möjliga tekniska begränsningar i samband med programuthyrning. Det var uppenbarligen inte bara att öppna brandväggen för extern

användning och luta sig tillbaka, detta trots att systemet var webbaserat och därmed, åtminstone i teorin, geografiskt oberoende.

Dessa erfarenheter tillsammans med det faktum att flera texter som behandlar webbaserad programuthyrning tar upp kvalitetsrelaterade problem bildar den bakgrund som detta arbetes frågeställning baseras på. Vi adresserar därmed en problemställning som både är teoretiskt och empiriskt grundad.

1.3 Frågeställning

Problembakgrunden som målats upp i stycke 1.2 utgör grunden för den undersökningsfråga detta arbete behandlar. Frågan lyder som följer:

<p><i>Är tekniska förutsättningar, vad gäller prestanda, pålitlighet och säkerhet, goda för att bedriva programuthyrning av medelkomplexa webbaserade system mot små och medelstora företag och har dessa förutsättningar signifikant förbättrats under de senaste åren?</i></p>
--

Vad som avses med medelkomplexa webbaserade system samt små och medelstora företag förklaras i stycke 1.5 *Avgränsning*. Metoden för att besvara undersökningsfrågan är hypotesprövning. Det tillvägagångssätt som används vid prövningen av hypoteserna kommer att beskrivas i detalj i kapitel 2 *Metod*. I detta kapitel ställs de gränsvärden och regler upp som används för att avgöra om förutsättningarna är ”goda”.

1.4 Val av hypoteser

Två centrala begrepp i vår metodik är kvalitet och hypotesprövning. *Kvalitet* därför att några av frågetecknen gällande programuthyrning gäller kvalitetsbegreppen prestanda, pålitlighet och säkerhet. *Hypotesprövning* för att denna typ av kvalitetsbegrepp lämpar sig väl för kvantitativa mätningar, och hypotesprövning är en bra metod att dra slutsatser ut resultaten av kvantitativa mätningar. Framgångsrik användning av hypotesprövning förutsätter naturligtvis att hypoteserna är formulerade på ett sätt som har relevans i förhållande till frågeställningen och att man har ett hållbart och korrekt sätt att testa dessa.

Hypotesprövningen baseras på ett antal mjukvarutester som genomförts mot två varianter av ett system av den typ som uppsatsens avgränsning avser (tidrapportering Beta och

Light). Mjukvarutesterna genomförs både internt och externt över Internet, och vid två tillfällen, 1999 och 2003. De hypoteser som testas behandlar prestanda, pålitlighet, säkerhet, systemutformning samt förändringar över tiden.

Av dessa fem hypoteser behandlar de tre första mjukvarukvalitetskriterier. Vi utgår här från litteratur som behandlar mjukvarukvalitet. Vidare har vi studerat litteratur kring IT-outsourcing och programuthyrning. Dessa kunskaper samt egna erfarenheter kring webbapplikationer används för att formulera hypoteserna utifrån antaganden kring hur systemen kommer att bete sig vid simulerad programuthyrning.

Prestanda, pålitlighet och säkerhet utgör några av de viktigaste och absolut vanligaste frågetecknen kring programuthyrning över Internet. *"The fact is that the early days of ASP were dogged by concerns about security, reliability and performance..."* [Shepherd 02]. I en undersökning bland 166 företag som inte är kunder till programuthyrningsföretag framkom detta tydligt. Företagen fick betygsätta betydelsen, från 1-5, av möjliga problemområden i samband med att hyra program över Internet. Överst med ett snittbetyg på 4,6 kom datasäkerhet, tvåa med 4,5 i snittbetyg kom prestandaproblem och femma kom pålitlighet med 3,9 i snittbetyg [Counts 02]. Tydligare än så kan det knappast bli. I denna undersökning och i flera andra texter återkommer prestanda, pålitlighet och säkerhet som tre viktiga faktorer för om programuthyrningskonceptet ska lyckas eller inte. Därför angrips dessa tre kvalitetskriterier i detta arbete.

1.4.1 Prestanda

Ett problem man stöter på i samband med prestandafrågor och Internet är att det svårt att avgöra var eventuella flaskhalsar ligger. Många faktorer spelar in, bl.a. leverantörens serverkapacitet, varierande belastning på nätverk och servrar, leverantörens bandbredd samt övriga tekniska utrustning, kundens bandbredd, hård- och mjukvara på klientdatorm, och naturligtvis den svåröverskådliga infrastruktur som kallas Internet däremellan.

En vanlig uppfattning är dock att Internet i sig faktiskt har kapacitet att stödja den typ av användning som studeras, detta under förutsättning att både leverantör och kund har den bandbredd och hårdvarukapacitet som krävs [Wainwright 98, Ledford 00].

Inom de ramar frågeställningen anger bör leverantörens servrar och bandbredd vara väl anpassade för den här typen av applikationer och borde inte innebära något avgörande problem vad gäller prestanda [Halpin 03]. Den tänkta kundens bandbredd är också hög och borde inte heller den innebära oöverstigliga prestandabegränsningar. Vi tror dock att prestanda kommer att uppvisa variationer eftersom man i samband med webbaserad programuthyrning via Internet inte åtnjuter någon garanterad bandbredd [Andréasson 96], men inte att dessa variationer kommer att utgöra något stort problem.

Sålunda formuleras den första hypotesen:

För webbaserad programuthyrning inom uppsatsens avgränsningar är prestanda acceptabel.

Vad menar vi då med acceptabel? Detta kommer vi att gå in på i avsnittet om evaluering av hypoteser (2.1.1).

1.4.2 Pålitlighet

“Reliability is the most important dynamic characteristic of almost all software systems.”
[Sommerville 96]

Många anser att pålitlighet är en av de absolut viktigaste egenskaperna hos ett mjukvarusystem [Counts 02, Susarla 01, Wetzell 01]. Mjukvarupålitlighet definieras ofta som sannolikheten för felfri exekvering över en viss tid och under vissa förutsättningar [Sommerville 96]. Man kan se på ett systems pålitlighet på olika sätt. Här studeras sannolikheten att systemet falerar när det utför en begärd operation. Det är svårt att hitta ett bra ord på svenska, men fortsättningsvis används ordet *felsannolikhet*.

En kompetent leverantör av programuthyrningstjänster förväntas välja internetleverantör, serverlösning och hårdvara med hög grad av pålitlighet. Givetvis är även utformningen av applikationen viktig för hög pålitlighet. Så länge leverantörer av webbapplikationer har detta i åtanke borde det i de flesta fall vara möjligt att åstadkomma lösningar med god pålitlighet. *“ASPs can apply their vast experience to implement best IT practices for superior levels of availability...”* [Halpin 03]

Vad gäller felsannolikhet ser vi ingen anledning till att denna skulle vara märkbart högre vid programuthyrning än vid intern drift. Vad som skulle kunna hända är att trafiken över Internet blir så fördröjd att det blir time out på servern vilket leder till att fel uppstår, men detta borde inträffa sällan. *“Technologically, the TCP/IP suite of protocols and the architecture underlying the Internet are stable and mature... ..Internet routing is dynamic, so packets sent through the Internet get to their destinations even if there are network outages along the way.”* [Harding 99]

Hypotes nummer två lyder:

För webbaserad programuthyrning inom uppsatsens avgränsningar är pålitligheten god.

Vi kommer precisera vad vi menar med god i avsnitt 2.1.2.

1.4.3 Säkerhet

Säkerhetsfrågan utgör ett orosmoment när system distribueras över Internet. Det är naturligtvis viktigt för en organisation att dess information är skyddad mot insyn av obehöriga samt från olika former av sabotage.

"Förutom problemet med bandbredd och tillgänglighet har ASP-konceptet en annan akilleshäla, nämligen säkerheten." [Larsson 01]

"Concerns of ASP customers is understandable. Outsourcing potentially exposes new vulnerabilities. Fueling that unease are well-publicized hacks of major corporations." [Hogan 00]

Då Internet trots all oro faktiskt används för överföring av känslig information, t.ex. ekonomiska transaktioner i samband med e-handel och bankärenden, måste det finnas etablerade metoder för att hantera säkerhetsproblem. Den information som hanteras i det typiska fallet av programuthyrning är dessutom mindre känslig och således mindre attraktiv för ett potentiellt angrepp.

Enligt Clive Shepherd [Shepherd 02] kan man förvänta lika hög eller högre säkerhetsnivå som på ett internt nätverk så länge programuthyrningsföretagen gör professionella val av säkerhetssystem och rutiner.

Med utgångspunkt från ovanstående resonemang formuleras den tredje hypotesen:

<i>För webbaserad programuthyrning inom uppsatsens avgränsningar är säkerheten god.</i>

Vi kommer precisera vad vi menar med god i avsnitt 2.1.3.

1.4.4 Systemutformning

I stycke 1.2.2 beskrevs hur problem uppstod då den första versionen av tidredovisning (Beta) började användas externt. Att ett system verkar fungera väl internt verkar inte utgöra någon garanti för att det även fungerar vid extern användning. Kan ett system fungera bra både internt och externt medan ett annat inte gör det trots att båda erbjuder samma funktionalitet och i grunden är baserade på samma webbaserade teknologi?

Vid design av klassiska klient-serversystem som används lokalt över ett snabbt nätverk ligger fokus för prestandaoptimering oftast på att snabba upp affärslogik samt läsning och uppdatering av data. När det gäller webbaserade system som distribueras över Internet bör man eventuellt prioritera något annorlunda: *"Among the prime victims of over-*

architecting are Web applications... ...they should be architected to favor speed of display over speed of data update.” [Landgrave 02]

Sevcik menar att prestanda över webben till stor del styrs av datamängden som överförs mellan server och webbklient från det att ett anrop initieras tills att sidan är laddad och uppritad [Sevcik 99]. Detta indikerar att det ställs speciella krav vid utformning av just webbaserade applikationer.

Med utgång från ovanstående formuleras hypotes fyra:

För webbaserad programuthyrning inom uppsatsens avgränsningar ställs specifika krav på utformning av system.

1.4.5 Tiden

Den avslutande tidshypotesen testar om man kan förvänta sig att förutsättningarna för denna typ av programuthyrning förbättras över tiden. Rent intuitivt känns det som att så borde vara fallet då teknikutvecklingen går framåt och datorer blir snabbare och snabbare. Man skulle dock kunna tänka sig att ett ökat antal internetanvändare, förändrat användarbeteende, förändringar i utbudet av webbtjänster samt skräppost, virusattacker och liknande kan leda till högre belastning på infrastrukturen vilket skulle kunna motverka förbättringar [Whong 97, Oracle 96, Myers 00]. *”Internet download speeds may well decline during the next few years”* [Nielsen 98]. Trots dessa farhågor tror ändå många att Internets kapacitet kommer att öka snabbare än belastningen [Loftus 00, Niccolai 00, Sevcik 01].

Baserat på ovanstående gör vi antagandet att prestandaförbättringar för Internet, hårdvara samt ökad tillgänglighet av bredbandstjänster kommer att innebära en generell förbättring av prestanda för webbaserade applikationer. Det vore önskvärt att kunna mäta om förändringar skett och i vilken riktning. Möjlighet till detta ges genom att mjukvarutester har genomförts vid två skilda tidpunkter, 1999 och 2003, och jämförelser av prestanda mellan dessa kan alltså göras. Den femte hypotesen lyder:

För webbaserad programuthyrning inom uppsatsens avgränsningar förbättras prestanda över tiden.

1.5 Avgränsning

Den frågeställning som presenterats ovan säger att vi är intresserade av att avgöra huruvida förutsättningarna för att bedriva webbaserad programuthyrning är goda. Vilka förutsättningar menas? Detta kommer *inte* behandlas: juridiska, ekonomiska, sociala

och/eller organisatoriska förutsättningar. Vad som *kommer* att behandlas är de tekniska förutsättningar för att bedriva webbaserad programuthyrning som anges i hypoteserna.

1.5.1 Typ av system

Vidare studeras enbart outsourcing av webbaserade applikationer, d.v.s. applikationer som körs över Internet via en webbläsare. Man kan tänka sig andra former av programuthyrning, men dessa ligger utanför avgränsningen. Systemet som undersöks används inte som ett ASP-system, men med hjälp av detta system kan de tekniska aspekterna på programuthyrning simuleras.

Frågeställningen avser medelkomplexa system. Med det menas inte enkla små interaktiva funktioner på webbplatser som t.ex. elektroniska anslagstavlor och inte heller stora affärskritiska system av typen ERP-lösningar. Applikationer som lämpar sig för webbaserad programuthyrning och faller inom ramarna för vad som menas med medelkomplexa system är t.ex. e-post, e-handelslösningar, tidredovisning, projekthanteringsstöd, resursplanering och mediabanker.

1.5.2 Kund

Vi har för avsikt att bedöma hur väl programuthyrning fungerar mot kunder med 10-150 anställda. Framförallt för några år sedan, men även fortsättningsvis, anges små och medelstora företag vara de som har mest att tjäna på att hyra program över Internet och att dessa bör utgöra programuthyrningsföretagens primära målgrupp [Susarla 01, Caldwell 98, Propson 01, Allen 02].

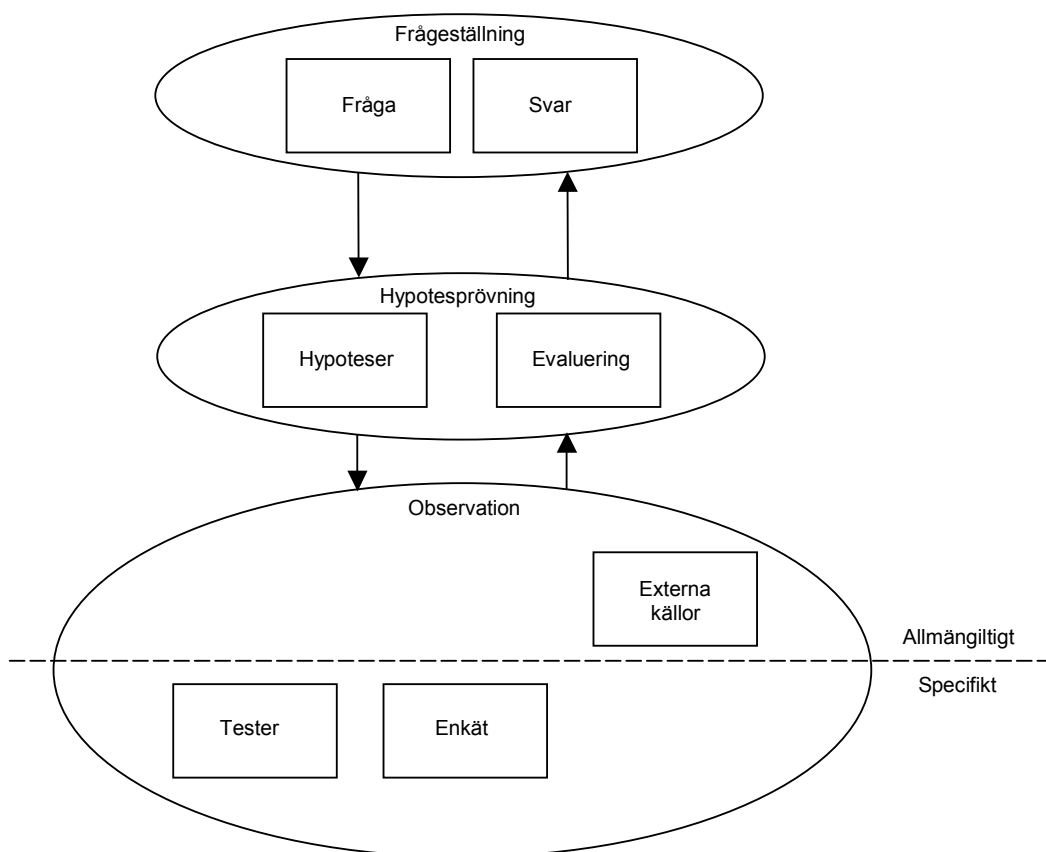
Många mindre företag har idag begränsade resurser vad gäller nätverk och internetuppkoppling. Utvecklingen går dock snabbt framåt. Vår tänkta kund representerats i denna studies mjukvarutester av en dator med fast 10 Mbit/s internetlina mot GU-net i Göteborg.

2 Metod

Arbetet med denna uppsats har varit uppdelat i fem huvudmoment. Dessa är:

- Formulering av undersökningsfrågan
- Formulering av hypoteserna
- Observation
- Evaluering av hypoteserna
- Besvarande av undersökningsfrågan

Med utgångspunkt från frågan har fem hypoteser tagits fram. Dessa prövas med hjälp av resultatet från observationer. De evaluerade hypoteserna hjälper oss sedan att besvara frågan.



Figur 2.1 – Visualisering av arbetsprocessen

Undersökningsfrågan är allmänt formulerad. Med detta menas att frågan inte bara gäller ett visst specifikt fall av webbaserad programuthyrning, utan webbaserad programuthyrning i *allmänhet* (inom de uppsatta avgränsningarna). I och med detta ska naturligtvis även svaret på frågan vara allmängiltigt. Även de fem hypoteserna är allmänt formulerade, men den data som samlas in för evaluering av hypoteserna är, med undantag av externa källor, baserade på ett *specifikt* undersökningsfall. Vi tänker oss alltså att resultaten av en specifik simulering av programuthyrning ska vara tillräckligt representativ för att allmängiltiga slutsatser ska kunna dras.

Fyra av de hypoteser vi ställer upp angrips med en positivistisk ansats. Kvantitativa tester ligger till grund för en analys som ger underlag för att styrka eller falsifiera hypoteserna. Säkerhetshypotesen kommer att angripas med en mer kvalitativ ansats, i vårt fall genom tolkning av vad andra sagt inom området. Eftersom övriga hypoteser angrips med en kvantitativ ansats hade det varit önskvärt att kunna göra så även med säkerhetshypotesen. Vi såg dock ingen praktisk möjlighet att genomföra detta.

Vi vill poängtera att valen av våra hypoteser inte är gjorda utifrån vilka potentiella evalueringsmetoder som lämpar sig, utan utifrån de tre kvalitetskriterier vi konstaterat är centrala i samband med tekniska förutsättningar för programuthyrning. Detta förklarar varför säkerhetshypotesen är med trots att metoden för dess evaluering skiljer sig från de övriga.

Även en enkätundersökning har genomförts. Denna skall främst ses som ett komplement till de kvantitativa tester som genomförs. Enkätundersökningen används inte för den faktiska evalueringen av hypoteserna utan mer för att verifiera om resultatet av evalueringen verkar hållbart.

Skälen att inte enbart använda det positivistiska arbetssättet är flera. Dels är vissa hypoteser svåra att testa med rationalistiska metoder eftersom det är svårt att klä den i siffror och för att det krävs en ganska specialiserad expertis för att genomföra testerna (vi syftar här främst på säkerhetsaspekterna), dels tror vi att det är fruktbart att använda mer än ett perspektiv.

Mer och mer övergår författare och forskare till att förespråka en blandning av metoder eftersom det leder till att fenomenet som undersöks kan betraktas ur flera olika synvinklar. Detta kallas ibland metodologisk triangulering, vilket är en begrepp lånat från navigationskonsten där man använder flera referenspunkter för att fastställa ett objekts position [Easterby-Smith 91]. Sålunda är ofta en pluralistisk ansats att föredra framför en enkelspårig.

Prof. Heiner Müller-Merbach ansluter sig till denna tanke och vädjar om en hög grad av pluralism inom vårt område. Han varnar för tendensen att välja ett synsätt, och sedan blunda för andra alternativ.

"Doesn't real professionalism in systems methodology require a pluralistic understanding of systems approaches and familiarity with a variety of approaches?"

[Müller-Merbach 94]

2.1 Evaluering av hypoteserna

Hypotesprövning är en vanlig metod inom många vetenskaper vilken går ut på att ett eller flera antaganden, hypoteser, formuleras. En hypotes är en kvalificerad gissning om ett förhållande i verkligheten och kan betraktas som ett preliminärt svar på en frågeställning [Ejvegård 96]. Detta arbetes hypoteser finns formulerade i stycke 1.4 ovan.

Genom att tolka och analysera data insamlad i experiment, fältstudier eller liknande försöker man sedan avgöra om antagandet i hypotesen stämmer. Om detta antagande är korrekt säger man att hypotesen har bekräftats. Om man kommer fram till att antagandet inte stämmer säger man att hypotesen har falsifierats.

Enligt vetenskapsteoretikern Karl Popper kan en hypotes dock aldrig slutligt bekräftas. Enligt denna skola leder ett misslyckade med att falsifiera en hypotes istället till att man korroborerar den, d.v.s. förstärker dess trovärdighet [Ejvegård 96]. Eftersom detta arbete försöker svara på en generellt ställd fråga, med hjälp av generellt formulerade hypoteser, genom studier av ett specifikt system anser vi att Poppers modell för hypotesprövning är lämplig. Därför kommer inte hypoteser bekräftas eller falsifieras utan snarare korroboreras (styrkas) eller falsifieras.

Här följer en beskrivning av hur evalueringen av hypoteserna genomförs.

2.1.1 Prestanda

Vad man vill göra för att evaluera denna hypotes är att jämföra resultaten av insamlad testdata med en siffra för vad som anses vara acceptabla uppstarts- respektive svarstider för system i allmänhet och webbaserade sådana i synnerhet. Det är naturligtvis svårt att presentera en allmängiltig siffra för vad som är acceptabelt i olika sammanhang eftersom denna typ av värden med nödvändighet är subjektiva. Vidare ställs det väldigt olika krav på olika typer av system och på vilken typ av operation som skall utföras. Styrsystemet till JAS 39 Gripen har naturligtvis betydligt högre krav på svarstider än genomförandet av en biljettbokning över Internet. Det är dessutom så att utöver det faktum att de krav som ställs varierar så varierar även användares tolerans från system till system. För att använda exemplet ovan: Användaren accepterar att en transaktion över Internet tar några sekunder, men för ett styrsystem i ett stridsplan blir en svarstid på 0,5 sekunder tydligt märkbar och irriterande.

0.1 second	This is about the limit for having the user feel that the system is reacting instantaneously, meaning that no special feedback is necessary except to display the result.
1.0 second	This is about the limit for the user's flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data.
10 seconds	This is about the limit for keeping the user's attention focused on the dialogue. For longer delays, users will want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect.

Tabell 2.1 – Olika gränser för acceptabla svarstider [Nielsen 94]

När man talar om svarstider hänvisar man fortfarande ofta till äldre studier inom människa-datorinteraktion. *"The basic advice regarding response times has been about the same for almost thirty years."* [Nielsen 94]

De siffror Nielsen anger har sitt ursprung hos R. B. Miller som tog fram dessa redan i slutet av 60-talet.

"When Miller wrote his guidelines, he was quite open in describing them as based only on his experience, and he called for experimental data that would allow for the formulation of better, empirically-based rules for setting computer response time for optimal human performance... ..these studies are still missing, for the most part... .. the literature is sadly lacking in empirical data to support the simplest assertions about how computer response time affects computer users." [Butler 83]

Ytterligare en person som ägnat sig åt problemen med generellt acceptabla svarstider är Calum Benson. Även han hänvisar till Miller och menar att 10-sekundersregeln gäller vid alla operationer som användaren förväntar sig ska ta tid [Benson 02].

Vi menar att det är rimligt att svarstider för en operation som läser eller sparar information i en databas via ett webbaserat geografiskt oberoende system är någonting som kan förväntas ta viss tid. Detta skulle alltså innebära att en övre gräns för acceptabla svarstider skulle kunna sättas till 10 sekunder. Baserat på egna erfarenheter och på signaler från användare av tidrapporteringsystemen verkar dock detta vara en alltför hög gräns i dessa sammanhang. Nielsen, som till skillnad från övriga auktoriteter vi hänvisar till är specialiserad på webbsystem, definierar operationer som tar mellan 2 och 10 sekunder som *reasonably fast operations* [Nielsen 00].

Ben Shneiderman, professor i datavetenskap vid Universitetet i Maryland och en ledande profil inom utformning av användargränssnitt, menar att svarstider för *common tasks* bör ligga under 2-4 sekunder.

“*Response times should be appropriate to the task:*

- *Typing, cursor motion, mouse selection: 50-150 milliseconds*
- *Simple frequent tasks: 1 second*
- *Common tasks: 2-4 seconds*
- *Complex tasks: 8-12 seconds”*

[Shneiderman 97]

Nielsens intervall för *reasonable fast operations* är framtaget för webbaserade system. Därför bör valet av gränsvärde ligga inom detta intervall. För att avgöra var i intervallet gränsen ska sättas vänder vi oss till Shneiderman. När vi tar ställning till Schneiders siffror vilka är generella för datorbaserade informationssystem väljer vi den övre gränsen för *common tasks* eftersom systemen som testas är webbaserade och användarnas förväntningar på svarstider därför är något lägre. Det övre gränsvärdet för svarstider sätts alltså till 4 sekunder. Rimligheten hos detta gränsvärde får även stöd av att John Bartlett anger 3,9 sekunder som ”*realistic performance parameter for response time*” för webb-lösningar [Bartlett 01].

