

Användbarhetstestning

- Ur en programmerares perspektiv

- MarketTools



LUNDS UNIVERSITET
Campus Helsingborg

LTH Ingenjörshögskolan vid Campus Helsingborg
Institutionen för datavetenskap

Examensarbete:
Dan Andersson

© Copyright Dan Andersson

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Lunds universitet
Lund 2015

Sammanfattning

MarketTools är ett auktionsverktyg som har skapats av Diflex AB. Ett auktionsverktyg som är jämförbart med "Blocket" eller "eBay", fast som är specialiserat för större auktioner så som till exempel lantbruksmaskiner eller bilar. Grunden var redan skapad och det fanns en fungerade hemsida, www.markettools.se.

Examensarbetet gick delvis ut på att vidareutveckla denna hemsida som är gjord i ASP.net och lägga till ytterligare funktionalitet som skulle göra det enklare för administratörerna av hemsidan för avslutade auktioner. En utskriftssida för avslutade auktioner och ett fakturaverktyg utvecklades och skulle senare användartestas.

Det primära syftet med examensarbetet var att undersöka olika testmetoders effektivitet efter tid för att förbereda testerna mot hur mycket användbar feedback som ges tillbaka från dessa - ur en programmerares perspektiv. Fokus låg på att få en djupare inblick i olika testmetoder som anses lämpliga för ett projekt av denna typ. Ett par metoder för att mäta användbarheten från detta perspektiv skulle väljas och användes genom hela projektet.

Examensarbetet resulterade i en jämförelse mellan direkt observation och undersökningar och sedan ställa deras styrkor respektive svagheter mot varandra.

Ur en programmerares perspektiv handlar allt om att konkret få ut informationen om var problemen ligger, vilket den direkta observationen ger väldigt mycket bättre än undersökningarna. Men med undersökningarna är det möjligt att man får in fler svar, samtidigt som risken finns att dessa inte ger konkreta svar på de frågor som programmeraren söker svar på. Detta kan resultera i information som inte kan användas vilket i sin tur inte medför att sidan blir mer användarvänlig eller till och med sämre än utgångsläget.

Nyckelord: Användbarhet, ASP.net, Diflex, Testning, Effektivitet.

Abstract

MarketTools is an auction tool created by Diflex AB. An auction tool that is comparable to “Blocket” or “eBay”, that is specialized for larger auctions like, for example, agricultural machinery or automobiles. The foundation was already created and there was a functioning website, www.markettools.se.

The work was partly on the further development of this website which is made in ASP.net and needed to add additional functionality that would make it easier for administrators of the website for the auctions ended. A print page for complete auctions and an invoice tool was developed and would later be user tested.

The primary aim of the thesis was to investigate the effectiveness of different test methods for time to prepare for the tests against how much useful feedback that is given back from them - from a programmer's perspective. The focus was on getting a deeper insight into the different test methods deemed appropriate for a project of this type. A couple of methods to measure the usefulness of this approach would be selected and used throughout the project.

The work resulted in a comparison between direct observation and studies and then set their strengths and weaknesses against each other.

From a programmer's perspective, it's all about concrete get the information about where the problem lies, as the direct observation gives much better than the surveys. But for the surveys, it is possible to receive more answers, while the risk is that these do not give concrete answers to the questions where the programmer is looking for answers. This may result in information that can not be used, which in turn does not cause the site to become more user friendly - or in some cases even worse than the starting point.

Keywords: Usability, ASP.net, Diflex, Testing, Efficiency.

Förord

Denna rapport är resultatet av ett examensarbete på 15 veckor (22,5 Högskolepoäng) som utförts på heltid under sommaren 2014 på Diflex AB i Ängelholm. Examensarbetet ingår som avslutande moment i mina högskoleingenjörstudier vid Lunds Tekniska Högskola, Campus Helsingborg.

Ett stort tack vill jag först och främst rikta till personal på Diflex AB. Då främst till Diflex ABs VD Kjell Ivarsson och Ingemar Nilsson på Ängelholm Redovisning för handledning och stor hjälp under projektets gång.

Även ett stort tack till all övrig personal, samt Mikael Karlsson som tagit sig tid att hjälpa mig komma in i systemet och få tillgång till servern för att underlätta arbetet jag utfört under examensarbetets gång. Stort tack för att jag fått genomföra detta mycket intressanta arbete.

Jag vill även rikta ett speciellt tack till min handledare på LTH, Mats Lilja för hans stöd och engagemang i detta examensarbete samt för god vägledning under arbetets utveckling.

Slutligen vill jag tacka min examinator på LTH, Christin Lindholm som har hjälpt mig att strukturera examensarbetet och ordna de sista bitarna i processen som ledde till en bra rapport och presentation.