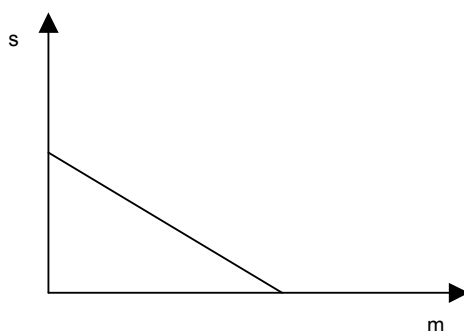
Man accepterar naturligtvis längre tider för uppstart av ett system än för en enskild operation vid användning av systemet. Dock menar Benson med stöd i Millers presentation av acceptabel prestanda att ingen operation överhuvudtaget ska ta längre än 10 sekunder [Benson 02]. Därför sätts gränsvärdet för uppstart av systemen till 10 sekunder.

Metoden som används för att evaluera prestandahypotesen innebär att man jämför erhållen testdata med de gränsvärden som satts upp. För att resultaten av ett enskilt test ska betraktas som acceptabelt krävs att den övre siffran för ett teoretiskt 90-procentigt spridningsintervall, baserat på testets medelvärde och dess standardavvikelse, ligger under uppsatt övre gräns. Detta innebär att 95% av det teoretiska utfallet ska ligga under gränsvärdet eftersom det 90-procentiga intervallet sträcker sig från 5% till 95%. Det som testas är alltså om den övre teoretiska 95%-percentilen för ett enskilt test ligger under övre uppsatt gränsvärde.

Teoretiskt spridningsintervall förutsätter normalfördelat material och beräknas genom formeln $x \pm z \cdot s$ där x är medelvärde, z är standardiserad normalfördelad variabel (standard normal deviate) för ett visst spridningsintervall och s är standardavvikelsen. Värdet på z är 1,645 vid ett 90-procentigt spridningsintervall.

Således, för att testresultatet ska betraktas som acceptabelt ska testets medelvärde plus 1,645 multiplicerat med testets standardavvikelse ligga under det gränsvärde som gäller för testet. Exempel: Antag att utfallen av ett test som mäter ett systems starttid har ett medelvärde på 6,5 sekunder och en standardavvikelse på 1,1. Formeln $6,5 + 1,645 \cdot 1,1$ ger 8,3095 sekunder. Detta innebär i teorin att vid 95% av tillfällena detta system startas tar denna operation 8,3095 sekunder eller lägre. Eftersom gränsvärdet satts till 10 sekunder anses prestanda för utförd test vara acceptabel.

Genom att placera in resultaten från enskilda testkörningar i en graf med medeltid (m) på x-axeln och standardavvikelse (s) på y-axeln kan man både se hur testerna förhåller sig till varandra var gäller tider och spridning och dessutom på ett enkelt sätt illustrera om tiderna är acceptabla genom att i grafen dra en linje som representerar gränsvärdet. Denna linje illustrerar att ju högre standardavvikelsen är desto lägre måste medelvärdet vara för att tillräckligt många av de enskilda testresultaten skall hamna under gränsvärdet. Förhållandet är dock oftast det motsatta, d.v.s. ett högre medelvärde ger ofta en högre standardavvikelse och tvärtom.



Figur 2.2 – Ytan under och till vänster om den lutande linjen representerar godtagbara tider

Av de två system som testats byggdes Light i högre grad med extern användning i åtanke. Därför är det de externa testerna av Light som är fokus i denna hypotes.

Prestandahypotesen styrks om medelvärde plus standardavvikelse multiplicerat med standard normal deviate för extern uppstartstest för Light understiger 10 sekunder samtidigt som medelvärde plus standardavvikelse multiplicerat med standard normal deviate för extern svarstidstest Light understiger 20 sekunder (20 sekunder eftersom varje test innefattar mätning av svarstid för 5 operationer, se 2.3.2, och gränsvärde för svarstider är uppsatt till 4 sekunder, $5 \cdot 4 = 20$).

2.1.2 Pålitlighet

Pålitlighetsmättet POFOD, probability of failure on demand, är ett mått på sannolikheten att systemet misslyckas med att utföra begärd operation. Till exempel betyder ett POFOD på 0,001 att i genomsnitt 1 av 1000 begärda operationer misslyckas [Sommerville 96].

$$\text{POFOD} = \text{Antal misslyckade anrop} / \text{Totalt antal anrop}$$

Ett stort antal tester genomförs endast i syfte att registrera eventuella störningar i driften. Syftet med detta är att avgöra om det finns någon skillnad i pålitlighet mellan intern och

extern drift. Med hänvisning till vad som sägs i 1.4.2 tycker vi inte att man skall behöva acceptera en märkbar skillnad i graden av pålitlighet.

För att testa pålitlighetshypotesen görs därför ett signifikanstest mellan resultaten för externa och interna pålitlighetstester. Metoden för detta är statistisk signifikanstestning av grupper med dikotoma variabler. Dikotoma variabler kan endast anta två värden, d.v.s. i detta fall kan en test resultera i att ett fel uppstod, eller att inget fel uppstod. Så kallade χ^2 -tester används för att avgöra om en testgrupp bestående av dikotoma värden skiljer sig från en annan. Testet beräknas enligt formeln nedan och resultatet skall jämföras med χ^2_α (χ^2 vid given signifikansnivå).

$$\chi^2 = \sum_{i=1}^2 \sum_{j=1}^k (o_{ij} - e_{ij})^2 / e_{ij}$$

χ^2 = erhållet χ^2 -värde

k = antal testgrupper

o_{ij} = observerat utfall

e_{ij} = förväntat utfall vid antagandet att skillnad mellan testgrupper ej föreligger

Metoden är hämtad från boken *Statistical Methods for Quality* [Miller 95].

För att skillnaderna mellan testgrupperna ska betraktas som signifikant skall det beräknade χ^2 vara högre än χ^2_α . Man väljer själv nivå för signifikanstestet. I vårt fall används en 95-procentig konfidensnivå och om då signifikant skillnad konstateras innebär detta att man kan vara 95% säker på att resultatet stämmer. Vid 95-procentig konfidensnivå är $\chi^2_{0,05} = 3,841$ då beräkningen involverar 2 testgrupper.

Pålitlighetshypotesen styrks om felfrekvensen för externa tester ej är signifikant högre än felfrekvensen för interna tester vid 95-procentig konfidensnivå. Hypotesen styrks även om skillnaden är signifikant, men POFOD för externa tester är mycket lågt, lägre än 0,0001, då pålitligheten i så fall ändå måste anses godtagbar.

2.1.3 Säkerhet

Kvantitativ data för att evaluera denna hypotes har inte tagits fram. Detta gör att den inte kan evalueras på samma sätt som övriga hypoteser. Vi kommer göra bedömningen om hypotesen håller utifrån externa källor och vad expertis på området säger. Först och främst studeras det som tydligast skiljer sig mellan intern och extern drift, d.v.s. datasäkerhet vid överföring över Internet. Eftersom potentiellt känslig data överförs via Internet vid webbaserad programuthyrning måste säkra metoder finnas för att skydda denna kommunikation. Den metod för att skydda data som studeras är SSL (Secure

Sockets Layer). Skälet till detta är att denna standard för säker överföring av data är den dominerande sedan flera år och stöds av samtliga etablerade webbläsare. Andra säkerhetsaspekter som vägs in är brandrisk, stöldrisk, hårdvaruproblem, interna läckor och företagsspionage.

För att säkerhetshypotesen ska styrkas krävs att säkerheten i regel anses vara lika god eller åtminstone inte tydligt sämre än när kunden driftar lösningen själv. Här vill vi påpeka att kvalitativ metod används eftersom säkerhetshypotesen svårligen låter sig angripas med siffror och gränsvärden på samma sätt som övriga hypoteser.

2.1.4 Systemutformning

Jämförelser mellan prestandatester för Beta och Light syftar till att avgöra om skillnaden i utformning, med avseende på t.ex. mängd och frekvens av datakommunikation mellan klient och server, har en avgörande betydelse för systemens beteende vid extern drift. Det rör sig här om att söka utreda om två system som båda fungerar bra vid intern drift ändå kan ha olika förutsättningar för extern drift.

Systemutformningshypotesen styrks om prestandahypotesen styrks för Light samtidigt som Beta klarar av de uppsatta gränsvärdena för prestanda vid intern drift, men inte vid extern. Om så är fallet skulle det visa att framgång vid intern drift inte automatiskt betyder att systemet lämpar sig för programuthyrningsförhållanden. Evalueringen av denna hypotes baserar sig på prestandahypotesen. Vi upprepar inte här detaljerna för hur prestandahypotesen evalueras, utan hänvisar till 2.1.1.

2.1.5 Tiden

Jämförelsen mellan tester som utförts 1999 och 2003 syftar till att avgöra om en märkbar förbättring av prestanda har skett under denna tid.

Metoden för detta baseras på statistisk signifikantestning. Så kallade t-tester används för att avgöra om skillnader av medel för tester utförda 2003 jämfört med medel för motsvarande tester 1999 är statistiskt signifikanta. Testet baseras på de två stickprovens storlek, standardavvikelse samt medelvärde och beräknas enligt formeln nedan.

$$t = (x_1 - x_2) / \sqrt{((s_1^2 / n_1) + (s_2^2 / n_2))}$$

t = t-värde

x_1 = medel för test 1

x_2 = medel för test 2
 s_1 = standardavvikelse för test 1
 s_2 = standardavvikelse för test 2
 n_1 = stickprovets storlek test 1
 n_2 = stickprovets storlek test 2

Metoden är hämtad från boken *Statistical Methods for Quality* [Miller 95].

För att avgöra om medelvärdet för test 1 är signifikant lägre än medelvärdet för test 2 skall det beräknade t-värdet vara lägre än det negativa värdet av det som kallas standardiserad normalfördelad variabel (standard normal deviate, z). Man väljer själv nivå för signifikanstestet. I vårt fall används en 95-procentig konfidensnivå och om då signifikant skillnad konstateras innebär detta att man kan vara 95% säker på att resultatet stämmer. Vid 95-procentig ensidig konfidensnivå är $z = 1,645$ då stickprovets storlek är större än 30. Det är naturligtvis alltid önskvärt med ett så stort stickprov som möjligt, men för att testa denna hypotes krävs färre tester än för övriga hypoteser. Därför är antalet tester genomförda 2003 färre än 1999.

Tidshypotesen styrks om medelvärdet för tester utförda 2003 är signifikant lägre än medelvärdet för motsvarande tester 1999.

2.2 Mjukvarutestning

“When you can measure what you are speaking about and express it in numbers, you know something about it, but when you cannot measure it, when you cannot express it in numbers, your knowledge in of a meagre and unsatisfactory kind.”

– Lord Kelvin [Wallmüller 94]

Trots att kvantitativa tester av mjukvarukvalitet länge förekommit så är det fortfarande en svår uppgift, bl.a. eftersom det fortfarande i mångt och mycket saknas självklara tekniker och måttstockar. Ändå är dessa tester en grundläggande förutsättning för kontroll av kvalitetsaspekter [Wallmüller 94].

Den litteratur vi funnit på området talar i princip uteslutande om mjukvarutestning i syfte att bedriva utveckling. Detta skiljer sig delvis från det vi är intresserade av eftersom vi samlar in testresultat i samband med forskning snarare än mjukvarukonstruktion, samt att vi studerar ett färdigt system snarare än ett under utveckling. Vi tittar ändå på en definition som står att finna i boken *The Complete Guide to Software testing*:

Definition of testing

<p><i>Testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results.</i> [Hetzel 98]</p>

Om man frågar ett antal yrkesutövare inom IT-branschen vilket syfte mjukvarutestning har så får man antagligen lite olika svar i stil med dessa [Hetzel 98]:

- Kontrollera att systemet uppfyller specifikationskraven
- Hitta buggar
- Avgöra om systemet är färdigt för användning
- Försäkra sig om att systemet fungerar
- Övertyga sig själv att jobbet är färdigt
- Avgöra om systemet agerar korrekt
- Förstå vilka gränser systemet har prestandamässigt
- Ta reda på vad systemet inte kan göra
- Evaluera systemets kapacitet

Samtliga dessa synsätt är korrekta och meningsfulla, men endast de tre sista (eventuellt även det sjätte) beskriver sådant som vi är intresserade av. Anledningen till detta är att vi inte utvecklar mjukvara utan istället utvärderar hur ett redan existerande system presterar under olika förhållanden. Detta innebär att definitionen ovan är lite för bred för våra syften. Om vi kortar ner den till ”Testing is any activity aimed at evaluating an attribute or capability of a program or system” får vi en definition som passar vårt arbete på ett bättre sätt.

Principen för att genomföra ett test är egentligen ganska enkel. Det hela går ut på att först välja ut det man vill mäta. Det kan t.ex. vara något attribut eller funktion hos en mjukvara. Sedan utformar man ett eller flera experimentfall som avslöjar något om det man ville mäta, och därefter genomför man experimenten och observerar resultaten i förhållande till vad man hade förväntat sig. Svårigheterna i detta förfarande ligger främst i att välja rätt saker att testa och att veta vad man kommer kunna förvänta sig av testresultaten.

Man brukar i litteraturen skilja på två perspektiv när det gäller testning, nämligen *black-box* och *white-box* [Hetzel 98]. Med *black-box*-testning menas att man studerar de yttre egenskaperna för föremålet som testas. Man bryr sig alltså inte så mycket om vad som händer under ytan utan kapslar in systemet och är bara intresserad av egenskaper hos utdata i förhållande till indata. *White-box*-tekniken å andra sidan innebär att testfallen är konstruerade för att testa den interna logiken i ett system.

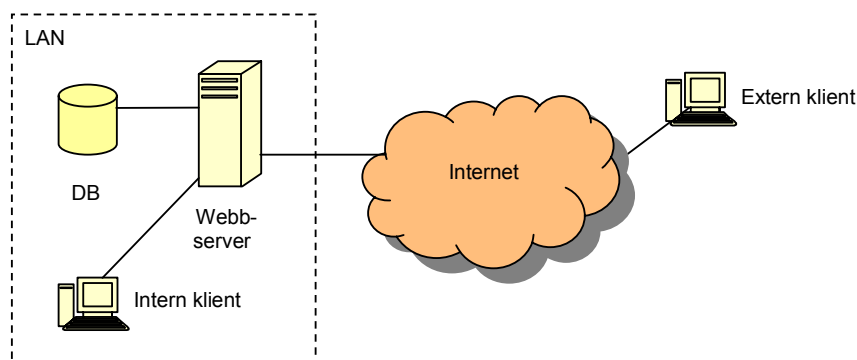
I vårt fall rör det sig om *black-box*-principen. Systemet är från början givet, och vi bryr oss i detta fall inte om vad som händer under ytan (även om vi ju råkar veta det ganska väl ändå). Vår grundprincip att mäta beteendet hos vårt system under två olika förutsättningar. Vi kör tester mot systemet dels innanför väggarna på Citat Solutions och dels över Internet. Om vi använder *black-box*-metaforen så handlar det i princip om att inkludera olika mycket i vår svarta låda. I och med att vi kapslar in en större mängd infrastruktur (d.v.s. Internet) i det ena fallet än i det andra så förväntar vi oss att

testresultaten ska utfalla olika. I avsnitt 2.3 finns en utförlig beskrivning av hur vi byggt upp testerna och i vilka syften.

Det faktum att objektet för våra studier och tester är ett system som vi själva utvecklat skulle kunna ha en viss betydelse i vårt arbete. Vid utvecklingstestning är det ofta eftersträvansvärt att utformning och genomförande av mjukvarutestningen görs av någon oberoende. Den som utför testerna måste *vilja* göra det, och så är ju kanske inte alltid fallet om man ska testa och söka efter brister i något man själv producerat. Detta torde dock inte innebära något större problem i vårt fall, vilket åter igen har att göra med att vi inte utvecklar utan forskar. Vi letar inte efter fel i det vi gjort, utan efter skillnader i systemets beteende under olika förutsättningar.

2.3 Tester

Nedan följer en beskrivning av de tester som utförts. Andledningen till att vi till största delen använder kvantitativa tester för att utvärdera valda kvalitetskriterier är att detta i litteraturen ofta rekommenderas för kvalitetsstudier [Sommerville 96]. Vi har alltså valt att lägga tyngdpunkten på kvantitativa metoder i vårt arbete att samla in material. Den grundläggande idén har varit att utgå från mjukvarukvalitetskriterierna prestanda och pålitlighet och utifrån dessa genomföra tester som genererar kvantitativ data. Vill man t.ex. utvärdera prestanda i olika sammanhang så utförs fördefinierade operationer ett stort antal gånger samtidigt som vi mäter den tid (i millisekunder) det tar.



Figur 2.3 – Översiktsbild av testmiljön

Systemen som testats var installerade på en webserver och en databasserver på Citat Solutions göteborgskontor (se *Bilaga C*). Klientdatoren var en PC med en webbläsare. Denna PC användes för samtliga tester med undantag av de som gjorts 2003. Det enda

som skilde de interna och externa testerna åt med avseende på teknisk miljö var infrastrukturen mellan klientdatorn och webbservern samt att kommunikation för de externa testerna skedde krypterat. För att kunna göra jämförelser mellan interna och externa tester har alltså ansträngningar gjorts för att hålla nere antalet variabler.

Testerna från 2003 genomfördes på nyare, och därmed snabbare, maskiner. Detta gäller både klientdator och servrar. Även infrastrukturen ser annorlunda ut med högre bandbredd hos leverantören och eventuella skillnader som kan finnas på det publika Internet. Detta är ett avsteg från principen att testa på identiska maskiner som tidigare gällt för att minska och kontrollera antalet variabler som kan påverka testresultaten. Vi anser dock att detta är i sin ordning, ja till och med nödvändigt, för att på ett korrekt sätt kunna evaluera hypotesen som berör förbättringar av förutsättningarna över tiden. Hade testerna gjorts på samma hårdvara som 1999 hade detta inte avspeglat det faktum att förutsättningarna förändras.

För att genomföra ett stort antal tester har testprocessen automatiserats. Ett script på klientdatorn simulerar en användares interaktion med systemen. Detta script har programmerats att utföra en sekvens operationer. Denna sekvens ser olika ut beroende på vilken typ av test som genomförs. Sekvensen upprepas ett antal gånger för varje test och tiden då sekvensen påbörjas respektive avslutas registreras i en databas. Innehållet i denna databas utgör sedan den testdata som presenteras i 3.1 (se även *Bilaga A*).

Testerna är indelade i 5 typer. De fyra första (A-D) berör prestanda och testresultaten från dessa används alltså för att evaluera prestandahypotesen. Test A och C är simulerad upp-

Typ av test	Beskrivning
A	Uppstartstest för Beta
B	Uppstartstest för Light
C	Svarstidstest för Beta
D	Svarstidstest för Light
E	Pålitlighetstest

Tabell 2.2 – Beskrivning av testtyper

start respektive användning (d.v.s. en sekvens operationer) av Beta. Test B och D är motsvarande tester för Light. Med hjälp av jämförelser mellan dessa kan evaluering av systemhypotesen göras. Den femte (E) är en pålitlighetstest vars resultat syftar till att evaluera pålitlighetshypotesen.

Samtliga tester är genomförda dels internt, det vill säga på Citat Solutions göteborgskontor, och dels externt, krypterat över internet. Dessutom har varje typ av test gjorts med ett litet antal tester via modem, detta för att kunna evaluera att avgränsningen gällande kundens infrastruktur var rimlig.

<i>Typ av test</i>	<i>Plats</i>	<i>Antal körningar</i>
A	Internt	500
A	Externt	500
A	Externt (modem)	50
B	Internt	500
B	Internt (ssl)	500
B	Externt	500
B	Externt (modem)	50
C	Internt	500
C	Externt	500
C	Externt (modem)	50
D	Internt	500
D	Externt	500
D	Externt (modem)	50
E	Internt	50 000
E	Externt	50 000
E	Externt (modem)	50
B	Internt (2003)	100
B	Externt (2003)	100
D	Internt (2003)	100
D	Externt (2003)	100

Tabell 2.3 – Planerade tester

2.3.1 Test A och B

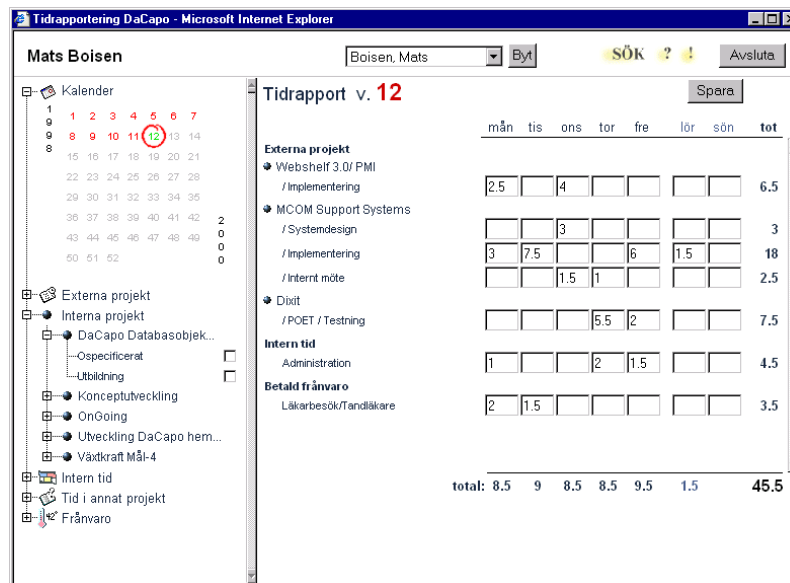
Test A och B simulerar och registrerar tider för uppstart av Beta respektive Light. Tiden för varje uppstart mäts i millisekunder och utgörs av differensen mellan starttid och sluttid.

Starttiden loggas då scriptet aktiverar hyperlänken som startar systemet. Sluttiden loggas då datahämtning och grafisk presentation är genomförd och systemet därmed är klart för användning på klientdatorm. Därefter stänger scriptet ner systemet och sekvensen startas igen i en iterativ process.

2.3.2 Test C och D

Test C och D simulerar och registrerar tider för en serie operationer som utförs mot ett system som redan är uppstartat. Operationerna som genomförs är valda för att simulera en ”normal användning” av systemen med uppdatering av fält, tillägg av projekt att rapportera tider för, sparande av rapport, o.s.v. Syftet med dessa tester är att mäta systemets svarstider till skillnad från test A och B vars syfte är att mäta uppstartstider.

Tiden för varje fullständigt genomförd serie operationer mäts i millisekunder och utgörs av differensen mellan starttid och sluttid. Starttiden loggas omedelbart före punkt ett i sekvensen som beskrivs nedan. Sluttiden loggas när sista punkten i sekvensen är avslutad. Därefter återställs systemet och sekvensen startas igen i en iterativ process.



Figur 2.4 – Tidrapporteringsklient Beta

Sekvens:

1. Siffran 4, för onsdagen, i rapportens översta rad ändras till 3.
2. Siffran 0,5 skrivs i torsdagsfältet för rapportens andra rad.
3. Projektet Dixit tas bort från tidrapporten.*
4. Siffran 0,5 tas bort från torsdagsfältet för rapportens andra rad.
5. Rapporten sparas.*
6. Siffran 3, för onsdagen, i rapportens översta rad ändras till 4.
7. Projektet Dixit läggs till i rapporten.*
8. Siffran 5,5 skrivs i torsdagsfältet för projektet Dixit.
9. Rapporten sparas.*
10. Rapporten ”lämnas in”.*

* Operationer som involverar kommunikation mellan klient och server.

Sekvensen innehåller tio moment, varav fem innebär manipulation av data som endast utförs på klienten. De övriga fem, markerade med asterix, utgör operationer där begreppet ”svarstid” blir relevant eftersom användaren här måste vänta på systemet att utföra en åtgärd. Tiden för varje enskild operationerna som involverar kommunikation med servern har inte loggats. Genom att dela den totala tiden för en sekvens med fem erhålls en genomsnittlig tid för en enskild operation i sekvensen.

2.3.3 Test E

I test E utförs en upprepad enskild operation som kräver kommunikation mellan klient och server. Den operation som simuleras är att spara rapporten från en redan uppstartad klient för Light. Ett mycket stort antal operationer utförs efter varandra och klienten loggar eventuella misslyckade operationer. På detta sätt kan man mäta i vilken grad systemet svarar utan att fel uppstår. Syftet är att avgöra systemets pålitlighet, inte vad som kan bli följden av ett misslyckande. *"Reliability metrics are all based around the probability of system failure, they cannot take account the consequences of such a failure"* [Sommerville 96]. Det finns ett flertal kvantitativa mått för att mäta olika aspekter av pålitlighet. De utförda testerna ger underlag för att beräkna probability of failure on demand (POFOD) som beskriver sannolikheten att systemet misslyckas att utföra en operation på begäran av användaren [Sommerville 96].

2.4 Enkät

En enkätundersökning genomfördes under våren 1999. Syftet med enkäten är att mäta användares personliga och subjektiva upplevelser av systemen. Vid sidan av de tester som genomförs i syfte att evaluera hypoteserna kan resultatet av enkätundersökningen användas som en avstämning gentemot resultaten av testerna. Med utgångspunkt i huruvida resultaten från enkätundersökningen sammanfaller med resultaten av testerna kan man förhoppningsvis säga något om resultatens pålitlighet.

Vid utformning av enkätundersökningen har vi försökt följa riktlinjer som beskrivits av Rolf Ejvegård i boken *Vetenskaplig metod* [Ejvegård 96].

Enkätundersökningen var webbaserad och genomfördes på Citat Solutions intranät. Samtliga frågor hade fasta svarsalternativ och svaren lagrades i en databas, detta för att ge möjlighet till kvantitativ analys.