Innehållsförteckning

1 Inledning	1
1.1 Riktlinjer för läsning av rapport	1
1.2 Bakgrund	1
1.3 Arbetets relation till utbildning	1
1.4 Syfte	2
1.5 Mål	2
1.6 Problemformulering	2
1.7 Avgränsningar	2
1.8 Förväntat resultat	3
2 Metod	4
2.1 Vad bör användbarhetstesterna användas på?	5
2.1.1 Bugfixar	5
2.1.2 Utskriftsfunktion för avslutade auktioner	6
2.1.3 Utskriftsfunktion för fakturaunderlag	7
2.2 Testobjekt	8
2.3 Mjukvaruloggning	8
2.4 Generella anledningar för struktur av testerna	8
2.5 Direkt observation (bilaga 1)	8
2.5.1 Hur designades detta test?	8
2.6 Undersökningar (bilaga 2)	9
2.6.1 Hur designades detta test?	9
2.7 Källkritik	9
3 Användbarhetstester	10
3.1 Vad är användbarhetstestning?	10
3.2 Olika definitioner	10
3.3 Bedömning av metoder	11
3.4 Vad är skillnaden mellan tekniker och metoder?	11
3.5 Metoder	12
3.5.1 <i>Expertutvärdering</i>	12
3.5.2 <i>Observation</i>	12
3.5.3 Undersökningar	13
3.5.4 Experimentell utvärdering	13
3.6 Tekniker	13
3.6.1 Direkt observation	13
3.6.2 Mjukvaruloggning	14
3.6.3 Intervju	14
3.7 Utvärderare	14
3.7.1 WAMMI	16
3.7.2 Hur valdes testpersonerna?	17
3.7.3 Exempel på användbar och ej användbar feedback	17
4 Teknisk Bakgrund	18
4.1 Vad är ASP.net?	18
4.1.1 C#	18
4.1.2 Databaser och SQL	19
4.1.3 Javascript	19
4.1.4 CSS och HTML	19
4.2 php	20

4.2.1 phpMyAdmin	20
4.3 Grundsyste met	20
4.4 Uppgiftsbeskrivning	21
4.5 Första anblick	22
4.6 Omdirigering till lokal databas för utveckling	22
4.7 AjaxToolKit	23
5 Resultat	24
5.1 Direkt observation	26
5.1.1 Fördelar	26
5.1.2 Nackdelar	26
5.2 Undersökningar	27
5.2.1 Fördelar	27
5.2.2 Nackdelar	27
6 Slutsats	28
6.1 Direkt observation	28
6.2 Undersökningar	28
6.3 Sammanställning	29
6.4 Framtida utvecklingsmöjligheter	29
7 Källor	30
7.1 Testmetoder	30
7.2 Företaget	31
7.3 Programmeringsspråk	31
7.3.1 ASP.net	31
7.3.2 C#.....	31
7.3.3 SQL.....	32
7.3.4 Javascript.....	32
7.3.6 php.....	32
7.4 Allmän information	32
8 Terminologi	33
9 Bilagor	35
9.1 Bilaga 1: Direkt observation	36
9.1.1 Användbarhetstest.....	36
9.1.2 Frågor.....	37
9.1.3 Uppgifter	38
9.1.4 Frågor angående test i efterhand	38
9.1.5 Scenario 1.....	39
9.1.6 Scenario 2.....	39
9.2 Bilaga 2: Undersökningar	40
9.2.1 Uppgift 1	42
9.2.2 Uppgift 2	42
9.3 Bilaga 3: Tidsplan	43

1 Inledning

MarketTools är ett auktionsverktyg som har skapats av Diflex AB. Ett auktionsverktyg som är jämförbart med "Blocket" or "eBay", fast är specialiserat för större auktioner så som lantbruksmaskiner eller bilar. Tanken bakom detta projekt är att ha en grundsida som enkelt ska kunna specialanpassas till företag som är intresserade av att påbörja en sådan marknad för sin försäljning. Examensarbetet handlar om att göra tillägg till MarketTools och göra användartester på dessa och samtidigt jämföra olika metoder och tekniker för att se vilken som var mest effektiv för detta syfte.

1.1 Riktlinjer för läsning av rapport

- När det skrivs hemsidan i texten så syftar det på hemsidan som examensarbetet görs på, www.markettools.se.
- När "projekt" nämns så menar det projekt som är motsvarande till examensarbetet – webbutveckling.
- När "programmeraren" nämns så menar detta författaren av detta examensarbete.
- Om texten är inflyttad så betyder det att det är ett direkt citat.
- Kapitelnamn som hänvisas till är i fetstil.
- [] betyder hänvisning till **kapitel 7, Källor**.
- Asterisk "*" betyder hänvisning till **kapitel 8, Terminologi**.

1.2 Bakgrund

Diflex AB grundades 1963, men har sen starten haft en rad olika namn då företaget varit med i olika kedjor t ex, Ericssons City, Kontors Landslaget och Koneo. År 2011 är första gången sedan starten företaget marknadsförs under ett eget namn, Diflex AB.

Genom åren har alltid Diflex inriktning varit datorer men vissa år har företaget tagit lite sidospår så som Hem-PC, inredning och ett auktionssystem. Under de senaste 5 åren så har Diflex blivit mer av ett tjänsteföretag och idag är stor fokus på drifttjänster och konsultation.

Tidigare elev på Campus Helsingborg, LTH, Mikael Karlsson har tidigare varit ansvarig för auktionssystemet MarketTools som är ett sidoprojekt på företaget där ingen tog över utvecklingen av detta efter att Mikael valde att fokusera på andra saker i livet. Kontakt med Mikael gav att ge en bra inblick i hur systemet var uppbyggt. Mikael fanns tillgänglig för att göra överlämningen av projektet så smidig som möjligt.

1.3 Arbetets relation till utbildning

Datateknikutbildningen handlar om datorsystem, både programvara och hårdvara samt de tekniska gränssnitten mot den miljö i vilken datorsystemen skall verka. Utbildningen i datateknik präglas av ett ingenjörsmässigt arbetssätt och ger dig en bred teoretisk kunskapsbas inom datateknikområdet. För att verklighetsförankra och ge dig färdigheter kopplas denna kunskap under utbildningen till praktiska uppgifter, projekt och problemlösning.

Att arbeta som dataingenjör innebär att man kan arbeta inom hela fältet mellan teknik och människa. Ett datorsystem är sällan en isolerad del, utan finns i samverkan med människor och andra system. Tekniken håller på att få en allt

större betydelse för hur vi lever våra liv; tänk bara på allt vi idag utför med hjälp av en mobiltelefon.

Ovanstående är beskrivningen från LTH^[24] angående given utbildning, Högskoleingenjör i Datateknik. Eftersom det är en utbildning som handlar om programmering och att hitta ingenjörsmässiga lösningar inom området var MarketTools en bra grund att göra detta examensarbete på. Vidare så är fokus på användbarhetstestande som även det är något som studeras under utbildningens gång.

1.4 Syfte

Detta examensarbete valdes av programmeraren för att ge personer som jobbar inom webb- och testbranschen en bättre inblick i den information de för vidare till programmerare i sina projekt. Och även för självständiga programmerare som vill göra en så bra produkt som möjligt. Arbetet handlar om att undersöka vilken metod som producerar det bästa resultatet i slutändan, inte bara för testarnas egen skull, utan för alla involverade i produkten så som programmerare, testare, kund och användare.

1.5 Mål

Målet med detta examensarbete är inte att få en entydig bild av vilken metod för användartestande som är bäst, utan att se för- och nackdelar med de valda metoderna och jämföra dessa ur en programmerares perspektiv, med andra ord se vilken metod som ger den mest konkreta feedbacken utan att behöva bearbetas och tolkas av en testgrupp för att en programmerare ska förstå vilka ändringar som behöver göras på en hemsida. Fast inget av detta är möjligt innan de ytterligare funktionerna på www.markettools.se är skapade.

1.6 Problemformulering

Ett auktionssystem med tillhörande hemsida finns sedan tidigare tillgängligt på företaget men var inte fullständigt. Tester för användbarhet såväl som ett par funktioner saknas enligt begäran från företaget och kund.

Syftet med examensarbetet var att undersöka olika testmetoders effektivitet efter tid för att förbereda testerna mot hur mycket användbar feedback som ges tillbaka från dessa. Två metod för att mäta användbarheten från detta perspektiv skulle väljas och användas genom hela examensarbetet.

När man sammanfattade hela examensarbetet med en frågeställning så blev det nedanstående.

- Vilken metod av användartestning ger den mest konkreta feedbacken ur en programmerares perspektiv?

Hur detta kommer mätas beskrivs i **kapitel 1.8, Förväntat resultat.**

1.7 Avgränsningar

Det fanns ingen på Diflex AB som var insatt i ASP.net eller programmeringen av sidan, vilket ledde till en väldigt omständigt process för att få ut information om planeringen av uppbyggnaden av hemsidan. Det fanns dokumentation men denna var bara fokuserad på auktionsaspekten och inloggning på hemsidan.

1.8 Förväntat resultat

Det fanns inget entydigt resultat angående de mest optimala testmetoderna, utan det beror helt på vad det är för projekt, budget och tidsram. Men det förväntades att hitta en metod som var bäst anpassad för nedan angiven målgrupp och för ett projekt av denna form. Mer specifikt så inriktade sig programmeraren på att kunna underlätta kommunikationen inom ett projekt med en separat programmerings-avdelning, testgrupp och projektledare då de två sistnämnda inte alltid har någon inblick i vad som är användbar information för en programmerare.

Testerna gjordes bara på en specifik sorts personer som är anpassad till hemsidan, denna målgrupp involverar:

- Personer mellan åldrarna 20 till 80 år.
- Personer som är intresserade av lantbruksmaskiner och har intresse att köpa dessa.