"Conducting a survey is often a useful way of finding something out, especially when 'human factors' are under investigation. Although surveys often investigate subjective issues, a well-designed survey should produce quantitative, rather than qualitative, results." [Boone]

Samtliga 46 användare av systemen ombads att svara på enkäten. Detta innebär att urvalet utgör hela populationen av användare för de undersökta systemen. I enkätens inledande frågor får användaren ange varifrån systemen används, detta i syfte att kunna göra nödvändiga jämförelser mellan olika användargrupper, främst interna och externa. För att uppnå en så hög grad av validitet som möjligt har enkätens frågor och inledande instruktion utformats för att understryka att det är subjektiva värderingar som efterfrågas, exempelvis "Hur upplever Du prestanda för tidrapportering Beta?".

Hypoteserna har stått i centrum vid val av frågor som ingår i enkäten. De aspekter som behandlas är:

- Prestanda
- Pålitlighet
- Säkerhet
- Jämförelse mellan system

Vid utformningen av enkäten har ambitionen varit att undvika ledande svarsalternativ. Så gott som alla frågor har samma antal svarsalternativ (6) och skalorna går gradvis från negativa till positiva omdömen. Avslutningsvis finns för varje fråga ett alternativ som ger möjlighet att inte besvara frågan. För att undvika svårigheter med att besvara respektive fråga och med att analysera resultatet har svarsalternativen utformats så att de utesluter varandra, detta för att undvika situationer där flera svarsalternativ känns ”rätt”. Enkätundersökningens frågor återfinns i *Bilaga B*.

Vid tolkning av enkätens resultat måste man ta hänsyn till bortfall. Det finns två typer av bortfall, enkätbortfall och internt bortfall. Det förstnämnda är ett mått på hur många som överhuvudtaget inte besvarar enkäten och det sistnämnda beskriver det bortfall som uppstår då en enskild fråga inte besvaras. Storleken på bortfallet påverkar vilka analysmetoder som kan användas och vilka slutsatser som kan dras av resultatet.

3 Resultat

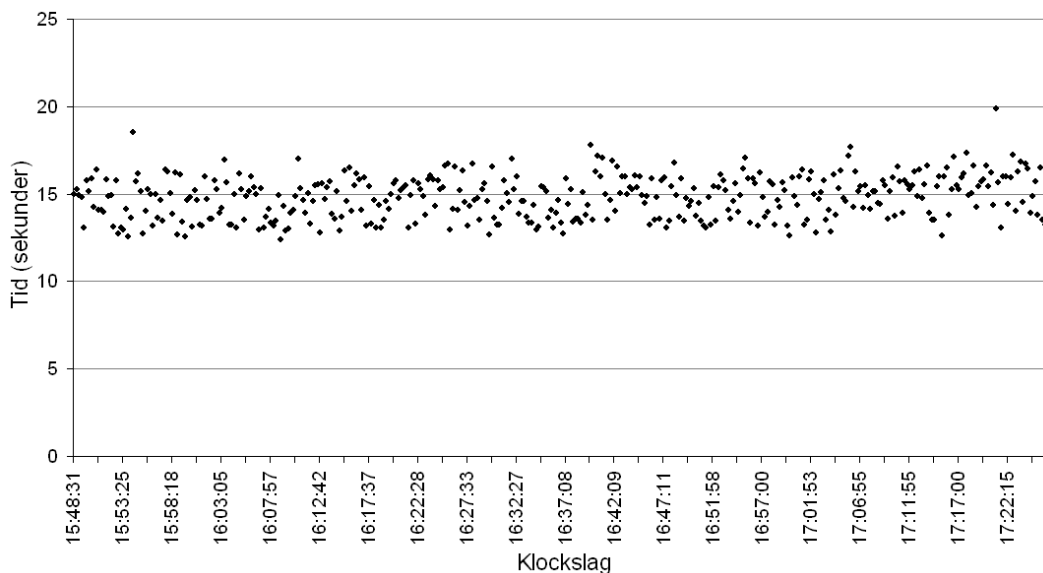
3.1 Mjukvarutester

Totalt genomfördes 29 tester. 3 tester av typen A, 6 av B, 3 av C, 5 av D och 12 av E. Detta betyder att samtliga planerade tester har genomförts. Testerna har totalt producerat 119 890 testvärden. De tester som gjordes 1999 genomfördes under en 5-veckorsperiod. Testerna 2003 genomfördes under en dag i januari.

Testnr	Typ av test	Plats	Antal körningar	Datum
501	A	Internt	500	1999-05-11
307	A	Externt	400	1999-04-24
323	A	Externt (modem)	10	1999-05-10
504	B	Internt	500	1999-05-13
503	B	Internt (ssl)	500	1999-05-13
308	B	Externt	500	1999-04-25
321	B	Externt (modem)	50	1999-05-10
517	C	Internt	500	1999-05-23
300	C	Externt	300	1999-04-22
324	C	Externt (modem)	5	1999-05-10
520	D	Internt	500	1999-05-23
305	D	Externt	400	1999-04-23
322	D	Externt (modem)	50	1999-05-10
502	E	Internt	10 000	1999-05-11
506	E	Internt	10 000	1999-05-14
507	E	Internt	10 000	1999-05-14
509	E	Internt	10 000	1999-05-15
510	E	Internt	10 000	1999-05-16
513	E	Internt	5 000	1999-05-19
315	E	Externt	5 000	1999-04-27
316	E	Externt	10 000	1999-04-27
317	E	Externt	10 000	1999-04-28
318	E	Externt	10 000	1999-04-28
319	E	Externt	10 000	1999-04-29
1000	E	Externt (modem)	50	1999-05-27
20033	B	Internt	100	2003-01-12
20031	B	Externt	100	2003-01-12
20034	D	Internt	100	2003-01-12
20032	D	Externt	100	2003-01-12

Tabell 3.1 – Genomförda tester

Samtliga testvärden för tester av typ A, B, C och D återfinns i *Bilaga A – Resultat av mjukvarutester* (ej modemtester). Testvärdena redovisas i spridningsdiagram enligt figur 3.1, där x-axeln anger klockslag och den loggade tiden (sekunder) för operationerna läses av på y-axeln. I exemplet nedan kan man följaktligen se att operationerna ligger runt 15 sekunder och de flesta värdena ligger mellan 12 och 17 sekunder. Varje operation i testet representeras av en punkt. Således återfinns 400 punkter i grafen för test nummer 307.

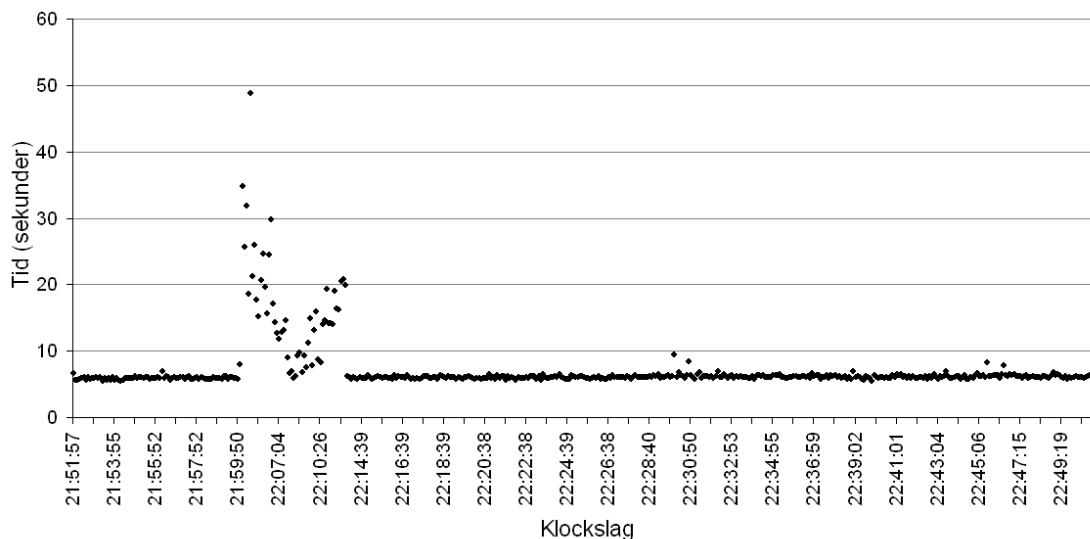


Figur 3.1 – Test 307. Typ A Uppstart Beta externt

3.1.1 Justering av testdata

När arbetet med testdata kommit så långt att den visualiserats i spridningsdiagram, och vissa slutsatser av resultatet kunde börja dras, upptäcktes att några av testerna verkade innehålla onormala testvärden. Detta yttrade sig i regelbundna mycket tydliga extremvärden. Vid en jämförelse av de tester som uppvisade detta fenomen framkom att ansamlingen av extremvärden inföll vid samma tidpunkter på dygnet, nämligen klockan 22:00, 23:00, 23:40 samt 04.00. Vi insåg snart att dessa med stor sannolikhet orsakats av backup-körningar på servrarna, och efter en kontroll med Citat Solutions IT-avdelning kunde detta bekräftas.

Denna upptäckt ställde till med problem eftersom extremvärdena var så avvikande att resultatet skulle ge en felaktig bild av systemens normala prestanda. Nu kan man hävda att dessa extremvärden är en del av de tekniska förutsättningar som testats och således borde vara kvar. Men i verkligheten skulle en leverantör av en programuthyrningstjänst se till att hantera detta problem på lämpligt sätt, exempelvis genom att lägga denna typ av störande aktivitet vid tidpunkter då kunden inte använder systemet. Frågan är förvisso inte trivial. Om det ställs krav på att systemet ska vara lika tillgängligt dygnet runt krävs att detta problem hanteras, men denna problematik omfattas inte av vår undersökning.



Figur 3.2 – Exempel på effekterna av backup-körning (Test 501, uppstart av Beta internt)

Måtten medelvärde och standardavvikelse används både vid jämförelser mellan tester och vid evaluering av prestandahypotesen. Dessa mått är känsliga för extremvärden, särskilt stora sådana [Gunnarsson 02]. För att överhuvudtaget kunna göra analyser av och jämförelser mellan tester blev det därför nödvändigt att bortse från de extremvärden som tydligt kunde kopplas till backup-körningar. Detta har varit ett förhållandevis okomplicerat ingrepp då tidpunkterna för backup-körningarna är kända. Inga testvärden eller grupper av testvärden som inte självklart kunnat kopplas till kända störningar har bortses ifrån.

Spridningsdiagrammet för test 501 (figur 3.2) är det tydligaste exemplet på att tester kraftigt påverkas av en backup-körning mellan kl 22:00 och 22:13. Medelvärdet för denna test var före justering 7,12 med en standardavvikelse på 4,05. Efter justering var medel 6,06 och standardavvikelsen 0,32. De senare värdena ger en rättvisare bild av stabiliteten i systemets uppförande under normala omständigheter.

Följande justeringar har gjorts:

Test 300: 22.00-22.13, 23.00-23.06 samt 23.40-23.55

Test 308: 04.00-04.05

Test 501: 22.00-22.13

Fortsättningsvis redovisas justerade värden. I bilagan för testresultat redovisas ojusterade testvärden i form av spridningsdiagram. I tabell 3.2 presenteras en sammanställning av samtliga testkörningar.

Testnr	Typ	Plats	Medel	Median	SA	95% central range	95% spridningsintervall vid normalfördelning	Antal fel
501	A	Internt	6,06	6,03	0,32	5,65 – 6,63	5,44 – 6,68	0
307	A	Externt	14,83	14,91	1,22	12,76 – 17,05	12,43 – 17,22	0
323	A	Externt (modem)	147,47	145,11	5,12	143,42 – 157,84	137,43 – 157,51	0
504	B	Internt	2,32	2,35	0,19	2,02 – 2,61	1,95 – 2,69	0
503	B	Internt (ssl)	2,16	2,16	0,09	2,00 – 2,36	1,98 – 2,34	0
308	B	Externt	4,35	4,20	0,39	3,89 – 5,28	3,58 – 5,11	0
321	B	Externt (modem)	24,67	23,94	1,66	23,51 – 28,69	21,41 – 27,93	0
517	C	Internt	16,85	16,81	0,33	16,41 – 17,47	16,2 – 17,51	0
300	C	Externt	46,50	45,98	5,53	39,89 – 57,86	35,67 – 57,34	0
324	C	Externt (modem)	385,19	378,49	23,52	358,75 – 410,40	339,1 – 431,29	0
520	D	Internt	7,77	7,70	0,29	7,49 – 8,37	7,20 – 8,34	0
305	D	Externt	17,20	17,15	0,80	15,84 – 18,84	15,64 – 18,76	0
322	D	Externt (modem)	107,75	106,86	4,13	103,73 – 119,91	99,66 – 115,84	0
502	E	Internt	n/a	n/a	n/a	n/a	n/a	0
506	E	Internt	n/a	n/a	n/a	n/a	n/a	0
507	E	Internt	n/a	n/a	n/a	n/a	n/a	0
509	E	Internt	n/a	n/a	n/a	n/a	n/a	0
510	E	Internt	n/a	n/a	n/a	n/a	n/a	0
513	E	Internt	n/a	n/a	n/a	n/a	n/a	0
315	E	Externt	n/a	n/a	n/a	n/a	n/a	0
316	E	Externt	n/a	n/a	n/a	n/a	n/a	0
317	E	Externt	n/a	n/a	n/a	n/a	n/a	1
318	E	Externt	n/a	n/a	n/a	n/a	n/a	0
319	E	Externt	n/a	n/a	n/a	n/a	n/a	0
1000	E	Externt (modem)	n/a	n/a	n/a	n/a	n/a	0
20033	B	Internt	1,83	1,82	0,08	1,72 – 2,00	1,68 – 1,98	0
20031	B	Externt	2,94	2,87	0,27	2,58 – 3,70	2,41 – 3,46	0
20034	D	Internt	6,88	6,80	0,21	6,60 – 7,39	6,48 – 7,29	0
20032	D	Externt	11,43	11,35	0,65	10,75 – 12,84	10,22 – 12,76	0

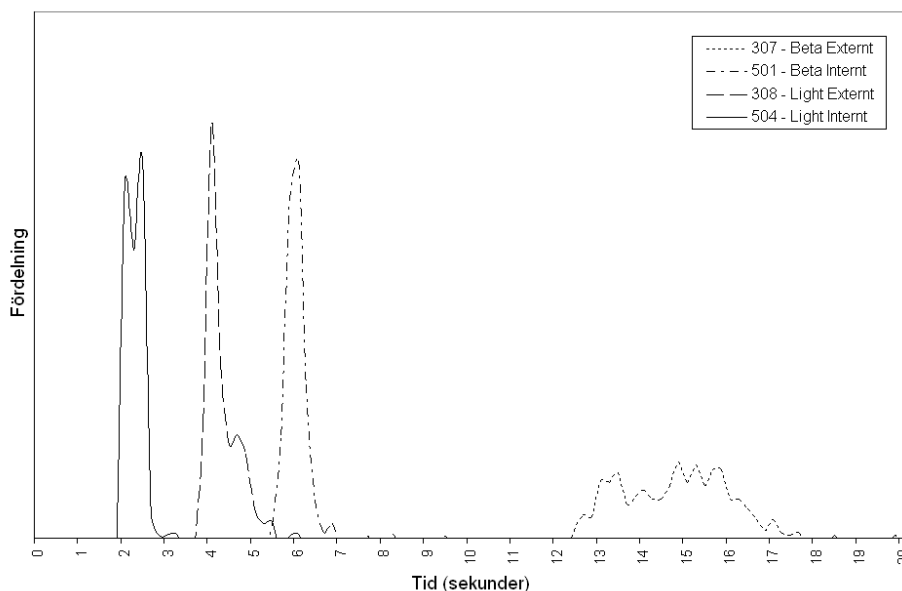
Tabell 3.2 – Sammanställning av testutfall

Om man tittar på fördelningskurvorna över testresultat, exempelvis figur 3.8, så kan man se en tendens till att kurvorna sträcks ut något i det övre intervallet (s.k. ”long tail”). Detta visar att utfallet av testerna inte är helt normalfördelat, vilket är ganska vanligt vid mätning av tider [Sevcik 03]. Denna tendens är dock ganska liten och om man i tabell 3.2 jämför 95-procentigt central range, som alltså anger de faktiska yttre gränserna för 95% av respektive tests utfall, med det beräknade teoretiska 95-procentiga spridningsintervall vid normalfördelning, som alltså anger de yttre gränserna för det intervall inom vilket 95% av ett oändligt antal testutfall återfinns, finner man att skillnaderna är ganska små. Slutsatsen blir att utfallet är tillräckligt normalfördelat för att vi ska kunna använda statistiska metoder som förutsätter normalfördelning.

3.1.2 Test A och B, uppstartstider

Testunderlaget från 1999 är betydligt mer omfattande än för de kompletterande tester som gjordes år 2003. Här presenteras först en sammanställning av testerna från 1999 och sedan en jämförelse mellan utfallen 1999 och 2003. Den första delen om testerna 1999 kommer att fungera som underlag för evaluering av prestandahypotesen samt system-

utförningshypotesen, medan den andra delen ger underlag för evaluering av tids-hypotesen.

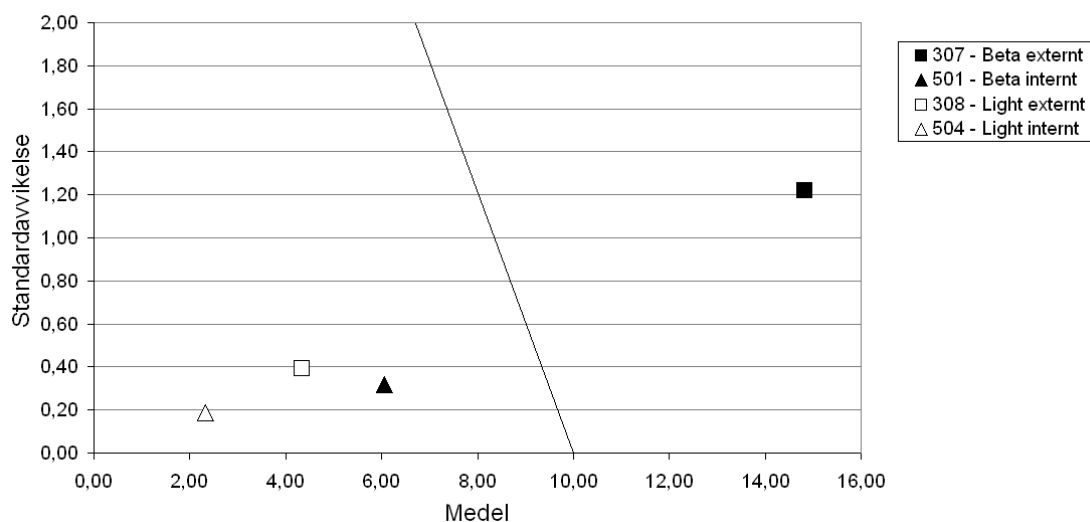


Figur 3.3 – Uppstartstester 1999

Figur 3.3 visar fördelning av starttider för båda systemen (Beta och Light) dels internt och dels externt över Internet. För tre av testkörningarna, nämligen Light internt samt externt och Beta internt, är utfallet väl samlat. Starttiderna ligger på mellan två och sju sekunder för samtliga dessa tre och spridningen för varje test är relativt låg. Uppstart för Beta externt utmärker sig dock med höga starttider och betydligt större spridning.

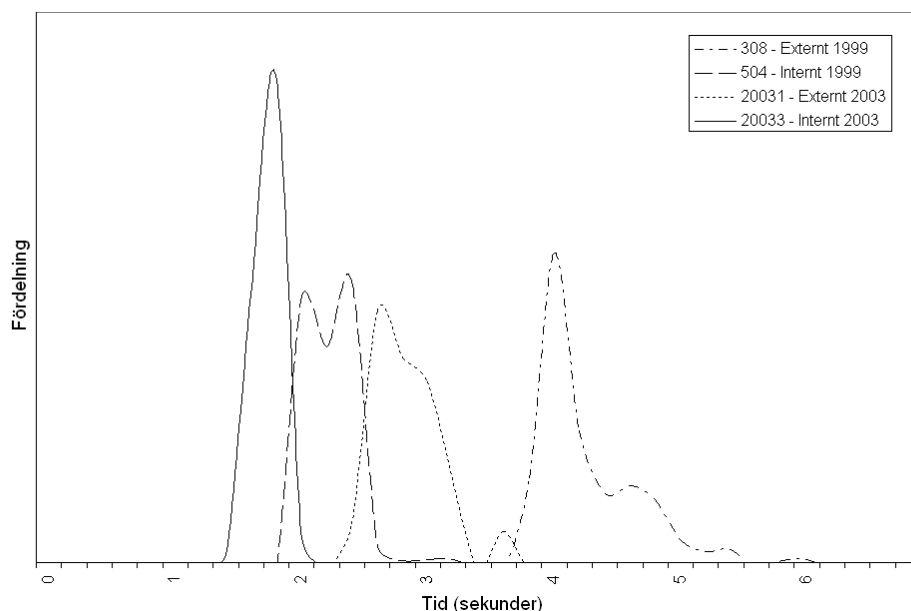
Figur 3.4 säger mer om resultatet av testkörningarna. Testresultaten är utplacerade i grafen där x-axeln anger medeltid för uppstarterna och y-axeln anger standardavvikelsen. Ju längre åt höger ett test är placerat desto längre medeltid och ju högre upp det är placerat desto högre är spridningen. Det är alltså önskvärt att testerna placerar sig så långt ner och så långt åt vänster som möjligt. Linjen som är dragen över grafen är det gränsvärde som satts upp för acceptabla uppstartstider (se 2.1.1).

Även i denna graf kan man tydligt se att Beta externt kraftigt avviker från de övriga testerna. Faktum är att detta test hamnar på fel sida linjen och alltså faller utanför vad vi definierat som acceptabelt. Skillnaderna mellan interna och externa tester för Beta är dessutom mycket större än motsvarande skillnad för Light.



Figur 3.4 – Standardavvikelse och medeltider vid uppstart av system 1999

Skillnader i prestanda mellan tester utförda 1999 och 2003 är tydliga. Det räcker med en snabb titt på figur 3.5 för att konstatera att uppstartstiderna är betydligt lägre för 2003, framförallt för de externa testerna. Vi vill ändå applicera statistiska metoder för att avgöra huruvida skillnaderna är signifikanta. För detta ändamål har så kallade t-tester utförts. T-tester visar om medelvärdena från två populationer signifikant skiljer sig från varandra.



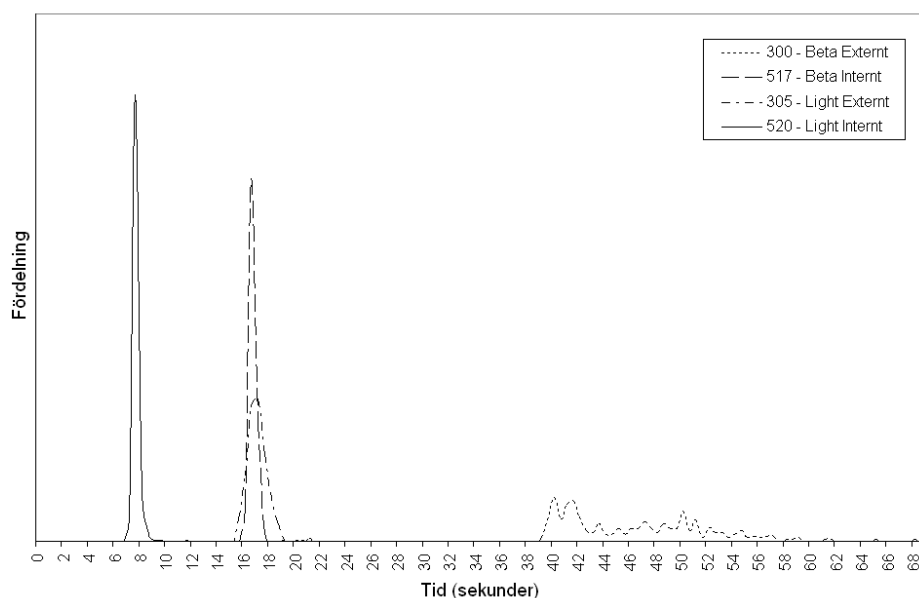
Figur 3.5 – Uppstart av Light 1999 och 2003

Vi sätter signifikansnivån till 0,05 vilket innebär att vi kan vara säkra på utfallet av t-testen till 95%. Denna test görs endast för de externa testerna eftersom det endast är dessa som har betydelse för evalueringen av tidshypotesen.

T-värdet för test 20031 i förhållande till test 308 blev -43,9 (enligt formeln i metodavsnitt 2.1.5). Detta ska jämföras med $-z_{0,05}=-1,645$. Då $-43,9 < -1,645$ kan det konstateras att skillnaden är statistiskt signifikant.