De olika testerna utformades så att så mycket information som möjligt kunde samlas in. Information som samlades in var bland annat tiden det tar att utforma testet, svarstid, hur många som svarar beroende på hur många tester som planerar att göras. Av intresse var även hur mycket bearbetning av testresultaten som krävs och hur värdefullt det är för en programmerare att få fram ett dåligt eller ett ej bearbetat resultat. Det sistnämnda kommer att mätas subjektivt då studien bara har en programmerare i examensarbetet. Mätningen bestod av hur lång tid det tar att tolka vilka ändringar som behövs och sedan se om resultatet motsvarade vad som förväntades av personer som utfört testerna. Tiden för att tolka testerna och göra motsvarande ändringar i koden var satt till 10 timmar. Vilket gav tillräckligt med tid för att gå igenom alla testresultat och samtidigt göra ändringar utan att behöva stressa för mycket, vilket hade kunnat försämra resultatet. Detta förutsatte självklart att det inte krävs några stora ändringar på testsidorna, det vill säga sidorna som programmerades och gjordes användbarhetstester på.

Resultatet blev att examensarbetet gav en tydlig visning av för- och nackdelarna mellan de olika metoderna så att läsarna enkelt kommer kunna förstå skillnaderna mellan de valda metoderna. Resultatet kommer att ge riktlinjer för val av testmetoder i andra projekt i samma skala utan att behöva lägga för mycket tid på att efterforska olika metoder själva.

2 Metod

Vid läsning av metod rekommenderas det att ta fram **bilaga 3 (tidsplan)** för en bra bild av upplägget. Även om det bara var en programmerare som arbetat på detta projekt så var projektmodellen agil*. Detta valdes för att regelbunden kontakt med kund/beställare var det enda sättet att få feedback om resultatet av utvecklingen var enligt beställning. Normalt i ett större projekt jobbar ett team efter en kravspecifikation och tolkar den tillsammans. Sedan utvecklar efter det tills en prototyp finns tillgänglig för att visa kund/beställare. Därefter ges förslag på ändringar så är kund blir nöjd med denna. Men i detta projekt så kontaktades kunden redan innan en prototyp fanns tillgänglig, eftersom kravspecifikationen kunde misstolkas så att ingen tid slösades bort. Detta betydde att programmeraren satte sig på företaget efter att ha fått en kravspecifikation en dag tidigare och verbalt förklarade han tolkade denna. Sedan skapades en prototyp som visar hur det kommer att se ut i verkligheten.

För att detta projekt skulle vara möjligt, för detta projekt, så planerades det in en två veckors inlärningsperiod i början av projektet vars enda syfte är att förstå hur ASP.net används för att kunna ge en bra inblick i projektet. Detta för att enklare kunna förstå vad som redan finns tillgängligt, vad som behöver förbättras direkt, vad som var realistiskt att göra inom given tidsram och vad som är möjligt utifrån förväntningar från Diflex AB.

Tanken var först att sitta på företaget i Ängelholm och arbeta från sin personliga laptop för att enkelt ha tillgång till koden hemifrån ifall detta kräves. Denna plan ändrades efter en del hårdvaruproblem med sagt laptop vilket gjorde testningen av kod väldigt långsam och ibland ej körbar alls. Arbetet utfördes istället för det mesta hemifrån eftersom där det fanns tillgång till en dator med SSD för att koden snabbare skulle kunna köras lokalt, samt behålla den enkla tillgången till koden för rapportskrivande. Möte med Diflex bestämdes via mailkommunikation och programmeraren diskuterade hur utvecklingen hade gått sedan det senaste mötet. Det togs upp problem som hade tagit längre tid än planerat och idéer om hur man skulle kunna förbättra diverse saker på hemsidan. Detta diskuterades tills båda parter var nöjda och det fortsatte att arbetas hemifrån tills ett resultat var uppnått. Om Diflex var nöjda så börjades detta användartestas.

Feedbacken från testerna togs in och analyserades av programmeraren och delades in i olika kategorier och efteråt tas ett genomsnitt av resultaten fram som självklart är lite subjektivt beroende på hur dessa tolkades. För att få in så lite personlig tolkning som möjligt sattes olika kriterier upp för hur testerna skulle analyseras. Om de behöver omformuleras eller om de inte tillför något och lika bra kan kastas direkt. Tanken var att hitta en bra balans så att testresultaten inte behöver bearbetas från att testerna är klara tills att programmeraren fick reda på vad som behövde förbättras eller läggas till.

Detta gjordes över en skala där genomsnittet sätts ut per testmetod och jämföras med varandra. Exempelvis så kan den första metoden som testas få information som ej kunde användas från hälften av testerna och bra feedback från resten, detta gav metoden 40-50 % i sagt kategori. Men om en annan metod ger bra, direkt feedback som tydligt säger "ändra detta, detta fungerar inte" så får denna metod över 80 %. Mer om metoderna beskrivs i **kapitel 3.5, Metoder**, då lite inblick i kända metoder krävs.

2.1 Vad bör användbarhetstesterna användas på?

Den här delen av rapporten är till för att diskutera på vilka delar av hemsidan som användbarhetstesterna kunde användas på och varför dessa valdes eller inte valdes.

2.1.1 Bugfixar

För att koden över huvud taget skulle kunna köras efter att den överförts från servern till en lokal kopia på programmerarens dator så krävdes det en del ändringar i koden vilket ledde till många långa diskussioner med handledaren på företaget. Diskussionerna uppkom efter att programmet fungerade där de låg uppe på servern när man använde hemsidan*, men när programmet kördes lokalt så vägrade detta köra på grund av ett fel i en funktion med fel antal in-variabler. Detta löstes relativt snabbt då funktionen hittades och alla in-variabler ordnades i anropet till denna. Frågan löstes dock aldrig, varför koden fungerade som den låg på servern. Det mest troliga förslaget var dock att den körde en äldre version av koden, och den senaste programmeraren (Mikael Karlsson) sedan hade gått in för att ändra något och inte fullföljt detta. Programmet upptäckte sedan detta och körde en tidigare version av koden för att ha ett fungerande program.

Andra problem inkluderade att ett par tables* i databasen* inte uppdaterades först så att informationsuthämtning blev mycket mer komplicerad eller inte alls möjlig för de ytterligare funktioner som önskades tilläggas från Diflex AB. Detta krävde en väldigt genomgående överblick och förståelse för hur databasen var uppbyggd för att inte göra några drastiska ändringar som skulle förstöra någon annan funktion som redan fanns på hemsidan. Detta var inte fallet, utan vissa funktioner fick skrivas om för att optimera processen för att lägga till nya funktioner.

2.1.2 Utskriftsfunktion för avslutade auktioner


För att förenkla för de personer som skall hantera hemsidan och enkelt kunna få en bra överblick av vilka auktioner som avslutats mellan två datum och få alla relevanta uppgifter för att fysiskt kontakta köparen med det högsta budet skapades följande sida.


Nedan, i **figur 1**, visas den viktigaste delen av utskriftssidan som skapades under projektets gång och som bland annat användes för att göra testerna på.

Skriv ut avslutade auktioner

Välj två datum för att lista auktioner som avslutats inom den tidsperioden

Välj bara avslutade auktioner där priset är uppnått:

Från: 

Till: 

33 auktion(er) avslutade mellan 2014-01-01 och 2014-10-06

Titel	Högsta bud	Köpare (e-mail)	Telefonnummer	Fakturaunderlag
Ford Traktor - testauktion	85000			Klicka här!
Test4	20000			Klicka här!
En överdrivet lång titel för att testa en väldigt specifik sak	55000			Klicka här!

Figur 1: Utskriftssida ^[15]

2.1.3 Utskriftsfunktion för fakturaunderlag

Uppgifter som skall stå med i fakturan kan bekvämt fås via en utskriftsfunktion efter det att köpet är accepterat av båda parter.

Nedan, i **figur 2**, visas den viktigaste delen av fakturaunderlagssidan som skapades under projektets gång och som bland annat användes för att göra testerna i examensarbetet på.

Fakturaunderlag

Titel:

En överdrivet lång titel för att testa en väldigt specifik sak

Auktionsnummer:

318

Alias (köpare):

██████████

Högsta bud:

55000

Slagavgift (total):

5000

Slagavgift (Diflex):

2500

Email (köpare):

██

Skriv ut

Figur 2: Fakturaunderlag ^[15]

2.2 Testobjekt

Av alla saker som lagts till och ändrats, samt de saker som redan fanns på hemsidan som ej testats tidigare under projektets gång. Så är utskriftssidorna överlägset de bästa sidorna att använda för genomförandet av användbarhetstesterna. De är helt utvecklade av programmeraren från grunden och har samtidigt ett tydligt mål: att skriva ut något specifikt objekt från databasen vilket enkelt kan formuleras om till en tydlig testbeskrivning. Testerna finns tillgängliga som **bilaga 1 och 2**.

2.3 Mjukvarulogging

Denna metod var tvungen att väljas bort på grund av tidsbrist och sågs som den minst effektiva då den tekniskt sett ingår i de andra två metoderna om de spelas in visuellt på skärmen med program som Fraps*.

2.4 Generella anledningar för struktur av testerna

- Varför Google Chrome och Firefox?

Det är ett väl känt faktum inom området att Internet Explorer inte erbjuder support för till exempel Adobe Flash Player* och många andra viktiga instrument. Därav rekommenderas alltid att användaren bör hålla sig undan Internet Explorer då många webplatser baseras på dessa verktyg.

- Varför dessa frågor?

Frågorna baserades på liknande tester och beskrivningar av hur ett test av denna typ är strukturerad generellt. Detta i kombination med att förstå andra metoder och tekniker så att det inte blev någon hybrid av olika tekniker, utan att det höll sig inom ramarna av det som skulle testas.

2.5 Direkt observation (bilaga 1)

Den direkta observationen gick till på följande vis. Person inom given målgrupp frågades om denna hade tid att testa hemsidan. En laptop sattes upp framför personen med tillgång till den lokala versionen av hemsidan för att slippa ha tillgång till internet för att göra det så bekvämt för testpersonen som möjligt. **Bilaga 1** följdes då den kursiva texten lästes upp av testansvarig. Både ljud och video (av skärmen) spelades in efter att testpersonen gett sitt medgivande att detta är ok.