3.1.3 Test C och D, svarstider

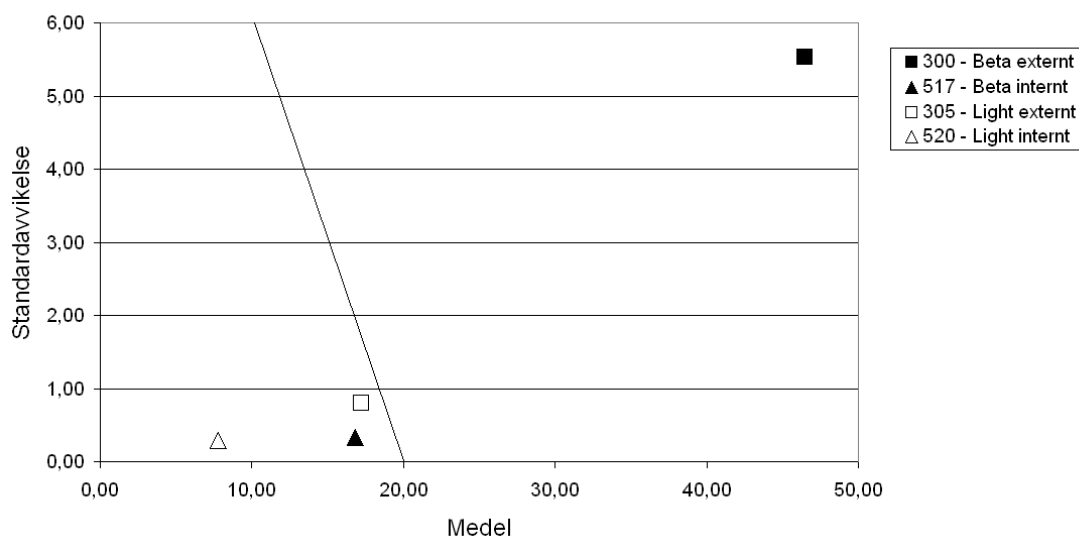
Figur 3.6 visar fördelning av testresultat över svarstidstester för båda systemen (Beta och Light) dels internt och dels externt över Internet. Varje testresultat omfattar flera operationer varav 5 innefattar kommunikation mellan klient och server (se 2.3.2).



Figur 3.6 – Fördelning av testresultat för svarstidstester 1999

Precis som för motsvarande figur för uppstartstider är det även här Beta externt som avviker. När man studerar grafen slås man av det faktum att Light ser ut att bete sig ungefär likadant internt som externt även om prestanda naturligtvis är sämre externt, medan Beta verkar bete sig helt annorlunda vid internt respektive extern användning. Beta uppvisar vid extern drift ett mycket högt medelvärde, men framförallt en spridning som inte alls liknar den som de andra tre testerna uppvisar.

I figur 3.7 markeras medeltid och standardavvikelse för svarstidstesterna på samma sätt som i figur 3.4 för uppstartstesterna ovan. Resultatet blir ungefär detsamma, d.v.s. båda Light-testerna och den interna Beta-testen är relativt samlade medan Beta externt återfinns betydligt längre upp till höger i figuren.



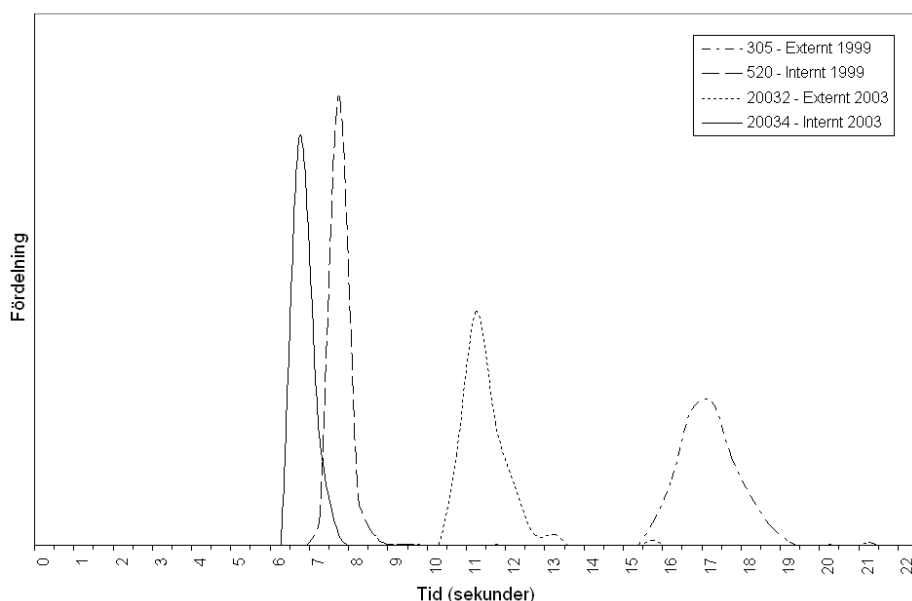
Figur 3.7 – Standardavvikelse och medeltider för svarstidstester 1999

Linjen som är dragen över grafen är det gränsvärde som satts upp för acceptabla svarstider multiplicerat med fem. Skälet att värdet multiplicerats med fem är att gränsvärdet gäller svarstiden för en operation medan medelvärdet grundas på tester som vardera består av fem tidskrävande operationer.

Medelvärdet per svarstid för respektive test är: 9,30 sekunder för test 300, 3,37 för test 517, 3,44 för test 305 och 1,55 för test 520. Dessa siffror visar att medelvärdet för tre av testerna ligger under de 4 sekunder som satts upp som övre acceptabel gräns för svarstider.

Skillnader i prestanda mellan tester utförda 1999 och 2003 är tydliga även för svarstidstester. Ett t-test med signifikansnivån 0,05 ger följande resultat.

T-värdet för test 20032 i förhållande till test 305 blev $-75,6$. Detta ska jämföras med $-z_{.05} = -1,645$. Då $-75,6 < -1,645$ kan det konstateras att skillnaden är statistiskt signifikant. Åter kan konstateras att en okulär granskning av figur 3.8 hade varit tillräcklig för att konstatera att en avsevärd förbättring av svarstiderna skett.



Figur 3.8 – Svarstidstester för Light 1999 och 2003

3.1.4 Test E, pålitlighet

E-testerna syftar dels till att ge underlag för ett signifikanstest och dels för att ge underlag för att beräkna ett värde för POFOD vid extern respektive intern användning (se 2.1.2). Trots det relativt höga antalet tester, 55 000 interna och 45 050 externa, verkar inte detta vara tillräckligt för att ge ett nyanserat och tillförlitligt POFOD-värde. Anledningen till detta är att endast ett fel registrerades.

Detta enda fel uppstod vid ett externt test (se tabell 3.3). En beräkning av POFOD för extern drift skulle med detta utfall bli $1/45\ 050 = 0,000022$. Det sanna värdet skiljer sig med stor sannolikhet från det uppmätta, men det verkar baserat på det stora antalet tester vara osannolikt att det skulle vara avsevärt högre. Däremot skulle det faktiska POFOD-värdet kunna visa sig vara betydligt lägre än det nu uppmätta om antalet körningar mångdubblats. För intern drift blir värdet 0 eftersom inga fel registrerades. Inte heller detta säger något annat än att det sanna värdet för POFOD är lågt.

	Externt	Internt	Totalt
Fel uppstod	1	0	1
Fel uppstod ej	45 049	55 000	100 049
Totalt	45 050	55 000	100 050

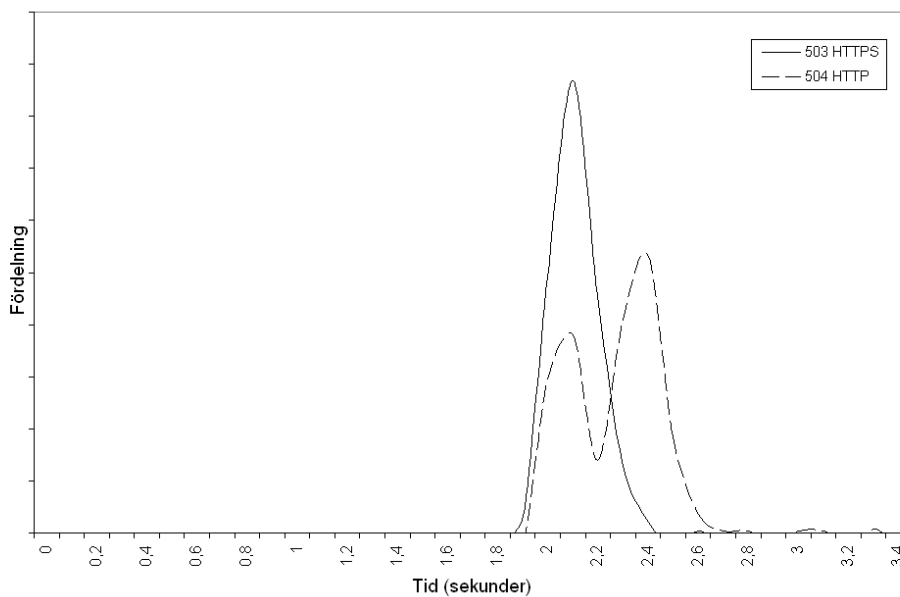
Tabell 3.3 – Utfall av test E

Det faktum att felfrekvensen är högre för externa tester än interna kan mycket väl enbart bero på slumpen. För att testa detta genomförs ett signifikanstest (se 2.1.2).

Ett χ^2 -test med 95-procentig konfidensnivå gav följande resultat. Erhållen χ^2 blev 1,222. Detta ska jämföras med $\chi^2_{0,05} = 3,841$. Då $1,222 < 3,841$ kan det konstateras att skillnaden ej är statistiskt signifikant.

3.1.5 Krypterat/okrypterat

Testerna 503 och 504 är exakt samma tester med den skillnaden att test 503 genomfördes med krypterad kommunikation mellan klient och webbserver. Även testet med krypterad kommunikation är alltså utfört internt. Syftet är att avgöra om och i så fall hur mycket kryptering påverkar prestanda.



Figur 3.9 – Jämförelse mellan krypterad och okrypterad trafik

Resultaten visar något förvånande att testet med okrypterad kommunikation både hade sämre medeltider och högre standardavvikelse. När spridningsdiagrammen för respektive test studeras ser man dock att någonting, gissningsvis en process på någon av servrarna, påverkat andra halvan av den okrypterade testkörningen (504). Denna påverkan verkar inte ha funnits vid test 503 vilket förklarar varför både medelvärdet och spridningen är lägre för detta test. I grafen ovan har 504 två toppar istället för den förväntade enda. Den andra av dessa är resultatet av ovan nämnda påverkan. Om man bortser från denna och endast ser till första halvan av testet, d.v.s. den del av testet som är jämförbar med test

503, tycks medelvärden och även spridningen vara någon lägre än för testet med krypterad kommunikation. Skillnaden är dock mycket liten vilket fortfarande kan te sig förvånande. Detta stämmer dock väl överens med studier som gjorts avseende prestandaskillnader mellan krypterad (https, ssl) och okrypterad (http) trafik.

“We find that secure Web servers perform well in comparison to non-secure servers. In particular, measurements show that on typical PCs encrypted Web communications using SSL and RC4 can transfer data at speeds similar to non-encrypted HTTP.” [Goldberg 98]

3.1.6 Modemtester

Ett mycket litet antal externa tester utfördes med modem. Resultatet var i princip givet från början. Modem är inte praktiskt användbar för en kund som köper en programuthyrningstjänst. Tiderna är mellan 5 och 10 gånger längre än motsvarande tester ovan. Lägsta medel för svarstidstester blev drygt 100 sekunder, d.v.s. en svarstid på över 20 sekunder per operation och medeltid för uppstart av Light blev 24,67 sekunder. Utfallet ligger långt över de gränsvärden på 4 respektive 10 sekunder som satts upp.

3.2 Enkätundersökning

Enkätundersökningens frågor och resultat presenteras i sin helhet i *Bilaga B – Resultat av enkätundersökning*. För att underlätta läsningen och ge möjlighet till jämförelser mellan svar som inkommit från interna respektive externa användare visas för varje fråga dessa grupper i separata cirkeldiagram. Vid sidan av dessa grafiska sammanställningar finns även en tabell som presenterar svaren för varje enskild enkät.

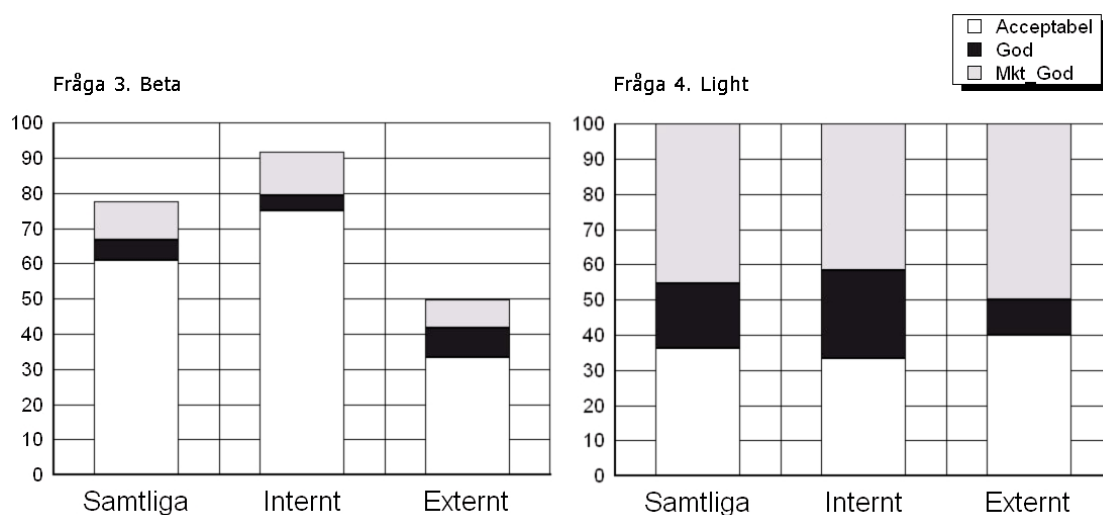
Enkäten gav 36 svar vilket innebär en svarsfrekvens på totalt 78%. Av dessa använde 33,3% i huvudsak systemen krypterat över Internet och 66,7% okrypterat internt. Enkäten är för liten för att applicera statistiska metoder på resultatet. *“Enkäter med färre än 40 tillfrågade är det knappt meningsfullt att bearbeta statistiskt på något sofistikerat sätt.”* [Ejvegård 96]. Trots att urvalet omfattar få personer och att svarsfrekvensen är låg på vissa enskilda frågor redovisas resultaten alltid i procentsatser. Detta därför att en relativ siffra lämpar sig bättre för direkta jämförelser mellan frågor och delpopulationer inom urvalsgruppen. Exempelvis är antalet som använder systemen internt större än antalet som använder dessa externa. Därför skulle en jämförelse mellan interna och externa användares svar baserat på absoluta tal bli missvisande eller åtminstone svårtolkat. För de frågor där svarsalternativet *svarar ej* är mycket högt är anledningen antingen att frågan svår att besvara utan omfattande kunskaper i ämnet (ex. frågor rörande systemens säkerhet) eller att frågan inte riktar sig till samtliga användare (ex. frågor som endast berör de som använt systemen både internt och externt). För att svar mellan olika frågor ska kunna jämföras kommer svar med svarsalternativet *svarar ej* konsekvent plockas bort

som internt bortfall (se 2.4) vid presentation av resultaten. Exempel: Där det nedan i resultatet anges att 15% har svarat mycket bra på en fråga betyder detta 15% av de som svarat på enkäten samt inte angivit svarsalternativet *svarar ej* på frågan. Där andelen som angivit svarsalternativet *svarar ej* är väldigt högt kommer detta anges.

Samtliga har använt Beta och över 60% har erfarenhet av att använda Light. Över hälften har erfarenhet av att köra ett eller båda systemen från olika geografiska platser. Hela 88,9% angav att det är viktigt eller mycket viktigt att kunna fylla i sin tidrapport var som helst d.v.s. att systemen är geografiskt oberoende.

3.2.1 Prestanda

På fråga 3 om hur prestanda upplevs för Beta svarade 91,7% av de som i huvudsak använder systemen internt att den var *acceptabel* eller bättre. Det var dock endast 16,7% som ansåg att prestanda var *god* eller *mycket god*. 8,4% gav betyget *dålig* eller *mycket dålig*. Av de som i huvudsak använder systemen externt svarade 49,9% att prestanda var *acceptabel* eller bättre på motsvarande fråga. 25% ansåg att prestanda var *dålig* och ytterligare 25% ansåg att den var *mycket dålig*. På motsvarande fråga för Light uppgav 100% av både interna och externa användare att prestanda upplevdes som *acceptabel* eller bättre. Hela 45,5% uppgav att prestanda upplevdes som *mycket god*.



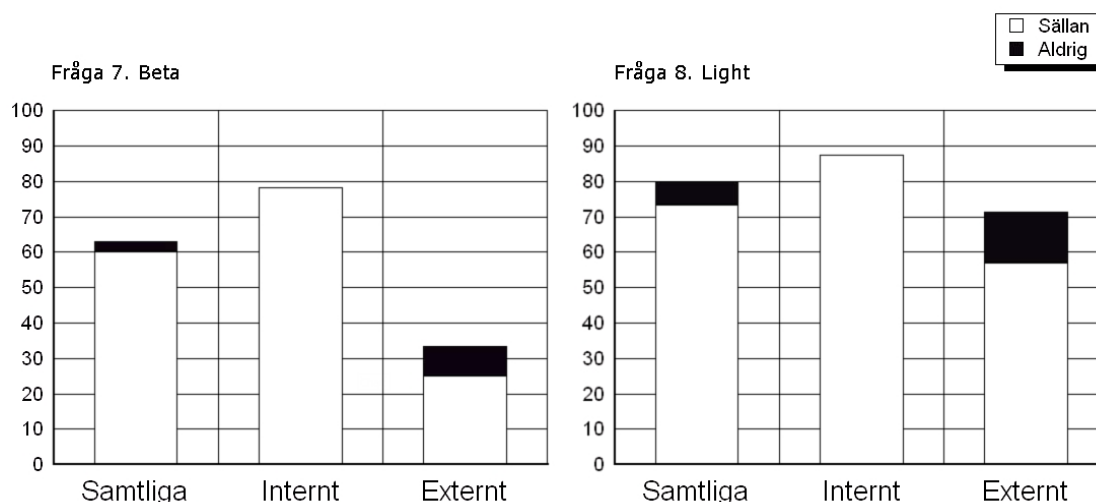
Figur 3.10 – Andel som svarat *acceptabel*, *god* eller *mycket god* på frågan "Hur upplever du prestanda för tidrapporteringen?"

På fråga 5 om hur användaren upplever variationer i prestanda för Beta uppgav 22,7% av de interna användarna att variationerna var *stora* eller *tydliga* medan motsvarande siffra

för de externa användarna var 40%. Ingen angav svarsalternativet *extrema*. På motsvarande fråga för Light angav 0% av både interna och externa användare att variationerna var *extrema* eller *stora*. 11,8% uppgav att variationerna var *tydliga*. För Light uppgav 35,3% och för Beta 9,4% av både interna och externa användare att variationerna upplevdes som *ej märkbara*. Av de som hade erfarenhet att använda Beta både internt och externt svarade 40% att prestanda var *mycket sämre* vid externa användning jämför med intern, 46,7% att prestanda var *sämre* och 13,3% att den var *lika bra*. Motsvarande siffror för Light var 12,5%, 62,5% och 25%.

3.2.2 Pålitlighet

På fråga 7 om hur ofta användaren upplever att systemet ej svarar vid användning av Beta angav 21,7% av de interna användarna svarsalternativen *ofta* eller *ibland*. Motsvarande siffra för de externa användarna var 66,7%. På fråga 8 om hur ofta användarna upplever att Light ej svarar vid användning uppgav ingen av användarna alternativen *mycket ofta* eller *ofta*. 20% av de interna och 26,6% de externa användarna uppgav *ibland*.



Figur 3.11 – Andel som svarat *sällan* eller *aldrig* på frågan "Hur ofta upplever du att systemet inte svarar vid användning?"

På fråga 9 om hur ofta Beta genererar felmeddelanden vid användning svarade 0% av de interna användarna *mycket ofta* eller *ofta*. Motsvarande siffra för de externa användarna var 16,6%. 17,4% av de interna användarna och 0% av de externa användarna valde svarsalternativet *aldrig* på denna fråga. På samma fråga fast för Light uppgav ingen att fel med felmeddelande uppstår *mycket ofta* eller *ofta*.

På fråga 11 om hur ofta fel uppstår vid användning av Beta utan att felmeddelande genereras var det ingen som angav *mycket ofta* eller *ofta*, men ingen av de interna och 12,5% av de externa användarna uppgav svarsalternativet *ibland*. På fråga 16 om hur pålitligheten upplevs för Beta och Light tillsammans jämför med det tidigare Excel-baserade systemet svarade ingen av de interna, men 50% av de externa användarna att pålitligheten upplevs som *sämre* för de nya systemen. 64,7% av de interna användarna uppgav att pålitligheten upplevdes som *bättre* eller *mycket bättre* för de nya systemen medan motsvarande siffra för de externa användarna var 12,5%.

3.2.3 Säkerhet

De frågor i enkäten som berör säkerhetsaspekter på intern respektive extern drift av systemen visar svaren ett mycket stort internt bortfall, över 80%. Av de få som svarat upplever ungefär hälften säkerheten som *sämre* vid extern drift och ungefär hälften som lika bra, men antalet som svarar är som sagt alltför lågt för att man ska kunna tillmäta dem någon vikt.

3.2.4 Systemutformning

Resultaten med avseende på skillnaden mellan de två systemen framgår till viss del i redovisningen under rubrikerna *Prestanda* och *Pålitlighet* ovan. Det finns ofta tydliga skillnader i svaren för Beta och Light. Vad gäller upplevelsen av prestanda var det ingen av de interna användarna som gav Light underkänt betyg i form av svarsalternativen *dålig* eller *mkt dålig*. 8,4% av samma användargrupp gav Beta dessa betyg. Av de externa användarna var det ingen som gav Light betygen *dålig* eller *mkt dålig*, men 50% gav Beta dessa betyg. Visserligen fanns alltså skillnader i bedömningen av prestanda för Beta och Light även bland interna användare, men det är endast bland externa användare man finner ett tydligt underkännande av prestanda för Beta.

3.3 Övriga källor

Nedan följer en kort redovisning av litteraturstudien avseende pålitlighet och säkerhet med anknytning till programuthyrning. Anledningen till att kvalitativt material samlats in för säkerhetsaspekten är att säkerhetsytesen evalueras baserat på detta material. Pålitlighetsytesen evalueras utifrån våra kvantitativa tester, men det skulle visa sig att dessa tester inte var tillräckliga för ett tillförlitligt resultat (se 4.6, *Hypotesernas beviskraft*). Därför finns här även ett kort stycke om pålitlighet.

3.3.1 Pålitlighet

Företaget Market Vision Research genomförde i början av år 2000 en undersökning som omfattade 800 medelstora företag angående deras anlitan av programuthyrningsföretag. När företagen ombads ange skäl för att anlita programuthyrningsföretag uppgav hela 67% "Increased system uptime". Detta var den 7:e vanligaste anledningen som angavs [Susarla 01]. Intressant är formuleringen av detta alternativ. Att 67% faktiskt uppger ökad pålitlighet. Inte bara *bra*, *acceptabel* eller *inte märkbart sämre*, utan *ökad*. Denna åsikt återkommer i ITAAs (Information Technology Association of America) årligen återkommande undersökning av ASP-kunder från 2003 där ökad tillgänglighet (availability) anges som en generell och mycket uppskattad fördel med programuthyrning. En av kunderna i undersökningen säger: "*We've had only two hours of unscheduled down time in three years. That's better than streetlights.*" [ITAA 03]. En kundundersökning utförd av tidningen Interactive Week bekräftar bilden. Ökad datasäkerhet och ökad stabilitet/tillgänglighet är bland de vanligaste anledningar som anges för att anlita programuthyrningsföretag [Wetzel 01].

Ofta har programuthyrningsföretagen flera uppkopplingsmöjligheter mot Internet för att inte riskera tillgängligheten [Counts 02].

"Customers who turn to ASPs often find better application reliability and availability than they experienced from their internal IT organisations, because the ASP has the staff, expertise and state-of-the-art equipment to maintain and sustain the most reliable and up-to-date service." [Halpin 03]

3.3.2 Säkerhet

Konsultföretaget Frost & Sullivan genomförde 2001 en undersökning angående inställningen till programuthyrning bland 150 företag med maximalt 250 anställda. Företag i denna storleken har av Frost & Sullivan identifierats som den primära målgruppen vid programuthyrning. En av de huvudsakliga slutsatserna av undersökningen var att oron över tekniska frågor gällande prestanda och säkerhet är fortsatt stor. "*Many companies are still apprehensive about using ASPs because of security, performance and support concerns. But the report indicated that most of those were based on technical worries*" [Eisner 01]. Säkerheten utgör en arkilleshäl för hela konceptet programuthyrning [Larsson 01].

"We found that when it came to network security, most ASP's had a better system in place than we had in our existing system. It was clear that having a secure system for protecting the integrity of all our records is a top priority for the ASP industry as a whole." [Counts 02]

Det är tydligt att säkerhetsaspekten är en av de frågor som programuthyrarna arbetat mycket med och här ligger man långt framme. Den stora skillnaden är att programuthyrarna i allmänhet är mer eller mindre experter inom säkerhet medan det typiska lilla till medelstora företaget inte på samma sätt har möjlighet att följa med på detta område som förändras mycket snabbt. En flora av säkerhetssystem/tjänster erbjuds av programuthyrare. Alla programuthyrare värda namnet erbjuder standardkryptering, uppdaterade antivirusprogram och brandvägg som ett minimum av säkerhet [Schneider 01]. De flesta verkar dock gå betydligt längre än så. Några av de vanligare åtgärder som erbjuds för ökad säkerhet är: kraftfull kryptering, system för användaridentifiering och inloggning, VPN (virtual private network), privata zoner av nätverket dedicerade för enskilda kunder, redundant hårdvara, backup-rutiner, dubbla brandväggar av olika fabrikat, övervakning av intrångsförsök, bevakning av säkerhetsvarningar, installation av säkerhetsuppgrederingar, brandklassade och inbrottskyddade serverrum, reservsystem för elkraft, sekretessavtal och kontroll av anställda [Schneider 01, Larsson 01, Wainwright 98].