2.5.1 Hur designades detta test?

När detta test skapades var idén redan klar om hur testet skulle utföras, enligt beskrivning i ovanstående rubrik, **2.5 Direkt Observation**. Det designades så att de viktigaste frågorna från test så som WAMMI fanns med och sedan undveks ”The Hawthorne Effect”. Mer om detta beskrivs under rubrik, **3.6.1 Direkt observation**.

2.6 Undersökningar (bilaga 2)

Bilaga 2 mailades ut till en lista personer i given målgrupp som var ihopsamlad av Diflex AB för att få så korrekt målgrupp som möjligt. I mailet fanns instruktioner och konkreta frågor som personen ombads att svara på skriftligen. Detta var utformat för att få ut samma information som den direkta observationen, men formulerat för att passa denna metod.

2.6.1 Hur designades detta test?

- Varför dessa frågor?

Till skillnad från den direkta observationen så fanns det mindre möjlighet att direkt få hjälp från programmeraren och testet fick därmed formuleras och läggas upp annorlunda jämfört med den direkta observationen. Det vill säga genom att simplificera vissa frågeställningar för att minska risken för misstolkningar.

2.7 Källkritik

För detta examensarbete har det krävts en stor mängd källor, inte bara för tillförlitlighetens skull utan för omfattningen av allt som behövde efterforskas. Programmeraren har gjort sitt bästa för att använda så många tillförlitliga och oberoende källor som möjligt för att kontrollera mot de primära källorna. Några av dessa områden finns det väldigt lite information om och att hitta flera källor från oberoende objektiva källor visade ibland inte möjlig. Det är värt att notera att en del av den dokumentation som använts tillhandahölls av större kommersiella företag såsom Oracle. Organisationer, företag och andra kan ibland vinkla sin information för deras egna intressen, vilket försämrade tillförlitlighet. Detta är något som har hållits i åtanke under läsningen.

När det gäller trovärdigheten av alla källor så har all information minst försökts dubbelkollats med en självständig sekundär informationskälla så att informationen kan backas upp och det inte bara är författarens åsikter som reflekterats i texten, boken eller videon.

En annan iakttagelse som kan medföra viss oro är att de allra flesta av källor i detta examensarbete är hämtade från internetkällor. Men källorna till dessa, om de är nämnda, har kollats upp på de sektioner som informationen har använts. Ett exempel på detta är från Ana Carvalhos rapport^[6] när hon har skrivit om användarkuben^[7] så har programmeraren gått till denna källa, som är från Jakob Nielsens bok ”Usability Engineering” som är tryckt av Academic Press som är en tillförlitlig källa. Vid möjlighet så har det även kollats på stycket i given bok för att se till att det som Ana Carvalho har skrivit stämmer överens med det som finns i boken. Motsvarande har gjorts på varje källa då det fanns möjlighet att göra detta.

För de källor där detta inte var möjligt så har den primära källans tillförlitlighet först övervägts och jämförts med oberoende andra källor för att se till att det är samma grundläggande information om ämnet.

3 Användbarhetstester

Examensarbetet fokuserade på att jämföra användbarhetstester på saker som har programmerats för att se till att de faktiskt hade den effekt som efterfrågades vid skapandet av dessa.

3.1 Vad är användbarhetstestning?

Användbarhetstestning är en serie tester som utförs på ett program eller i detta fall en hemsida för att testa om de idéer som programmeraren eller företaget har kommit på faktiskt fungerar i verkligheten. Kort beskrivet så ger man en person möjligheten att få ett scenario om vad denna skall utföra i givet program eller på given hemsida. Man ser då till att beskrivningen av scenariot är ej går att misstolka så att det enda som skulle kunna hindra lösningen av givet scenario eller fördröja denna kan vara ett problem i designen. Denna information bearbetas och leder ibland till förslag om precis vad som ändras, vilket gör det relativt lätt att lista ut hur detta skall lösas. Alternativt får programmeraren tillbaka information som inte kunde användas från testgruppen som säger att denna design inte fungerar, kom på en ny, utan att ge konkret feedback på var problemen uppstår eller vad som egentligen behöver ändras. Detta leder till mycket frustration hos programmeraren och produkten blir inte så bra som den hade kunnat bli. Detta är precis varför detta examensarbete utfördes.

3.2 Olika definitioner

Den grad i vilken specifika användare kan använda en produkt för att uppnå ett specifikt mål på ett ändamålsenligt, effektivt och för användaren tillfredsställande sätt i ett givet sammanhang^[12]

Ovan är hur det Europeiska Användbarhets Centrat har definierat användbarhet.

Det finns många olika definitioner för vad användbarhet i detta sammanhang. Det finns en standard, men som med många standarder kan denna tolkas på många olika sätt. Nedan kommer några olika förslag från forskare som har ägnat mycket tid att arbeta med användbarhetstestande.

Shackel (1990) hänvisar till fyra olika delar som är grunden för användbarhetstestning.