I ITAAs årligt återkommande undersökning av ASP-kunder 2003 blev slutsatsen att den allra främsta fördelen med programuthyrning är just leverantörernas kompetens och säkerhetssystem. Särskilt när det gäller oförutsägbara händelser som elavbrott, stöld, brand och hårdvaruproblem. Samtliga företag ansåg enligt undersökningen att riskerna var signifikant lägre vid programuthyrning än om de hade driftat systemen själva [ITAA 03]. I föregående års undersökning angav kunderna ökad säkerhet som en av de 10 viktigaste anledningarna till att gå över till att hyra program. Drygt 80% angav att målet de satt upp gällande ökad säkerhet också hade uppnåtts [ITAA 02].

Det är betydligt enklare att finna personer som lyfter fram de positiva sidorna med programuthyrning än tvärtom. Dock anges i nummer 832 av InternetWeek från 2000 att programuthyrarna fortfarande saknar standards och verktyg för att kunna säkra prestanda, pålitlighet, systemsamverkan och säkerhet. Samtidigt hävdas även här att dessa problem kommer att lösas med tiden. [Boyd 00].

Vad gäller krypterad datatrafik och SSL så verkar det som att de krypteringsmetoder som är aktuella inom ramen för denna undersökning erbjuder en hög nivå av skydd av informationen. De externa testerna som utfördes 1999 använde sig av 40-bitars kryptering. Numera anses denna kryptering vara *"breakable with moderate effort"* [Goldberg 98]. Då testerna utfördes var denna krypteringsnivå dock standard. Enligt Alex Frausin, IT-Manager på Fraunhofer-Chalmers Research Center for Industrial Mathematics och expert på datasäkerhet, så ansågs denna kryptering då erbjuda en tillräcklig säkerhet för applikationer som inte ställer väldigt höga krav, som t.ex. banktransaktioner. Det var fullt möjligt redan 1999 att knäcka en 40-bitars kryptering, men insatsen som krävdes ansågs så stor att denna kryptering skulle ge tillräckligt skydd för de flesta tillämpningsområden. *"Of course, a cryptosystem need not be utterly unbreakable to be useful. Rather, it needs to be strong enough to resist attacks by likely enemies for whatever length of time the data it protects is expected to remain valid."* [Garrett]

Numera är 128-bitars kryptering standard. Detta var inte möjligt 1999 då amerikanska exportrestriktioner hindrade högre grad av kryptering utan speciella undantagstillstånd. 128-bitars kryptering är 309 485 000 000 000 000 000 000 gånger kraftfullare än 40-bitars [RSA 03] och anses vara *"enough to be unbreakable by known methods in typical large computer facilities"* [Goldberg 98].

4 Diskussion

Frågan som arbetet söker besvara lyder: ”Är tekniska förutsättningar, vad gäller prestanda, pålitlighet och säkerhet, goda för att bedriva programuthyrning av medelkomplexa webbaserade system mot små och medelstora företag och har dessa förutsättningar signifikant förbättrats under de senaste åren?”

Denna fråga föranledde formulering av ett antal hypoteser som ansågs vara till god hjälp för att besvara den. Av de formulerade hypoteserna måste prestandahypotesen, pålitlighetshypotesen, säkerhetshypotesen samt tidshypotesen styrkas för att frågan ska kunna besvaras jakande. Resonemanget bakom detta är att en kedja inte är starkare än sin svagaste länk. Om en enda av hypoteserna falsifieras kan inte frågan besvaras jakande förutsatt att hypotesen är relevant i förhållande till frågeställningen. Om man exempelvis skulle komma fram till att prestanda inte är tillräckligt god så följer att de tekniska förutsättningarna inte är goda för programuthyrning inom avgränsningens ramar. Med systemhypotesen förhåller det sig dock litet annorlunda. Även om denna hypotes falsifieras kan man mycket väl tänka sig ett jakande svar på frågeställningen. Om man t.ex. skulle komma fram till att alla system som fungerar bra internt även gör det externt skulle denna hypotes falsifieras, men detta betyder inte att de tekniska förutsättningarna för programuthyrning är dåliga, snare tvärt om.

Även om samtliga fyra relevanta hypoteser styrks kan det mycket väl vara så att svaret på frågan egentligen är nej. Andra metoder hade kunnat användas för att testa hypoteserna så som de är formulerade. Det finns ingen garanti för att resultatet av hypotesprövningen i så fall blivit detsamma.

Har hypoteserna korroborerats? Den uppmärksamme läsaren har redan konstaterat att så är fallet. Med undantag av säkerhetshypotesen kan samtliga evalueras rent matematiskt enligt de regler som satts upp för hypotesprövningen. Vid sidan av de undersökningar som gjorts i syfte att evaluera hypoteserna kan resultatet av enkätundersökningen användas som en avstämning av användarnas subjektiva upplevelser av systemen med avseende på undersökta kvalitetskriterier. Sammanfaller resultaten från enkäten med resultaten av hypotesprövningarna ges ytterligare tyngd åt dessa.

Låt oss resonera kring de redovisade resultaten i förhållande till hypoteserna.

4.1 Prestanda

Prestandahypotesen styrks därför att medelvärde plus standardavvikelse multiplicerat med standard normal deviate för extern uppstartstest för Light blir $4,35 + (0,39 \cdot 1,645) = 4,99$ sekunder (vilket klart understiger gränsvärdet 10 sekunder) samtidigt som medel-

värde plus standardavvikelse multiplicerat med standard normal deviate för extern svarstidstest Light blir $17,20 + (0,80 \cdot 1,645) = 18,52$ (vilket understiger gränsvärdet 4 sekunder $\cdot 5 = 20$ sekunder).

Av siffrorna ovan framgår att uppstartstiderna med god marginal klarade uppsatt gränsvärde. Svarstiderna låg också på rätt sida gränsvärdet, men här var marginalerna betydligt mindre. Betyder detta att det testade systemet har väldigt god prestanda gällande uppstart, men tveksamma svarstider vid vanliga operationer. Inte nödvändigtvis. Det kan lika väl vara så att skillnaderna visar på svårigheten med att sätta upp realistiska gränsvärden för mätningar av vad som är ”acceptabelt”.

Ovan menar vi att prestanda är acceptabel vid extern drift. Testresultaten visar dock att prestanda är betydligt bättre vid intern drift än extern. Faktum är att tiderna är ungefär de dubbla vid extern drift. Även om hypotesen har korroborerats enligt uppsatta regler är denna skillnad något som talar emot webbaserad programuthyrning. Man kan fråga sig varför man ska acceptera en prestanda som är avsevärt lägre än vid intern drift. Dock har dessa skillnader minskat mellan 1999 och 2003. Mer om detta senare.

Användarnas subjektiva åsikter om systemets prestanda som delges i resultatet av enkätundersökningen visar utan tvekan att prestanda för Light är acceptabel både i intern såväl som i extern drift. Samtliga som lämnade ett svar på frågan angav att prestanda för systemet upplevs som minst acceptabel. Hela 50% av de externa användarna svarade mycket god. Faktum är att trots att uppmätt prestanda är bättre internt än externt var det ändå en större andel av de externa användarna som gav systemets prestanda högsta betyg. Förklaringen ligger i det som alltid betonas som orsaken till att det är svårt att objektivt mäta prestanda, nämligen att upplevelsen av systems prestanda i högsta grad är subjektiv och baseras på förväntningar och tidigare erfarenheter. Medan de externa användarna tidigare endast hade erfarenhet av ett långsamt system och därför upplevde förbättringen som avsevärd hade de interna redan innan Light implementerades haft ett system som upplevdes acceptabelt gällande prestanda. Visserligen byggde den tidigare erfarenheten på samma system, Beta, men detta uppvisade, som prestandamätningarna visar, betydligt längre svarstider för externa användare än interna.

4.2 Pålitlighet

Pålitlighetshypotesen styrks eftersom skillnaden mellan interna och externa tester inte är signifikant. Dessutom är POFOD (Probability of failure on demand) 0,000022 för externa tester, vilket understiger gränsvärdet 0,0001.

Faktum är att detta mycket låga värde för POFOD förvånade oss. Endast ett fel på över 45 000 tester måste anses som ett bra resultat, även om det inte kan uteslutas att en mer realistisk utformning av testerna kan ge ett annat utfall. Ingenting i våra kvantitativa undersökningar har dock pekat på att pålitligheten skulle innebära något större problem för

webbaserad programuthyrning. Detta styrks ytterligare av undersökningar gjorda på kunder av programuthyrning (se 3.3.1) där man talar om i vilken utsträckning programuthyrning innebär en förbättring av pålitligheten snarare än om den är sämre eller lika god som vid intern drift.

Tittar man på enkätundersökningen får man en något annorlunda bild av pålitlighetsaspekten. Varken externa eller interna användare säger sig uppleva pålitlighetsproblem ofta eller mycket ofta. Däremot är det fler externa användare än interna som upplever problem med tillgänglighet eller systemfel ”ibland”. Även om svarsfrekvensen för dessa frågor är låg så är tendensen ganska tydlig och den ligger inte i linje med vad som framkommit i tester och andra undersökningar. Detta underminerar i någon mån denna hypotes i förhållande till uppsatsens frågeställning.

4.3 Säkerhet

Säkerhetsytesen styrks därför att de källor vi tagit del av nästan entydigt menar att företag som driver webbaserad programuthyrning i regel har högre kompetens och ett aktivare säkerhetsarbete än den genomsnittliga kund som faller inom de ramar som studien satt upp. Kunder av denna storlek har oftast inte resurser nog att följa med i den snabba utvecklingen på säkerhetsområdet. Viktigast av allt är dock att de standardmetoder som idag finns för kryptering ger ett mycket gott skydd för trafik över Internet. Krypteringsmetoderna har tydligt förbättrats över tiden främst på grund av att USA avskaffat tidigare exportrestriktioner av avancerade krypteringsmetoder. När befintliga kunder av webbaserad programuthyrning i flera undersökningar menar att säkerhetsaspekten är en av de viktigaste fördelarna med att välja att programuthyrning måste detta väga tungt.

Vad gäller de säkerhetsrelaterade frågorna i enkätundersökningen gav dessa för låg svarsfrekvens för att ha någon betydelse i sammanhanget.

4.4 Systemutformning

Systemutformningshypotesen styrks eftersom prestandahypotesen styrks för Light samtidigt som Beta inte klarade de uppsatta gränsvärdena vid extern drift, men godkändes vid intern (se figur 3.4 samt 3.7).

Testerna visar tydligt att prestanda är acceptabel för Light internt och externt, samt för Beta internt, men att Beta externt lider av kraftiga prestandaproblem och stor variation i uppstarts- och svarstider. Beta vid extern drift är inte i närheten av att klara de uppsatta gränserna och spridningen är enorm jämfört med övriga tester (se figur 3.3 och 3.6). Den relativa skillnaden mellan externa och interna medeltider är högre för Beta än för Light,

men riktigt tydligt blir det först när man tittar på variationer för svarstider. Light extern uppvisar ca 175% större variation (standardavvikelse) än Light internt medan motsvarande siffra för Beta är förbluffande 1575%. Lägga märke till hur fördelningskurvan för test 300 förmligen kollapsat (figur 3.6). Slutsatsen måste bli att utformningen av systemet är mycket viktig vid webbaserad programuthyrning.

Enkätundersökningen bekräftar delvis bilden. För Beta anser 8,4% av de interna användarna att prestanda är dålig eller mycket dålig. Motsvarande siffra för externa användare är 50%. För Light anger ingen användare, vare sig intern eller extern att prestanda är dålig eller mycket dålig. Nästan dubbelt så hög andel av de externa användarna jämfört med de interna uppgav att Beta uppvisar tydliga eller stora variationer i prestanda. Siffran är tydlig, men motsvarar inte fullt ut den mycket stora uppmätta variationen i prestanda (se ovan). Troligen spelar psykologiska faktorer in. Det sunda förnuftet och egen erfarenhet av upplevelsen av tid säger att människans förmåga att upptäcka variationer i tider är betydligt bättre för korta tider än för längre.

4.5 Tiden

Tidshypotesen styrks eftersom medelvärdet för tester utförda 2003 är signifikant lägre än medelvärdet för motsvarande tester 1999. Faktum är att skillnaderna var tillräckligt stora för att lätt kunna fastställas bara genom en snabb titt på insamlad testdata. Att en förbättring skett är föga förvånande, däremot var denna större än förväntat. Både uppstart och svarstider var mer än 30% lägre vid externa tester 2003 än motsvarande 1999. För de interna testerna var tiderna runt 15-20% lägre 2003. Möjliga förklaringar till prestandaförbättringen är snabbare internt nätverk hos leverantören, högre bandbredd till och från Internet, snabbare dataöverföring över Internet, snabbare hårdvara på servrar samt hos klienten, effektivare mjukvara (operativsystem, databas och webbläsare). Black-box metoden gör det omöjligt att avgöra i vilken grad varje enskild faktor påverkar resultatet. Dock kan vi ändå dra vissa slutsatser eftersom tester genomförts med olika delar av infrastrukturen inblandad. Eftersom en förbättring av prestanda har skett vid interna tester måste faktorer som inte innefattar kommunikation över Internet ha förändrats. Gissningsvis utgör förbättring av hårdvara en stor del av denna prestandaskillnad. Dessutom har prestanda ökat mer både i reella och relativa tal för externa tester. Alltså har även kommunikationen över Internet och/eller de inblandade parternas bandbredd mot Internet påverkat resultatet. Det är känt att leverantörens bandbredd tidigare var lägre och detta utgör med stor sannolikhet en viktig förklaring till förbättringarna.

4.6 Hypotesernas beviskraft

Frågeställningen ska besvaras med hjälp av de evaluerade hypoteserna. Möjligheten att besvara frågeställningen avgörs av hur mycket vikt man kan lägga vid hypoteserna och

för att avgöra det bör man fundera över hypotesernas *beviskraft*. Låt oss låna några principer och termer från argumentationsanalysen [Björnsson 94]. Argumentationsanalys är en metod att beskriva och värdera styrkan i en argumentation. Man talar här om en tes och ett antal argument som syftar till att stödja eller motsäga denna. Av avgörande betydelse för tesens styrka är argumentens beviskraft i förhållande till tesen. Varje arguments beviskraft bedöms i sin tur genom att man väger samman dess hållbarhet och relevans. Med hållbarhet menas sannolikheten att argumentet är sant. Med relevans menas styrkan i sambandet mellan tes och argument.

Exempel:

Om man har *Katten heter Måns* som tes och argumentet för detta är *Alla katter heter Måns* så har argumentet en mycket hög relevans, men vår erfarenhet säger oss att argumentet är falskt och därför saknar hållbarhet.

Man kan jämföra uppsatsens frågeställning med en tes och hypoteserna med argument. Har hypoteserna tillfredsställande grad av beviskraft? Vår bedömning är att hypotesernas hållbarhet kan ifrågasättas på några punkter:

- För pålitlighetshypotesen används måtenheten POFOD. Denna tar inte hänsyn till tidsfaktorn, d.v.s. den säger egentligen inget om sannolikheten för ett att ett system är tillgängligt under en viss tidsenhet, utan endast hur många fel som uppstår per anrop. Kanske hade det varit mer lämpligt att använda en metod som tar hänsyn till ett systems tillgänglighet över tiden istället för som nu endast felsannolikheten per anrop. Resultatet under 3.3.1 visar att det främst är tillgänglighet som efterfrågas.
- Det verkar som att stickprovets storlek inte var tillräcklig som underlag för ett tillförlitligt värde för POFOD.
- Enkätresultatet antyder att pålitligheten är sämre vid extern drift än vid intern. Detta motsäger utfallet av signifikanstestet som säger att pålitligheten inte är signifikant sämre. Detta är ytterligare ett tecken på att detta resultat inte är helt hållbart.
- Evalueringen av säkerhetshypotesen har inte baserats på egna kvantitativa data. Istället har denna skett utifrån andras, potentiellt vinklade, utsagor.
- För att insamlad prestandadata skulle vara användbar tvingades vi justera denna och bortse från vissa delar av testresultatet då detta påverkats av backup-körningar. Detta är naturligtvis olyckligt och hade kunnat undvikas om testerna utförts på annat sätt.
- Val av gränsvärden både för prestandahypotesen och pålitlighetshypotesen har gjorts med ett visst mått av godtycklighet.

Vad gäller prestanda och justering av data så hade inte heller beräkningar på ojusterad data falsifierat hypotesen. Hållbarheten påverkas alltså inte särskilt mycket av detta. Däremot är valet av gränsvärden helt avgörande för hypotesprövningens utfall. Hållbarheten bedöms som medelhög.

Pålitlighetshypotesen lider av en del problem. Särskilt den första punkten ovan sätter vissa frågetecken för dess hållbarhet, som följaktligen lämnar en del övrigt att önska. Hållbarheten bedöms därför som låg.

Säkerhetshypotesen är svårbedömd, men det faktum att de källor som studerats är relativt samstämmiga gör att hållbarheten trots allt bedöms som hög.

Systemutformningshypotesen: Två system, som bägge fungerar väl vid intern drift, uppvisade stora skillnader vid extern drift med avseende på prestanda. Av detta följer en mycket hög hållbarhet.

Resultaten visar en tydlig prestandaförbättring från 1999 till 2003. Vi betvivlar inte riktigheten i detta.

Följande bör noteras angående hypotesernas relevans i relation till frågeställningen. Hypoteserna valdes för att vara så relevanta som möjligt, och vi anser fortfarande att de har en hög grad av relevans. Det finns dock ett undantag, och det är hypotesen som behandlar systemutformning. Frågeställningen har förändrats och snävats av något sedan denna hypotes formulerades. Den fick ändå vara kvar eftersom förhoppningen var att man skulle kunna dra en del intressanta slutsatser av den.

<i>Hypotes</i>	<i>Hållbarhet</i>	<i>Relevans</i>	<i>Beviskraft</i>
Prestanda	Medel	Hög	Medel
Pålitlighet	Låg	Hög	Låg
Säkerhet	Hög	Hög	Hög
Systemutformning	Mycket hög	Ingen	Ingen
Tid	Mycket hög	Hög	Hög

Tabell 4.1 – Hypotesernas beviskraft

Den sammantagna beviskraften för hypoteserna i förhållande till frågeställningen är acceptabel, men inte god nog för ett helt säkert svar.

4.7 Slutsats

Resultaten ovan tyder på att de tekniska förutsättningar som undersökts var goda redan 1999, och 2003 är de ännu bättre, men ändå har inte webbaserad programuthyrning tagit fart i den utsträckning som många förutsåg runt millenieskiftet. Tyder detta på att frågeställningen besvarats felaktigt? Även om vi inte utesluter den möjligheten så tror vi inte att det är så. Artiklar och rapporter som behandlar programuthyrningskonceptet pekar sällan ut tekniska brister utan visar snarare på andra orsaker. Inom i princip alla nya ”heta” branscher går upp till 80% av de tidiga aktörerna i konkurs [Allen 02]. IT-bubblan sprack i ett kritiskt skede för ASP-företagen vilket gjorde att luften gick ur många satsningar. Detta är naturligtvis inte unikt för programuthyrningsföretag utan har drabbat

stora delar av IT-branschen. ASP fick en oseriös stämpel efter att många tidiga spelare var orutinerade och bara ute efter att håva in pengar genom att utnyttja modeordet, menar Robert Brown, analytiker på Gartner [IDG 02]. Andra faktorer som ofta tas upp är: satsningar på fel typ av system, svårigheter att upprätta fungerande avtal mellan parterna och att kunder inte varit mogna för konceptet.

I arbetet med denna uppsats har det inte framkommit några resultat som gör att arbetets frågeställning måste besvaras med ett nej. Samtliga hypoteser med relevans till frågan har korroborerats vilket alltså tyder på att tekniska förutsättningar, vad gäller prestanda, pålitlighet och säkerhet, är goda för att bedriva programuthyrning av medelkomplexa webbaserade system mot små och medelstora företag och att dessa förutsättningar signifikant har förbättrats under de senaste åren.

4.8 Framtida forskning

Under genomförandet av detta arbete har vi naturligtvis stött på ett antal områden och aspekter på ämnet som lämpar sig för vidare forskning. Av dessa vill vi särskilt lyfta fram följande tre.

- En viktig fråga i samband med programuthyrning gäller systemens pålitlighet. En av våra hypoteser angriper detta, men som konstaterats i stycket om hypotesernas beviskraft har vi för denna hypotes inte kommit fram till ett hållbart resultat. Hypotesens relevans är dock hög och här finns utrymme att göra betydligt mer omfattande och kanske mer genomtänkta undersökningar.
- Evalueringen av systemutformningshypotesen säger oss *att* utformningen av system är mycket viktigt vid webbaserad programuthyrning. Däremot behandlas inte frågan om *hur* ett system bör utformas för att uppvisa acceptabla egenskaper i dessa sammanhang. Det vore väldigt intressant att ta del av arbeten som lägger fram rekommendationer och riktlinjer för detta.
- Detta arbetes slutsats är att de tekniska förutsättningarna är goda. Detta i kombination med att många artiklar och analyser framhåller ASP-branschens problem och "misslyckande" som en följd av andra orsaker än rent tekniska innebär att studier behövs som belyser t.ex. affärsmodeller, typ av erbjudanden och studier av hur den potentiella marknaden ser ut.

5 Referenser

5.1 Böcker

Andréasson, Sven-Arne & Carlsson, Christer (1996). *Datakommunikation för Informatik*. Göteborg: Institutionen för Datavetenskap, Chalmers Tekniska Högskola [Andréasson 96]

Benson, Calum et al (2002). *GNOME Human Interface Guidelines*. The GNOME Usability Project [Benson 02]

Björnsson, Gunnar et al (1994) *Argumentationsanalys*. Stockholm: Natur och kultur [Björnsson 94]

Easterby-Smith, Mark & Thorpe, Richard & Lowe, Andy (1991). *Management Research: An Introduction*. London: Sage Publications Ltd [Easterby-Smith 91]

Ejvegård, Rolf (1996). *Vetenskaplig metod*. Lund: Studentlitteratur [Ejvegård 96]

Hetzel, Bill (1998). *The Complete Guide to Software Testing*. Massachusetts: QED Information Sciences Inc [Hetzel 98]

Miller, Irwin (1995). *Statistical Methods for Quality*. USA: Prentice Hall [Miller 95]

Nielsen, Jakob (1994). *Usability Engineering*. San Fransisco: Morgan Kaufmann [Nielsen 94]

Nielsen, Jakob (2000). *Designing Web Usability: The Practice of Simplicity*. Indianapolis: New Riders Publishing [Nielsen 00]

Shneiderman, Ben (1997). *Designing the User Interface*. Addison-Wesley [Shneiderman 97]

Sommerville, Ian (1996). *Software Engineering*. USA: Addison-Wesley [Sommerville 96]

Wallmüller, Ernest (1994). *Software Quality Assurance: A practical approach*. Hertfordshire: Prentice Hall [Wallmüller 94]

5.2 Artiklar/Rapporter

- Allen, Doug (2002). *ASPs: After the Smoke Clears, Still a Lot of Smoke*. Network Magazine, March 2002 [Allen 02]
- ASP Outsourcing: The Customer Experience*. (2003). ITAA [ITAA 03]
- Bartlett, John et al (2001). *Understanding Web Performance*. Business Communications Review, October 2001 [Bartlett 01]
- Boone, Kevin. *How to conduct a survey*. www.kevinboone.com [Boone]
- Boyd, Jade (2000). *Technical Limitations Hold Back ASPs*. USA: InternetWeek #832 [Boyd 00]
- Butler, T. W. (1983) *Computer response time and user performance*. Boston: ACM Special Interest Group on Computer-Human Interaction [Butler 83]
- Caldwell, Bruce (1998). *Web host appeal*. Informationweek #714 [Caldwell 98]
- Counts, Mark et al (2002). *Outsourcing with an ASP* [Counts 02]
- Eisner, Adam (2001). *Outlook brightens for European ASPs*. Web Host Industry Review, October 2001 [Eisner 01]
- Fiasko för outsourcing öppnar för ASP*. (2002). www.idg.se [IDG 02]
- Garrett, Paul. <http://www.math.umn.edu/~garrett/crypto/faq.html> [Garret]
- Goldberg, Arthur et al (1998) *A Comparison of HTTP and HTTPS Performance*. New York University [Goldberg 98]
- Gunnarsson, Ronny (2002). *Forskningsmetodik - Kvantitativa (statistiska) och kvalitativa ansatser*. <http://infovoice.se/fou/> [Gunnarsson 02]
- Halpin, Eamus (2003). *Application Service Provision – ASP*. iFuel [Halpin 03]
- Harding, T et al (1999). *Requirements For Inter-operable Internet EDI*. The Internet Society [Harding 99]
- Hogan, Hank (2000). *ASPs Under Pressure to Explain Security Policies to Customers*. ASPstreet [Hogan 00]
- ITAA ASP Customer Satisfaction Survey*. (2002). ITAA [ITAA 02]

- Landgrave, Tim (2002). *ASP.NET, MVC design pattern*. Builder.com [Landgrave 02]
- Ledford Jerri (2000). *ASP – The future is near*. TVA News [Ledford 00]
- Loftus, Peter (2000). *IBM Web Exec Sees Internet Getting Faster, Easier To Use*. Wall Street Journal, April 2000 [Loftus 00]
- Larsson, Bernt (2000). *ASP – I framtiden hyr du programmen via webben*. Nätverk & Kommunikation [Larsson 01]
- Myers, Wayne (2000). *Belittling designers, two kinds of accessibility*. Wayne Myers-Education [Myers 00]
- Müller-Merbach, Heiner (1994). *A System of System Approaches*. Interfaces 24:4 July/August [Müller-Merbach 94]
- Niccolai, James (2000). *Future Web: Fast, Smart, and Private*. IDG News Service [Niccolai 00]
- Nielsen, Jakob (1998). *Nielsen's Law of Internet Bandwidth*. Jakob Nielsen's Alertbox, April 1998 [Nielsen 98]
- Oracle Corporation (1996). *The Edge Server*. Interactive Media Server Group [Oracle 96]
- Propson, DG (2001). *Who can you trust? USA: Small Business Computing & Communications* [Propson 01]
- RSA Security (2003). <http://www.rsasecurity.com/standards/ssl/basics.html> [RSA 03]
- Schneider, Robert (2001). *Security questions to ask an ASP*. Tech Target [Schneider 01]
- Sevcik, Peter J (1999). *Performance Issues Facing the World Wide Web*. BCR Volume 29 [Sevcik 99]
- Sevcik, Peter J (2001). *Internet Bandwidth: It's Time for Accountability*. BCR Volume 31 [Sevcik 01]
- Sevcik, Peter J (2003). *Web Performance – Not a Simple Number*. BCR Volume 33 [Sevcik 03]
- Shepherd, Clive (2002). *Hosting the E-Learning Party*. Fastrak Consulting Ltd [Shepherd 02]
- Susarla, Anjana (2001). *Myhts about Outsourcing to Application Service Providers*. IT Pro, May/June 2001 [Susarla 01]

Verzion Online. <http://biz.verizon.net/resources/glossary/> [Verzion 03]

Wainwright, Phil (1998). *Packaged Software Rental: The Net's Killer App*. London: Farleit Limited [Wainwright 98]

Wetzel, R (2001). *Faster, easier, cheaper*. USA: Interactive Week [Wetzel 01]

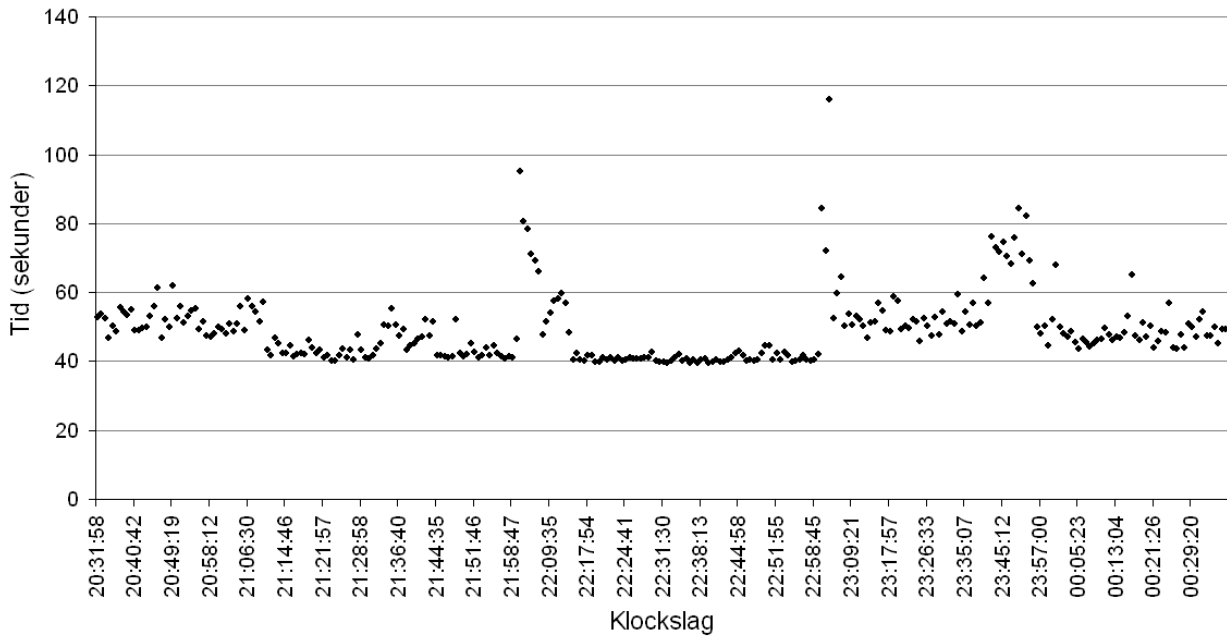
Whong, Jason (1997). *An Easy Way to Help Ambrosia*. [Whong 97]

Bilaga A

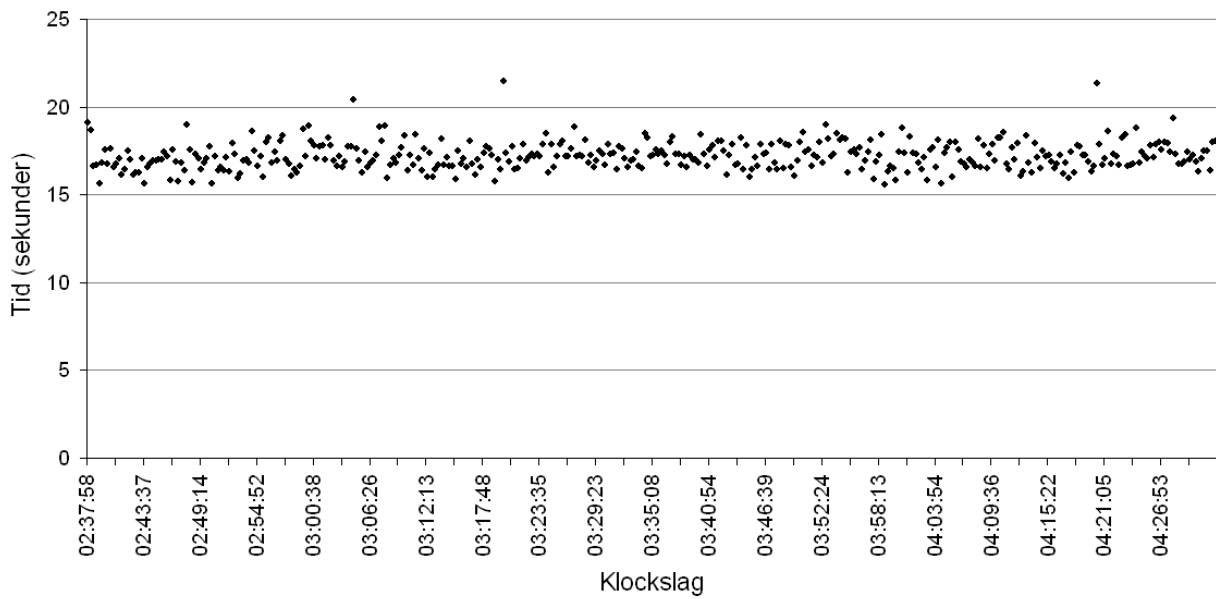
Resultat av mjukvarutester

Bilaga A – Resultat av mjukvarutester
Tekniska förutsättningar för webbaserad programuthyrning
Stefan Häggkvist, Andreas Östberg

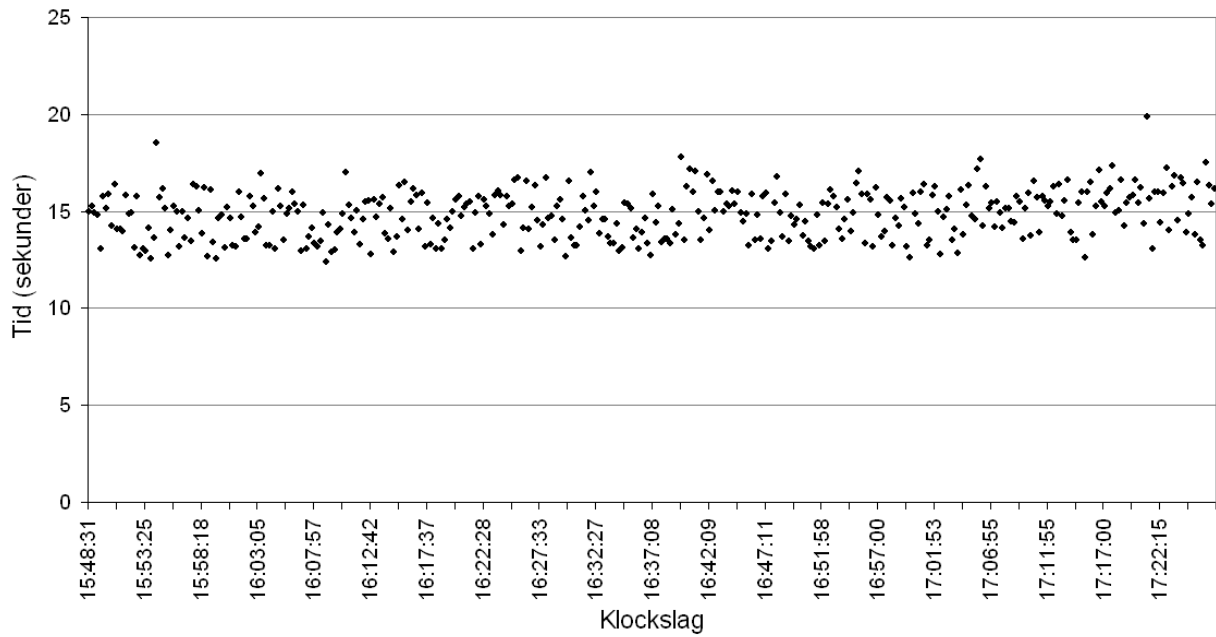
Test 300
Typ C. Svarstid Beta externt.



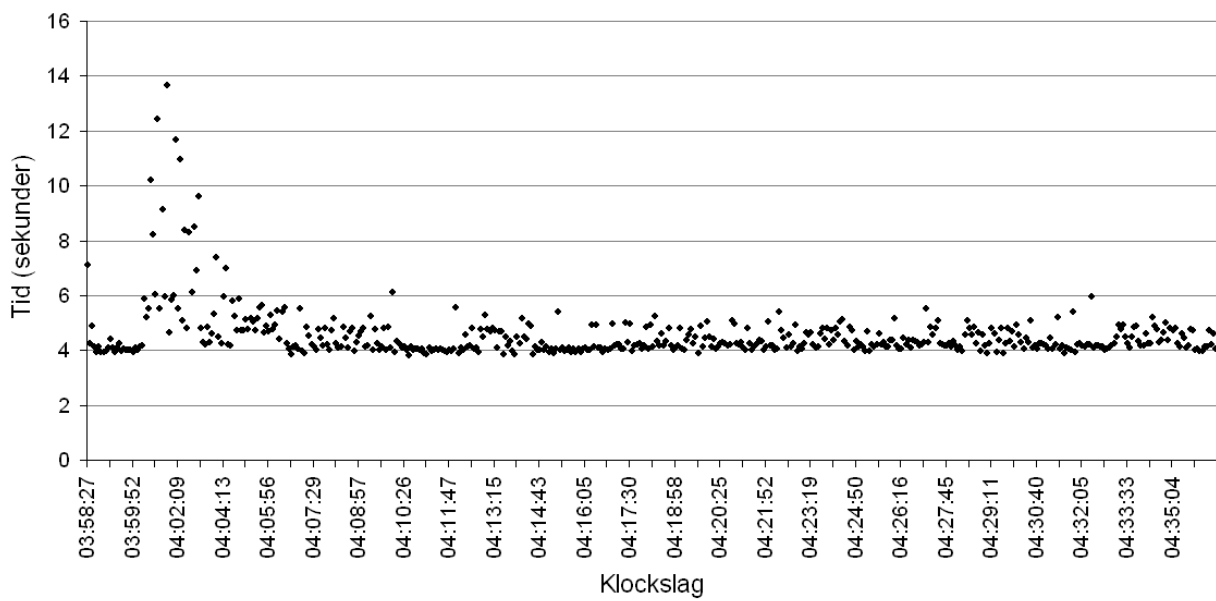
Test 305
Typ D. Svarstid Light externt.



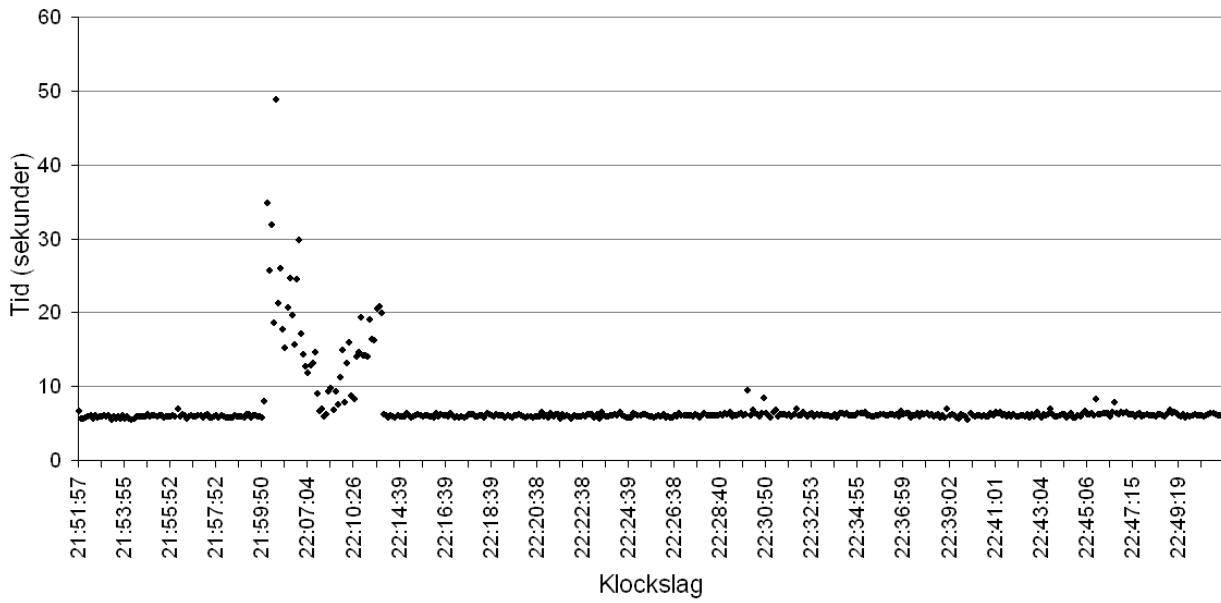
Test 307
Typ A. Uppstart Beta externt.



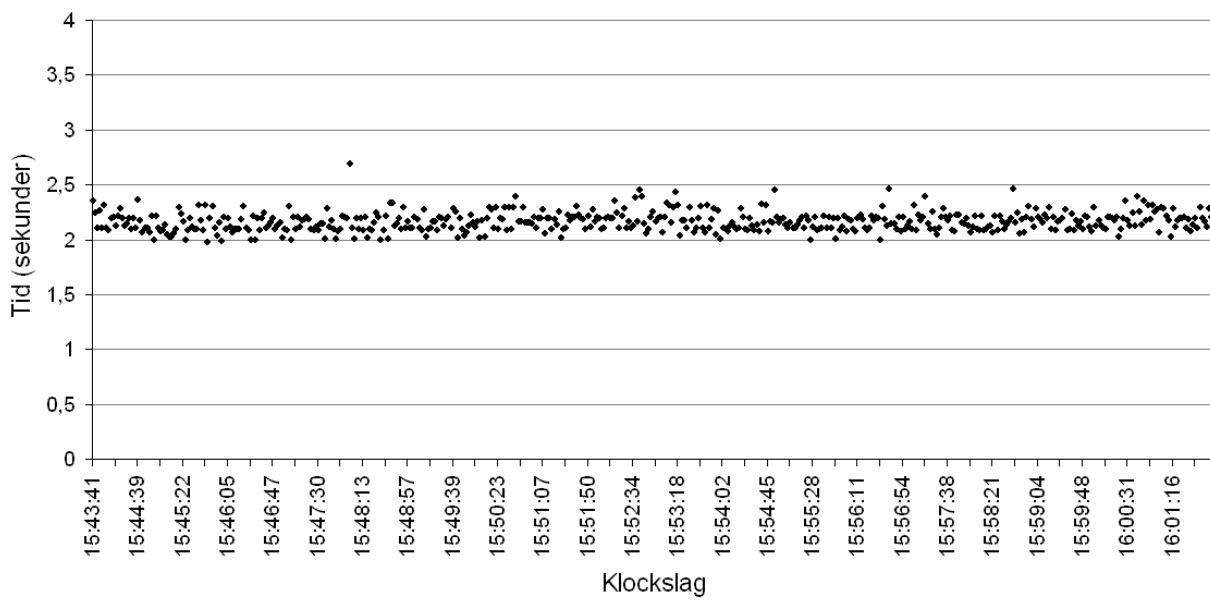
Test 308
Typ B. Uppstart Light externt.



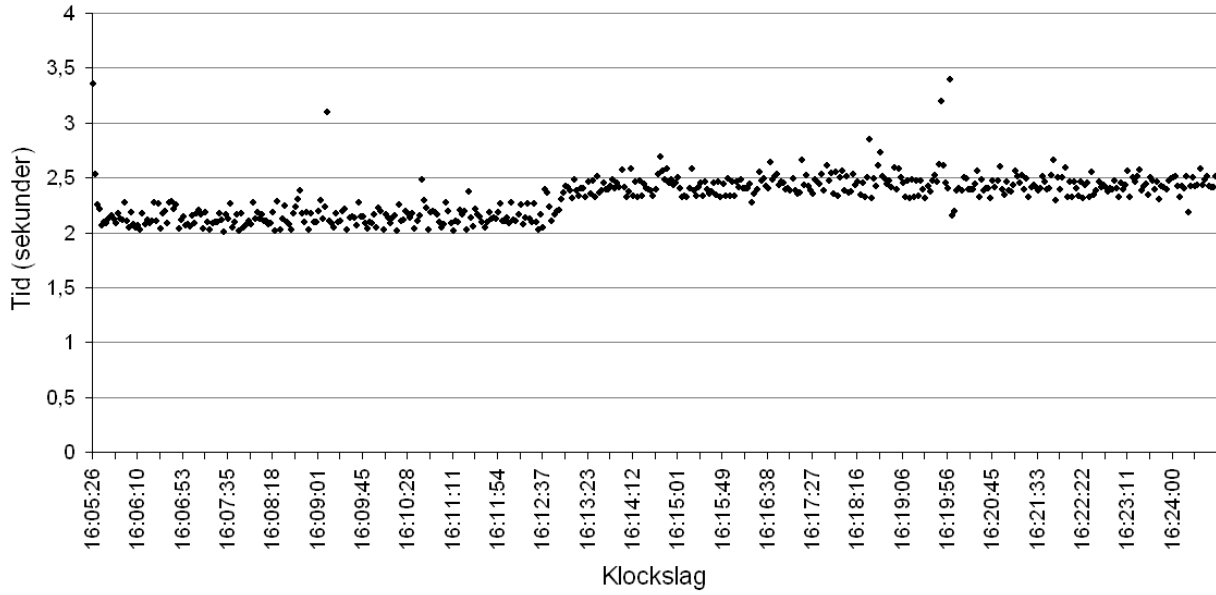
Test 501
Typ A. Uppstart Beta internt.



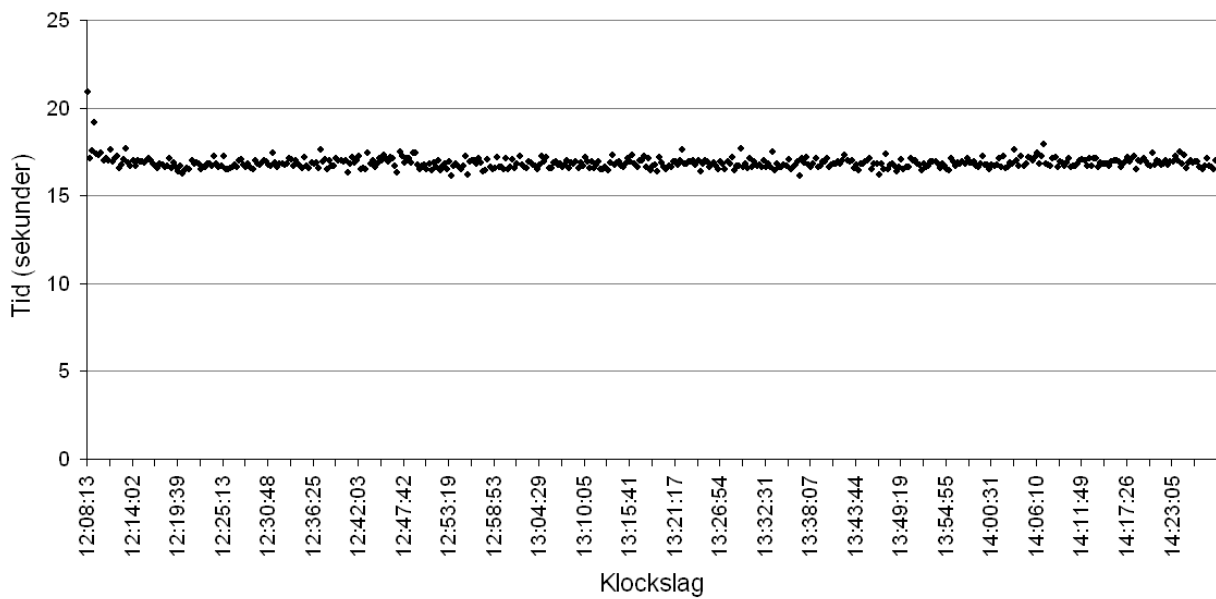
Test 503
Typ B. Uppstart Light internt (krypterat)



Test 504
Typ B. Uppstart Light internt

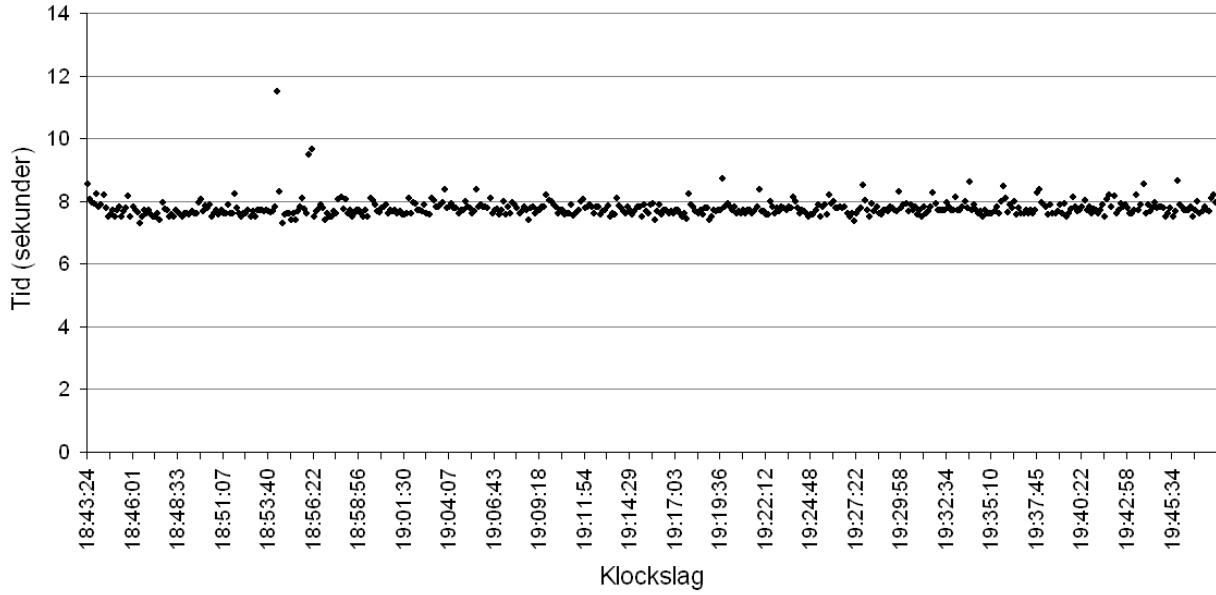


Test 517
Typ C. Svarstid Beta internt.