1. Lärbarhet (hur lätt det är att lära sig)
2. Helhet
3. Flexibilitet
4. Attityd

Rubin (2011) och Booth (1989) håller med varandra om att användbarhet innehåller ett eller flera av de fyra faktorer som anges nedan.

1. Användbarhet
2. Effektivitet (användarvänlighet)
3. Lätt att lära
4. Attityd

Vissa punkter delas bland de tolkningar som nämndes ovan och det är de gemensamma punkterna som programmeraren utgår ifrån i sin rapport. Baserat på dessa definitioner om användbarhet kan vi dra slutsatsen att det finns två breda områden för att samla in relevant information:

- System och användarprestanda (effektivitet, lätthet att lära och enkelhet att använda)
- Nöjda användare.

3.3 Bedömning av metoder

De metoder som har bedömts i detta examensarbete är direkt observation, **kapitel 9.1 Bilaga 1: Direkt Observation**, och Undersökningar, **kapitel 9.2 Bilaga 2: Undersökningar**. Metoderna som har testats bedömdes på en skala av 1 till 10 utifrån följande kriterier.

Skalan går från 1 till 10 poäng och skall tolkas på följande sätt.

- Svarsprocent.

Hur stor andel av testerna som får in svar jämföras mot hur många tester som utfördes eller efterfrågades. Alla testmetoder kan inte garantera att man får in svar av varje person och kräver därmed mer tid. Eventuellt ges inte hela bilden, då man kanske missar en specifik målgrupp som kanske är den mest tekniskt utmanade gruppen.

1 på skalan betyder att 10% av de tillfrågade gav svar och 10 betyder att 100% av de tillfrågade gav svar.

- Konkret feedback.

Är informationen som testet resulterar i något som kan användas direkt för en programmerare eller behöver det formuleras om eller tolkas om för att ge en förståelse av problemet alls?

1 på skalan betyder information som ej kunde användas av programmeraren och 10 på skalan betyder att testaren beskriver precis vad problemet är och har gett förslag på en bättre lösning.

- Tid lagd på förklaring.

1 på skalan betyder att det krävdes väldigt mycket tid för att förbereda testerna i allt från struktur om av hur detta skulle se ut till genomförandet. 10 på skalan betyder att det krävdes väldigt lite tid.

- Korrekt tolkning av uppgifter.

1 på skalan betyder att personen har gjort en tolkning av testerna (bilagorna) men att detta inte har gett ut informationen som programmeraren hoppades på. 10 på skalan betyder att testmetoden inte kräver mycket förklaring och att testet är väl designat.

3.4 Vad är skillnaden mellan tekniker och metoder?

En teknik är ett alternativ eller hjälpmedel för att utföra en viss metod av användbarhetstester. Till exempel om metoden som valts är observation, så kan man ta hjälp av inspelning eller mjukvarulogging – eller båda tillsammans. Man skulle kunna säga att en metod är en samling av tekniker som kan används tillsammans. Samtidigt så är valet av metoden även ett

val av vilka resurser som finns tillgängliga. Till exempel om det finns en expert man kan använda sig av, eller om man har möjligheten att hitta många personer inom en målgrupp för en undersökning i form av ett frågeformulär.

Nedan, i **tabell 1**, visas hur metoder och tekniker kan kombineras.

Metod	Teknik
Expertutvärdering	Walk-through Papper och penna
Observation	Direkt observation Inspelning Mjukvaruloggning Tänka högt
Undersökningar	Intervju Frågeformulär
Experimentell utvärdering	Mjukvaruloggning Frågeformulär Intervju

Tabell 1 – Metoder och tekniker för att samla in data ^[11]

3.5 Metoder

Enligt Ana Carvalhos rapport^[6] som bland annat är baserad på Preece (1993), finns det fyra metoder för utvärdering av användbarhet:

1. Expertutvärdering
2. Observationer
3. Undersökningsutvärdering
4. Experimentell utvärdering

De olika metoderna innebär olika typer av bedömare, olika antal användare, och olika typer av uppgifter som ska samlas in. Hon fortsätter att beskriva dessa på följande sätt i underrubrikerna.

3.5.1 Expertutvärdering

Expertutvärdering, även känd som heuristisk utvärdering, utförs vanligtvis av erfarna personer inom design och mänskliga faktorer som ombeds att beskriva de potentiella problem som de förutser för mindre erfarna användare. Dessa experter kommer ofta med lösningar på de problemen som de hittar.

3.5.2 Observation

Observationsutvärdering innebär att samla in data som ger information om vad användarna gör när de interagerar med programvaran. Flera olika datainsamlingstekniker kan användas, som kommer att beskrivas under **kapitel 3.6, Tekniker**. Det finns två breda kategorier av uppgifter som kan erhållas. Den första är hur användarna tacklat uppgifterna som utformats, där de största svårigheterna ligger och vad som kan göras. Och den andra är prestationsmått som mäter frekvensen av korrekt genomförda uppgifter, snabbhet (effektivitet) och frekvens

av deltagarnas fel eller missförstånd efter beskrivning. Det kan vara svårt att förstå om det är det sistnämnda som är problemet om testaren ej är helt ärlig mot personen som observerar. Mer om detta specifikt nämns i **kapitel 3.6.1, Direkt observation**.