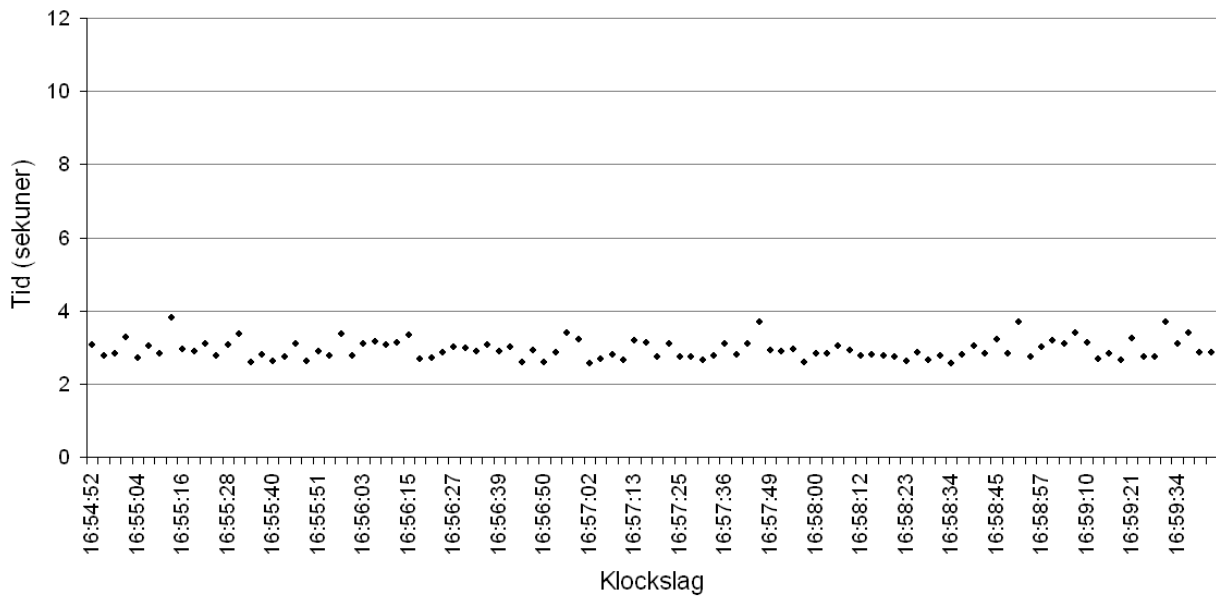


Bilaga A – Resultat av mjukvarutester
Tekniska förutsättningar för webbaserad programuthyrning
Stefan Häggkvist, Andreas Östberg

Test 520
Typ D. Svartstid Light internt.

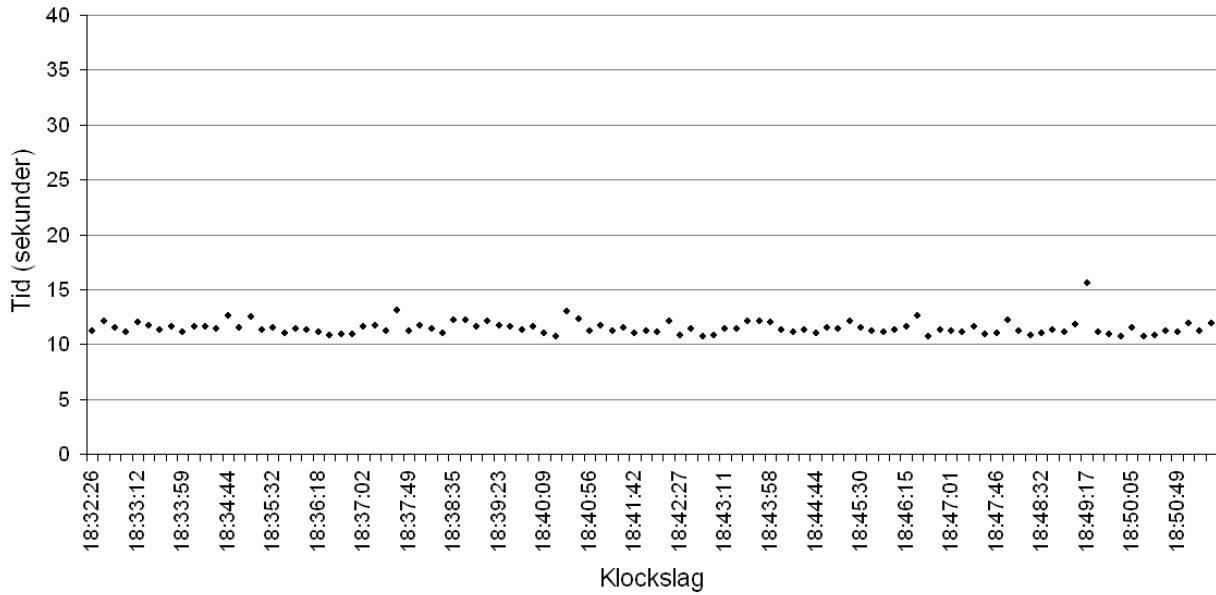


Test 20031
Typ B. Uppstart Light externt.

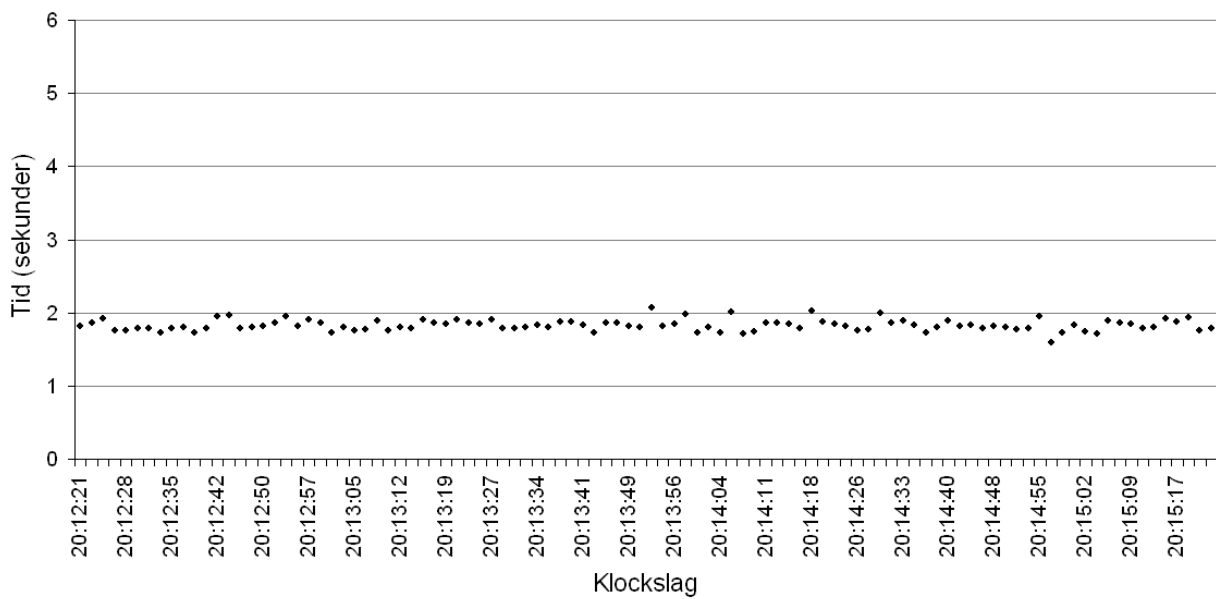


Bilaga A – Resultat av mjukvarutester
Tekniska förutsättningar för webbaserad programuthyrning
Stefan Häggkvist, Andreas Östberg

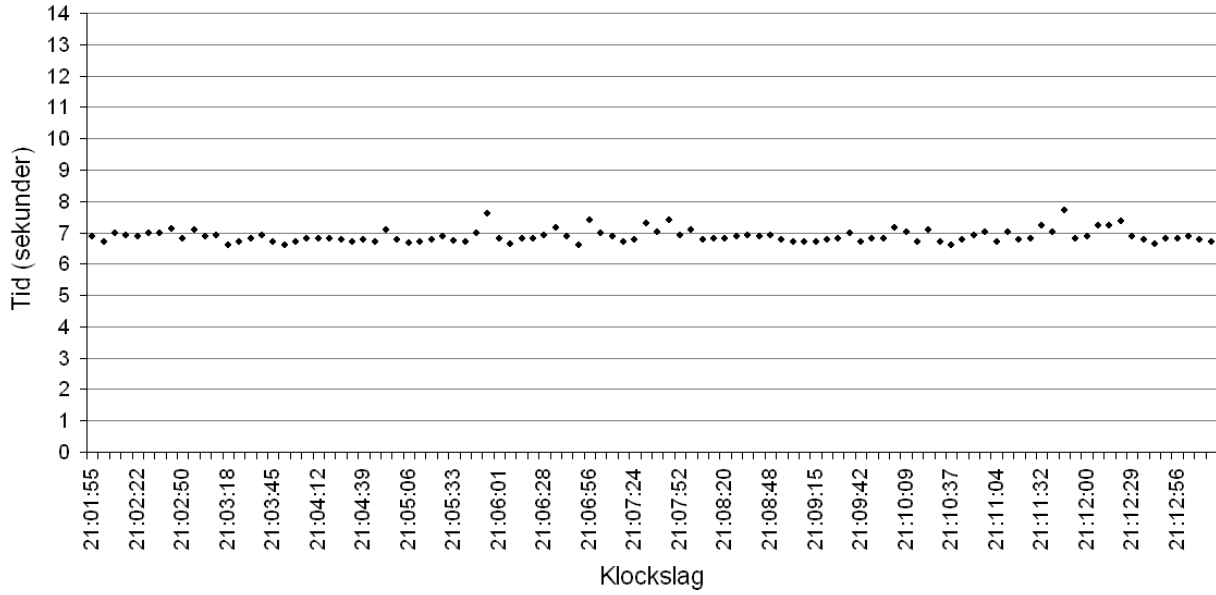
Test 20032
Typ D. Svarstid Light externt.



Test 20033
Typ B. Uppstart Light internt



Test 20034
Typ B. Svarstid Light internt.



Bilaga B

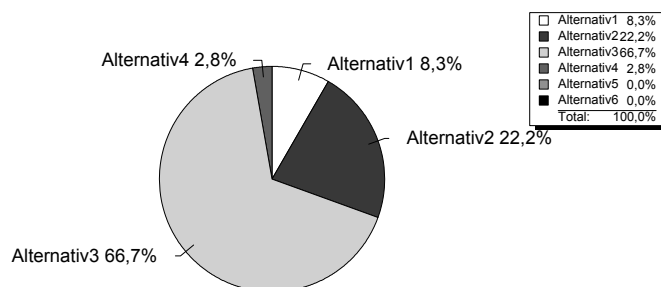
Resultat av enkätundersökning

Fråga 1

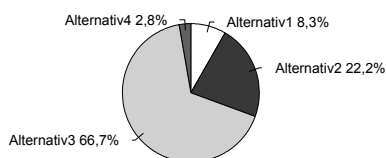
Var använder du tidrapporteringssystemen för det mesta?

1. DaCapos stockholmskontor
2. Stockholm på Ericsson
3. DaCapos göteborgskontor
4. Någon annan plats

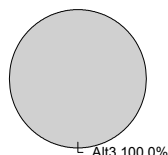
Samtliga inkl. "svarar ej"



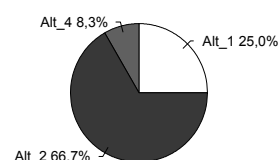
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

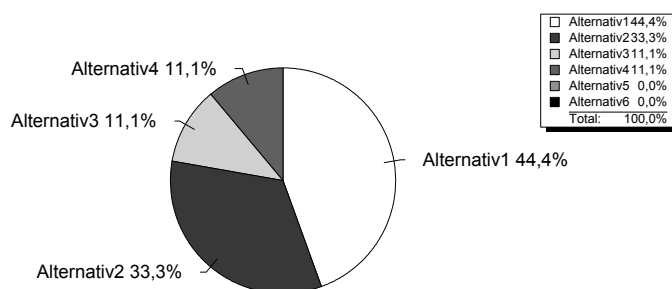


Fråga 2

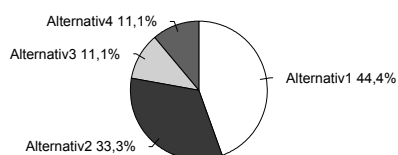
Hur ofta använder du tidrapporteringssystemen på annan plats än den du angivit ovan?

1. Aldrig
2. 1-3 ggr / kvartal
3. 1-3 ggr / månad
4. 1-3 ggr / vecka

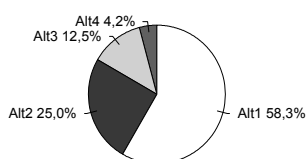
Samtliga inkl. "svarar ej"



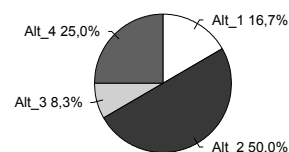
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

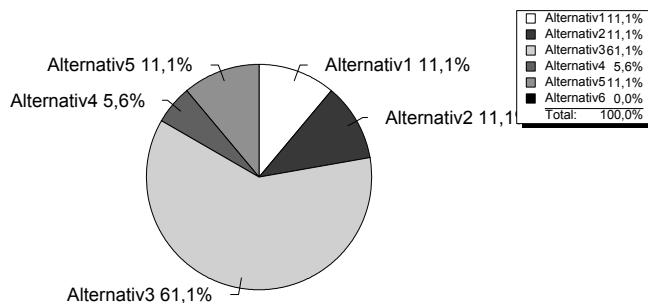


Fråga 3

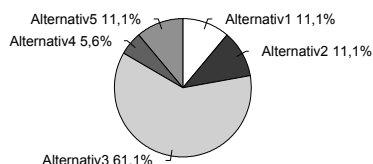
Hur upplever du prestanda för tidrapportering Beta?

1. mkt dålig
2. dålig
3. acceptabel
4. god
5. mkt god
6. svarar ej

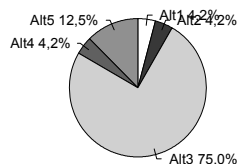
Samtliga inkl. "svarar ej"



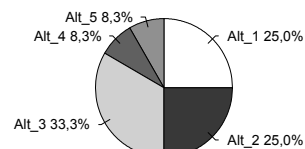
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

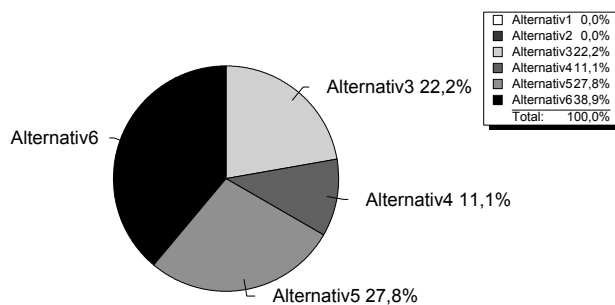


Fråga 4

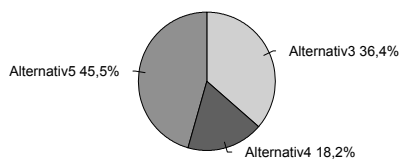
Hur upplever du prestanda för tidrapportering Light?

1. mkt dålig
2. dålig
3. acceptabel
4. god
5. mkt god
6. svarar ej

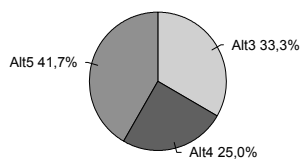
Samtliga inkl. "svarar ej"



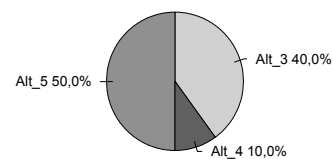
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

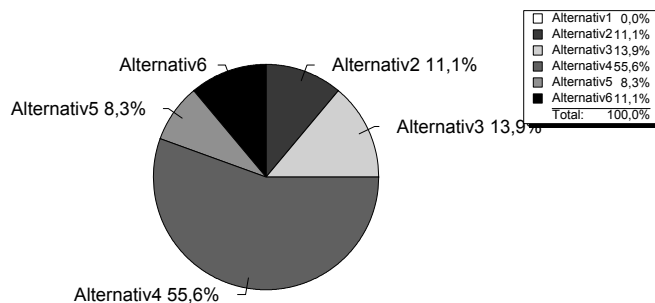


Fråga 5

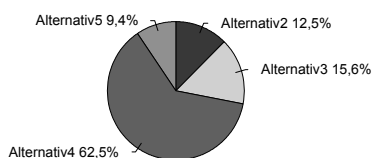
Hur stora variationer upplever du hos systemens prestanda från gång till gång för tidrapportering Beta?

1. extrema
2. stora
3. tydliga
4. små
5. ej märkbara
6. svarar ej

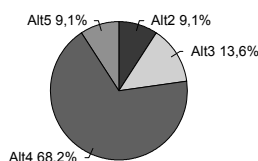
Samtliga inkl. "svarar ej"



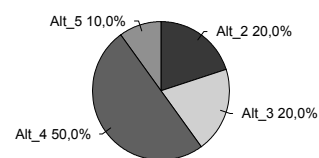
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

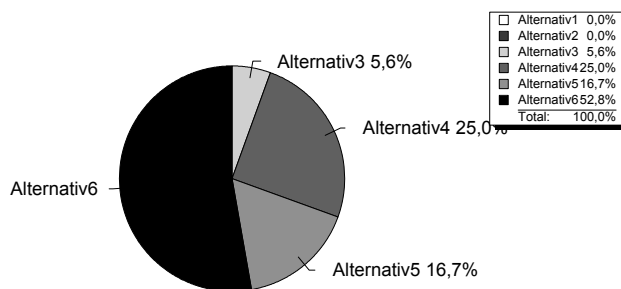


Fråga 6

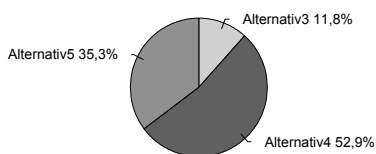
Hur stora variationer upplever du hos systemens prestanda från gång till gång för tidrapportering Light?

1. extrema
2. stora
3. tydliga
4. små
5. ej märkbara
6. svarar ej

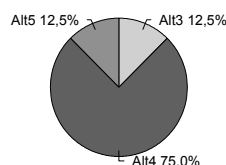
Samtliga inkl. "svarar ej"



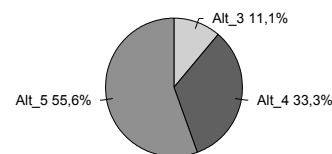
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

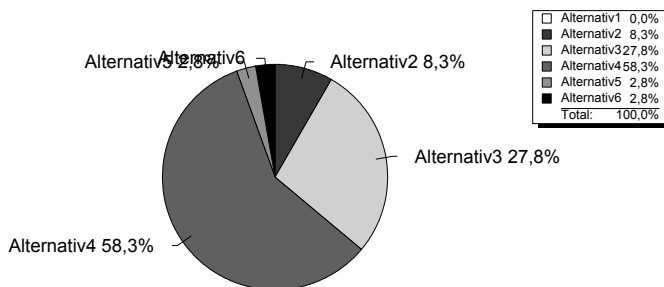


Fråga 7

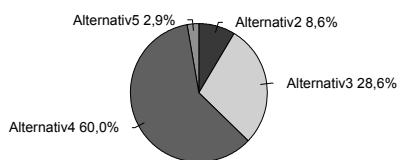
Hur ofta upplever du att systemen ej svarar vid användning för tidrapportering Beta?

1. mkt ofta
2. ofta
3. ibland
4. sällan
5. aldrig
6. svarar ej

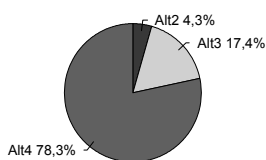
Samtliga inkl. "svarar ej"



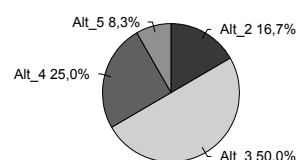
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

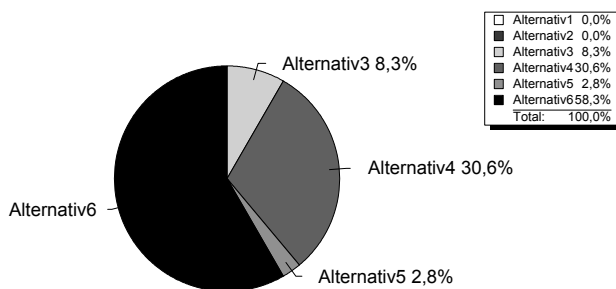


Fråga 8

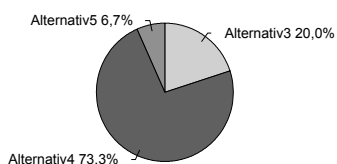
Hur ofta upplever du att systemen ej svarar vid användning för tidrapportering Light?

1. mkt ofta
2. ofta
3. ibland
4. sällan
5. aldrig
6. svarar ej

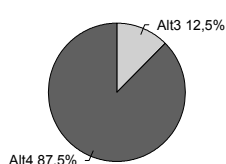
Samtliga inkl. "svarar ej"



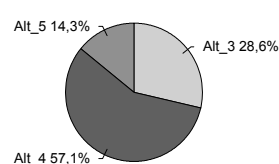
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

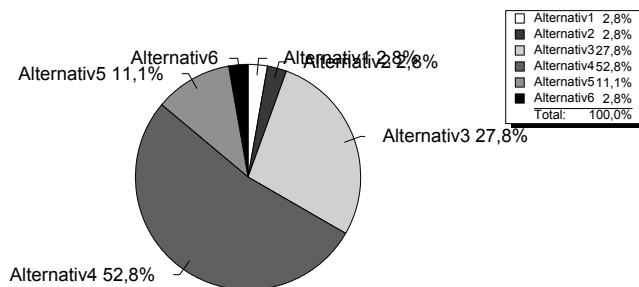


Fråga 9

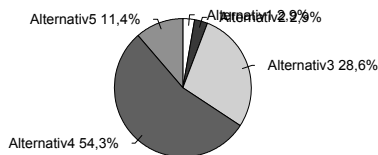
Hur ofta genererar systemen felmeddelanden vid användning för tidrapportering Beta?

1. mkt ofta
2. ofta
3. ibland
4. sällan
5. aldrig
6. svarar ej

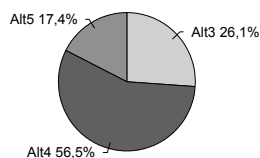
Samtliga inkl. "svarar ej"



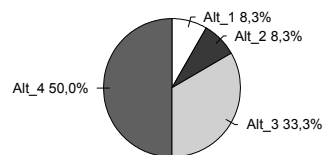
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"



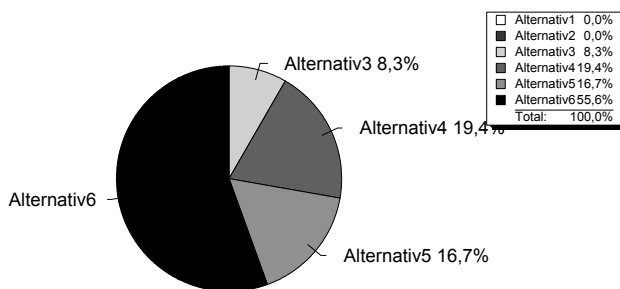
Fråga 10

Hur ofta genererar systemen felmeddelanden vid användning för tidrapportering Light?

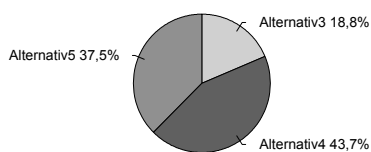
Hur ofta genererar systemen felmeddelanden vid användning för tidrapportering light?

1. mkt ofta
2. ofta
3. ibland
4. sällan
5. aldrig
6. svarar ej

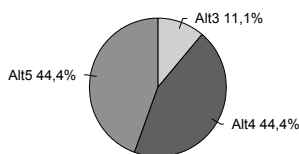
Samtliga inkl. "svarar ej"



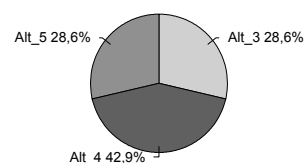
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

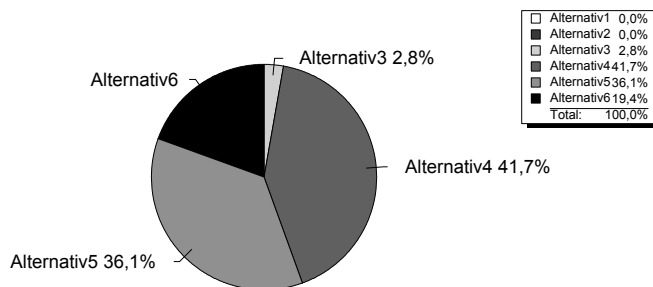


Fråga 11

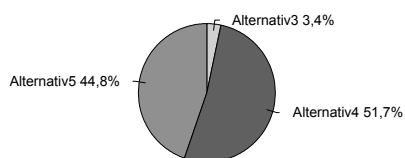
Hur ofta uppstår fel utan att du får felmeddelande för tidrapportering Beta?

1. mkt ofta
2. ofta
3. ibland
4. sällan
5. aldrig
6. svarar ej

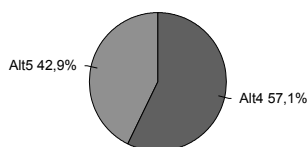
Samtliga inkl. "svarar ej"



Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

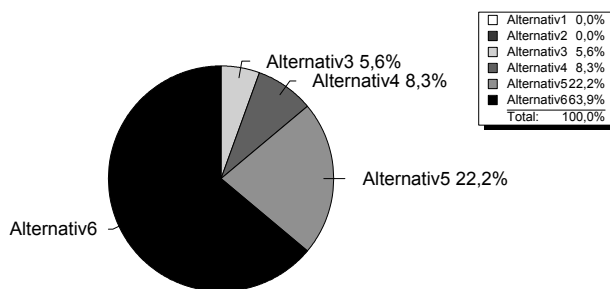


Fråga 12

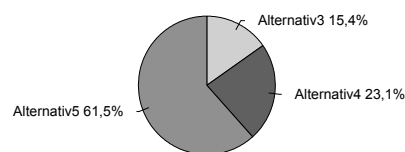
Hur ofta uppstår fel utan att du får felmeddelande för tidrapportering Light?

1. mkt ofta
2. ofta
3. ibland
4. sällan
5. aldrig
6. svarar ej

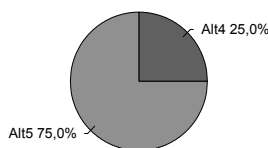
Samtliga inkl. "svarar ej"



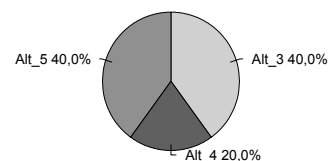
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Interna (Göteborg) utan "svarar ej"

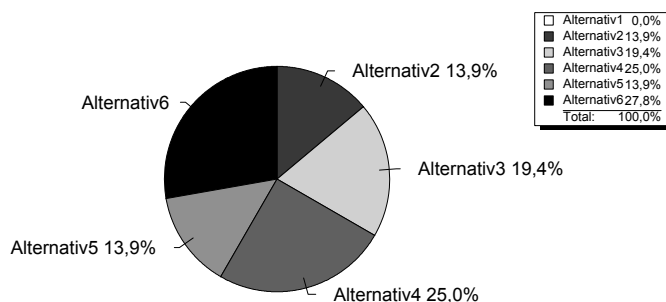


Fråga 13

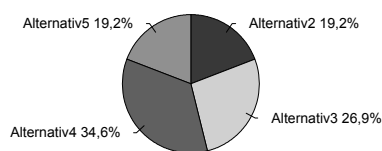
Hur upplever du kommunikationen med de som underhåller och vidareutvecklar systemen?

1. mkt dålig
2. dålig
3. acceptabel
4. god
5. mkt god
6. svarar ej

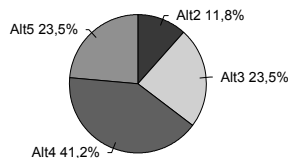
Samtliga inkl. "svarar ej"



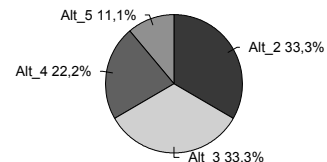
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

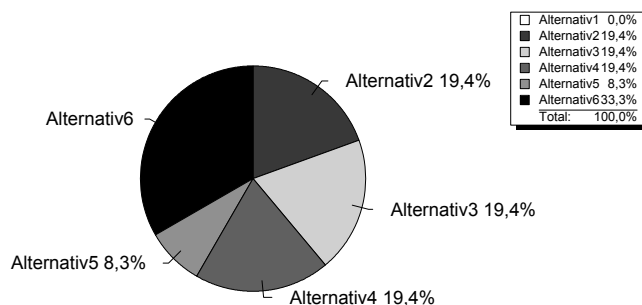


Fråga 14

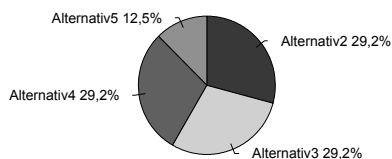
Hur upplever du din möjlighet att påverka vid underhåll/vidareutveckling av systemen?

1. mkt dålig
2. dålig
3. acceptabel
4. god
5. mkt god
6. svarar ej

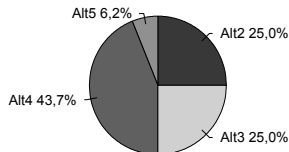
Samtliga inkl. "svarar ej"



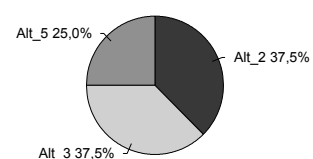
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

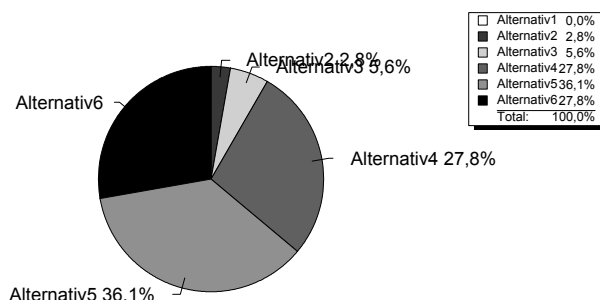


Fråga 15

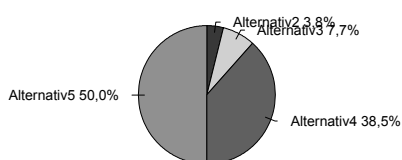
Skillnad mellan det nya tidrapporteringssystemen och hur det fungerade innan (Excel) med avseende på användarvänlighet. De nya systemen är:

1. mkt sämre
2. sämre
3. lika bra
4. bättre
5. mkt bättre
6. svarar ej

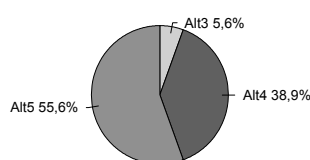
Samtliga inkl. "svarar ej"



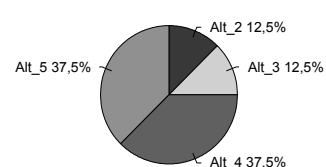
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

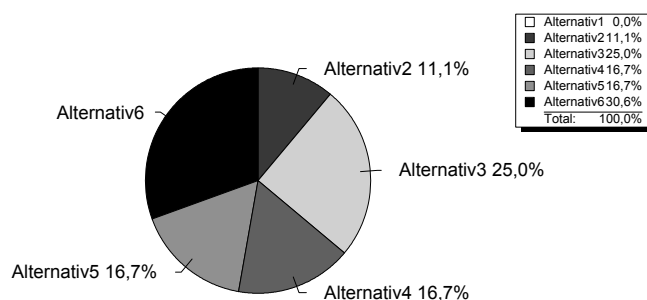


Fråga 16

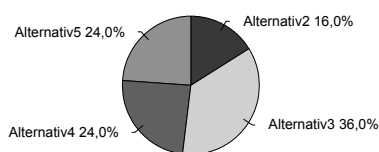
Skillnad mellan det nya tidrapporteringssystemen och hur det fungerade innan (Excel) med avseende på pålitlighet. De nya systemen är:

1. mkt sämre
2. sämre
3. lika bra
4. bättre
5. mkt bättre
6. svarar ej

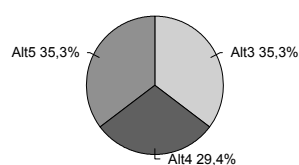
Samtliga inkl. "svarar ej"



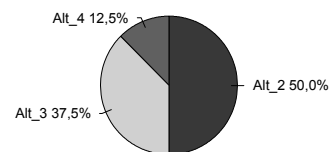
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

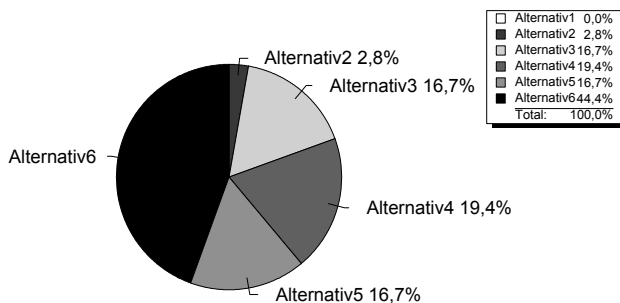


Fråga 17

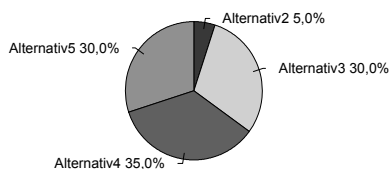
Skillnad mellan det nya tidrapporteringsystemen och hur det fungerade innan (Excel) med avseende på säkerhet. De nya systemen är:

1. mkt sämre
2. sämre
3. lika bra
4. bättre
5. mkt bättre
6. svarar ej

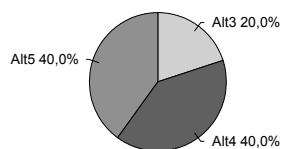
Samtliga inkl. "svarar ej"



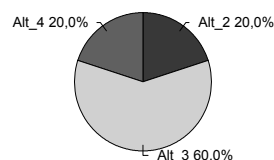
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

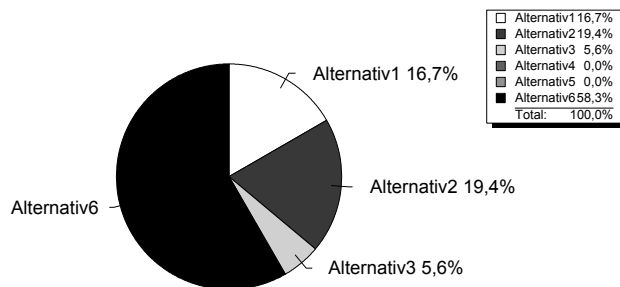


Fråga 18

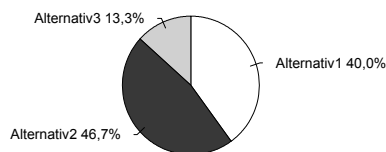
Skillnad mellan att använda tidrapporteringsystemet Beta på Göteborgskontoret och att göra det någon annanstans (prestanda)? Att rapportera utifrån är:

1. mkt sämre
2. sämre
3. lika bra
4. bättre
5. mkt bättre
6. svarar ej

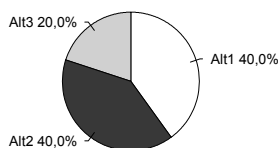
Samtliga inkl. "svarar ej"



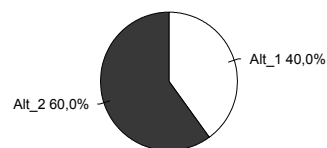
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

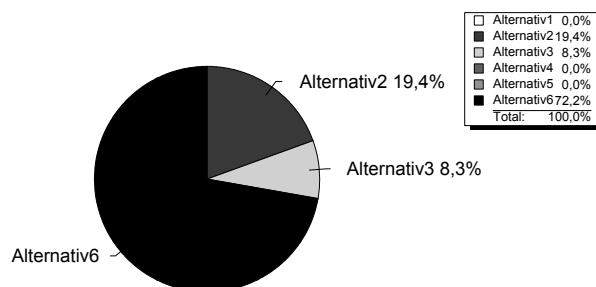


Fråga 19

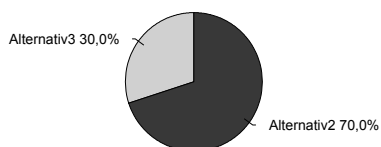
Skillnad mellan att använda tidrapporteringssystemet Beta på Göteborgskontoret och att göra det någon annanstans (pålitlighet)? Att rapportera utifrån är:

1. mkt sämre
2. sämre
3. lika bra
4. bättre
5. mkt bättre
6. svarar ej

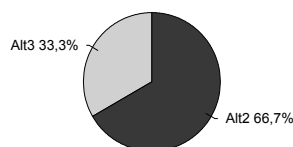
Samtliga inkl. "svarar ej"



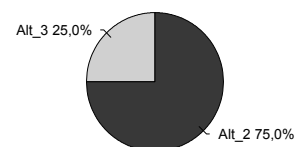
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

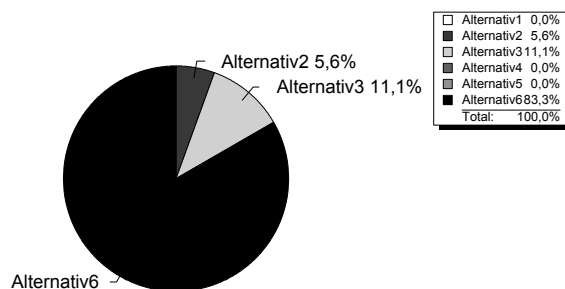


Fråga 20

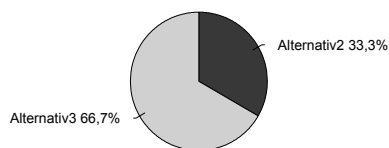
Skillnad mellan att använda tidrapporteringssystemet Beta på Göteborgskontoret och att göra det någon annanstans (säkerhet)? Att rapportera utifrån är:

1. mkt sämre
2. sämre
3. lika bra
4. bättre
5. mkt bättre
6. svarar ej

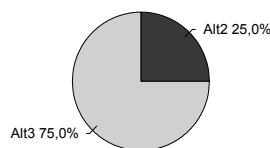
Samtliga inkl. "svarar ej"



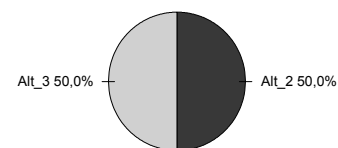
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Interna (Göteborg) utan "svarar ej"

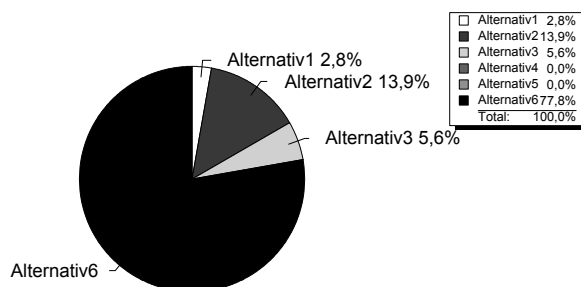


Fråga 21

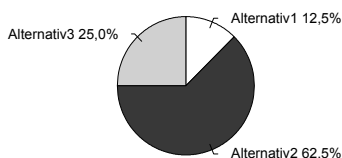
Skillnad mellan att använda lightversionen på Göteborgskontoret och att göra det någon annanstans (prestanda)? Att rapportera utifrån är:

1. mkt sämre
2. sämre
3. lika bra
4. bättre
5. mkt bättre
6. svarar ej

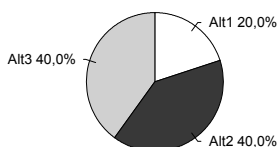
Samtliga inkl. "svarar ej"



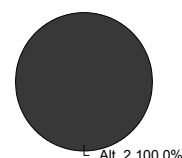
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

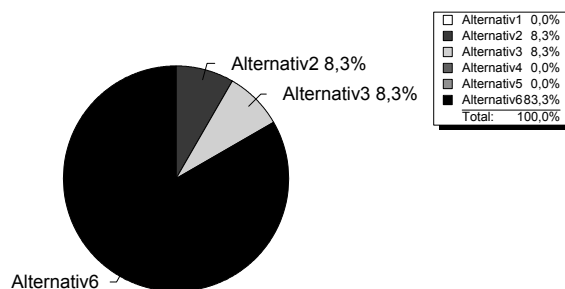


Fråga 22

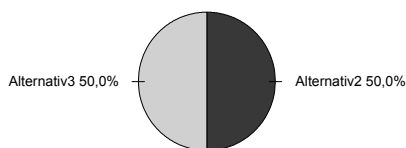
Skillnad mellan att använda lightversionen på Göteborgskontoret och att göra det någon annanstans (pålitlighet)? Att rapportera utifrån är:

1. mkt sämre
2. sämre
3. lika bra
4. bättre
5. mkt bättre
6. svarar ej

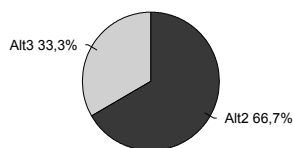
Samtliga inkl. "svarar ej"



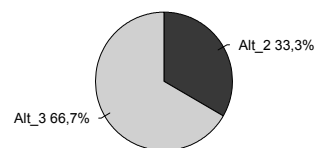
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

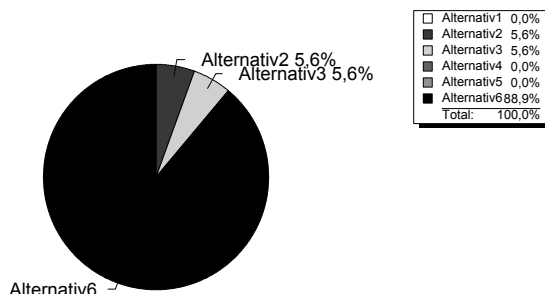


Fråga 23

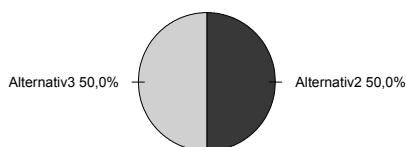
Skillnad mellan att använda lightversionen på Göteborgskontoret och att göra det någon annanstans (säkerhet)? Att rapportera utifrån är:

1. mkt sämre
2. sämre
3. lika bra
4. bättre
5. mkt bättre
6. svarar ej

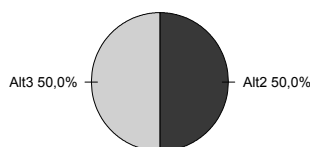
Samtliga inkl. "svarar ej"



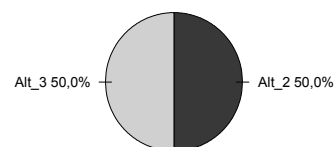
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"

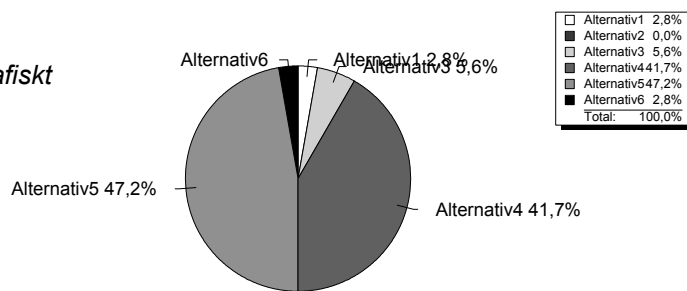


Fråga 24

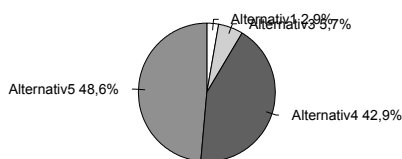
Hur viktigt anser du att det är att man kan fylla i sin tidrapport var som helst (dvs att systemet är geografiskt oberoende)?

1. helt oviktigt
2. ganska oviktigt
3. tja, sådär...
4. viktigt
5. mkt viktigt
6. svarar ej

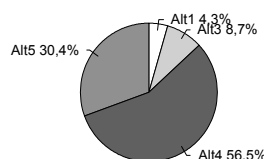
Samtliga inkl. "svarar ej"



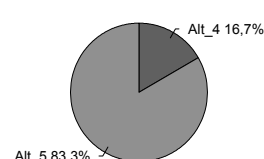
Samtliga utan "svarar ej"



Interna (Göteborg) utan "svarar ej"



Externa utan "svarar ej"



Enkätundersökningens samtliga svar

Enkät svar

Fråga	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	
1	1	2	4	3	3	2	2	3	3	3	3	3	2	3	1	3	3	3	3	3	3	3	1	2	3	3	3	3	2	3	2	3	3	3	3	2	3
2	1	2	2	1	1	2	2	2	4	1	1	2	3	3	2	1	1	1	2	1	3	1	4	4	2	2	2	1	1	2	2	1	1	3	4	1	
3	3	3	1	3	5	3	4	3	4	3	5	3	2	3	3	3	3	3	3	2	5	3	2	2	1	3	3	3	1	3	1	3	3	3	5	3	
4	5	5	3	3	6	3	5	6	6	6	6	5	6	5	3	6	6	5	6	4	6	5	3	4	3	3	3	4	6	6	5	6	4	3	5	5	
5	4	3	5	4	4	4	4	4	4	4	5	3	2	4	4	4	6	2	4	5	6	3	2	3	4	4	4	4	3	4	6	4	2	4	6	4	
6	4	5	5	4	6	4	5	6	6	6	6	4	6	4	4	6	6	4	6	5	6	6	3	5	6	6	4	6	6	5	6	3	4	6	6	6	
7	2	3	4	4	4	3	3	3	4	4	4	4	4	4	3	4	4	2	4	3	6	4	3	3	3	4	4	2	4	4	5	3	4	4	4	4	
8	6	4	4	4	6	6	3	6	6	6	6	4	6	6	3	6	6	4	6	3	6	6	4	4	6	6	4	6	6	5	6	4	4	6	4	6	4
9	4	2	4	5	5	3	4	4	4	3	4	4	3	4	3	4	4	3	4	5	6	4	3	4	3	4	4	5	4	3	1	4	3	3	4	4	
10	6	5	4	5	6	6	4	6	6	6	6	4	6	6	3	6	6	4	6	5	6	6	3	4	6	4	5	6	6	6	5	6	5	3	6	4	
11	4	6	6	5	4	5	5	5	6	4	5	5	4	4	6	4	5	5	4	5	6	5	3	4	4	5	6	6	4	4	5	6	4	4	5	4	
12	6	5	6	5	6	6	5	6	6	6	6	5	6	6	6	6	6	5	6	5	6	6	3	4	4	6	5	6	6	6	3	6	5	4	6	4	
13	4	5	3	4	6	2	3	6	6	5	5	6	6	4	6	5	3	6	2	3	6	4	2	3	3	4	4	4	2	4	6	4	6	3	4	5	
14	6	5	2	4	4	2	3	6	6	4	5	6	3	3	6	2	4	6	2	2	6	2	3	2	3	3	3	6	6	4	6	4	6	3	5	4	
15	6	6	2	6	6	3	6	6	5	5	5	5	5	3	5	4	4	5	4	5	6	5	4	4	5	5	6	5	5	4	6	4	6	4	4	4	
16	6	6	2	6	6	2	6	6	6	5	5	4	4	3	2	3	4	4	3	5	6	5	2	3	3	5	6	6	3	3	6	4	6	4	3	5	
17	6	6	6	6	6	3	6	6	6	5	5	4	6	4	6	4	4	3	4	6	6	5	2	4	3	5	6	3	6	3	6	5	6	4	3	5	
18	6	6	1	6	6	2	6	2	6	1	6	1	1	3	6	6	6	1	6	2	6	1	6	2	2	2	6	6	6	6	6	6	6	3	2	6	
19	6	6	3	6	6	2	6	6	6	6	6	2	2	3	6	6	6	2	6	2	6	3	6	2	2	2	6	6	6	6	6	6	6	6	6	6	
20	6	6	6	6	6	2	6	6	6	6	6	3	6	3	6	6	6	6	6	2	6	3	6	3	6	6	6	6	6	6	6	6	6	6	6	6	6
21	6	6	2	6	6	2	6	6	6	1	6	3	6	6	6	6	6	2	6	2	6	6	6	6	2	6	6	6	6	6	6	6	6	3	6	6	
22	6	6	3	6	6	2	6	6	6	6	6	2	6	6	6	6	6	3	6	2	6	6	6	3	6	6	6	6	6	6	6	6	6	6	6	6	6
23	6	6	6	6	6	2	6	6	6	6	6	3	6	6	6	6	6	6	6	2	6	6	6	6	3	6	6	6	6	6	6	6	6	6	6	6	6
24	4	5	5	4	5	5	5	4	1	5	5	4	5	5	5	4	4	4	5	5	6	3	4	5	4	4	4	4	4	4	5	4	4	5	5	3	

Bilaga C

Teknik

Teknisk miljö

		1999	2003
Webbserver	<i>Hårdvara</i>	<ul style="list-style-type: none"> • Intel Pentium II 266 MHz • 256 MB • HD 8 GB 	<ul style="list-style-type: none"> • Intel Pentium III 800 MHz • 512 MB • HD 36 GB som raid5
	<i>Mjukvara</i>	<ul style="list-style-type: none"> • Windows NT Server 4.0 • Microsoft IIS 4 	<ul style="list-style-type: none"> • Windows NT Server 4.0, Servicepak 6 • Microsoft IIS 4
Databasserver	<i>Hårdvara</i>	<ul style="list-style-type: none"> • Intel Pentium Pro 200 MHz • 256 MB • HD 8 GB 	<ul style="list-style-type: none"> • Dual Intel Pentium III 800 MHz • 512 MB • HD 36 GB raid5
	<i>Mjukvara</i>	<ul style="list-style-type: none"> • Windows NT Server 4.0 • Microsoft SQL Server 6.5 	<ul style="list-style-type: none"> • Windows NT Server 4 SP 6 • Microsoft SQL Server 7 SP 2
Klientdator	<i>Hårdvara</i>	<ul style="list-style-type: none"> • Intel Pentium Pro 200 MHz • 128 MB • HD 4 GB 	<ul style="list-style-type: none"> • Intel Pentium III 500 MHz • 512 MB • HD 80 GB
	<i>Mjukvara</i>	<ul style="list-style-type: none"> • Windows 98 • Microsoft Internet Explorer 4.01 	<ul style="list-style-type: none"> • Windows NT 2000 Professional • Microsoft Internet Explorer 5.5
Nätverk		<ul style="list-style-type: none"> • LAN 10 Mbit/s Ethernet (hubbat) • Bandbredd mot Internet 256 Kbit/s (Citat Solutions) • Bandbredd mot Internet 10 Mbit/s (Externt klient) • Modem 28.8 Kbit/s 	<ul style="list-style-type: none"> • LAN 100 Mbit/s Ethernet (switchat) • Bandbredd mot Internet 10 Mbit/s (Citat Solutions) • Bandbredd mot Internet 10 Mbit/s (Externt klient)
Kommunikation		<ul style="list-style-type: none"> • Internt Http över LAN • Externt Https RC4-Export cipher, 40-bit key över Internet 	<ul style="list-style-type: none"> • Internt Http över LAN • Externt Https RC4 cipher, 128-bit key över Internet