3.5.3 Undersökningar

Undersökningar används för att veta användarnas åsikter eller att förstå deras önskemål om en befintlig eller potentiell produkt med hjälp av intervjuer eller enkäter som genomförs antingen personligen eller via mail.

3.5.4 Experimentell utvärdering

I en experimentell utvärdering så kommer en utvärderare kunna manipulera ett antal faktorer som är associerade med gränssnittet och studera deras effekt på användarnas prestanda. Det är extremt viktigt att planera allt mycket noga när denna metod används.

3.6 Tekniker

Det finns flera tekniker såsom direkt observation, videoinspelning, mjukvarulogging, interaktiv observation, verbala protokoll, intervjuer och enkäter som kombineras inom de olika metoderna.

3.6.1 Direkt observation

Direkt observation innebär att observera användarna under tiden de utför uppgifterna, utvärderaren antecknar om användarprestanda och tidssekvenser av handlingar. Dock kan denna teknik påverka användarnas prestanda, eftersom de kan reagera som de tror utvärderaren skulle förvänta sig istället för att reagera som de hade gjort om de satt själva och försökte lista ut det. Denna situation kallas ”The Hawthorne Effect”. För att undvika en sådan situation så kan man använda sig av videoinspelning av skärmen. Inspelelingen därefter spelas upp och användarnas beteende och problem analyseras. Här rekommenderas också att kombinera med att be testpersonen tänka högt och spela in ljudet samtidigt för att få ut så mycket information som möjligt.

Ibland får användare och utvärderare samarbeta om tolkningen av protokollet eller med andra ord uppgifterna som skall utföras. Så det är viktigt att testpersonen fortfarande är tillgänglig men håller sig på så pass bra avstånd att det inte påverkar testpersonens utförande av uppgifterna. Detta för att förtydliga uppgifterna så att informationen som vill samlas in, faktiskt är det som samlas in. Detta kan annars leda till att testmetoden i sig döms efter ett dåligt utformat eller formulerat test. Genom att testpersonen tänker högt gör användarna det möjligt för utvärderaren att förstå hur de ser på produkten och det gör det lättare att identifiera användarnas missuppfattningar, vilket är det absolut viktigaste med ett användartest.

Denna teknik, om utförd korrekt, kan ge ut väldigt mycket information som inte kräver mycket tolkning från utvärderare för att lista ut var problemen ligger. Detta leder självklart till tydligare information när listan med förslag till ändringar kommer till programmeraren.

Ett av problemen är att användarna ofta har svårt att uttrycka sina tankar högt när denna försöker lösa ett svårt problem. En lösning till detta är att para ihop två personer som tillsammans ska försöka lösa problemet. Men då uppstår ytterligare ett problem, att det sämre simulerar en verklig situation då personerna kan få mer hjälp än de normalt hade fått om de satt själva. Detta knyter tillbaks till att ha utvärderaren i närheten att kunna prata med för att förtydliga uppgifter och denna kan be personen förklara hur den tänker ifall en svår situation

uppstår utan att ge för mycket eller någon information alls till testpersonen. Det är inte garanterat att alla tester kan slutföras ifall designen är dålig på produkten. Om detta problem uppstår så bör utvärderaren snabbt gå in och ändra sin strategi och avsluta testperioden med en kort intervju då man frågar testpersonen om förslag på hur denna hade tänkt att man kunde designat lösningen på givet problem.

3.6.2 Mjukvarulogging

En annan användbar teknik är mjukvarulogging. Genom denna metod registreras interaktionen mellan användaren och programvaran. Data samlas diskret utan att påverka användarens sätt att arbeta. Den består vanligen av en tidsstämplad logg över användarens input och mjukvarusvar. Det är möjligt att rekonstruera exakt vad användaren gjorde och den tid som används för varje funktion, och även för att analysera frekvensen för användning av vissa funktioner. Personligen rekommenderar programmeraren någon form av videospelningsprogram som kan fånga precis allt som personen gör, även ljud och ansiktsuttryck eller var på hemsidan personen fäster blicken på hemsidan. Detta kan självklart tas väldigt långt men man bör sätta en gräns för att det inte ska ta upp för mycket tid för informationen det ger ut för given produkt.

3.6.3 Intervju

Intervjun är ett sätt att samla in uppgifter i en undersökning. Intervjuer kan vara strukturerade (sekvens av förutbestämda frågor) eller flexibla (det har vissa ämnen och därefter tas frågor fram efter svaren som ges). Det andra sättet att samla in data i en undersökning är genom enkäter. Det finns två typer av frågor: öppna frågor (testaren ger hans/hennes eget svar) och slutna frågor (testaren väljer svaret från ett urval av alternativa svar). I detta projekt valdes den första av dessa två då det passade bäst för de relativt enkla uppgifterna som skulle utföras av testpersonerna, och vissa saker kan vara svåra att förutse när det gäller svarsalternativ som väljs.

3.7 Utvärderare

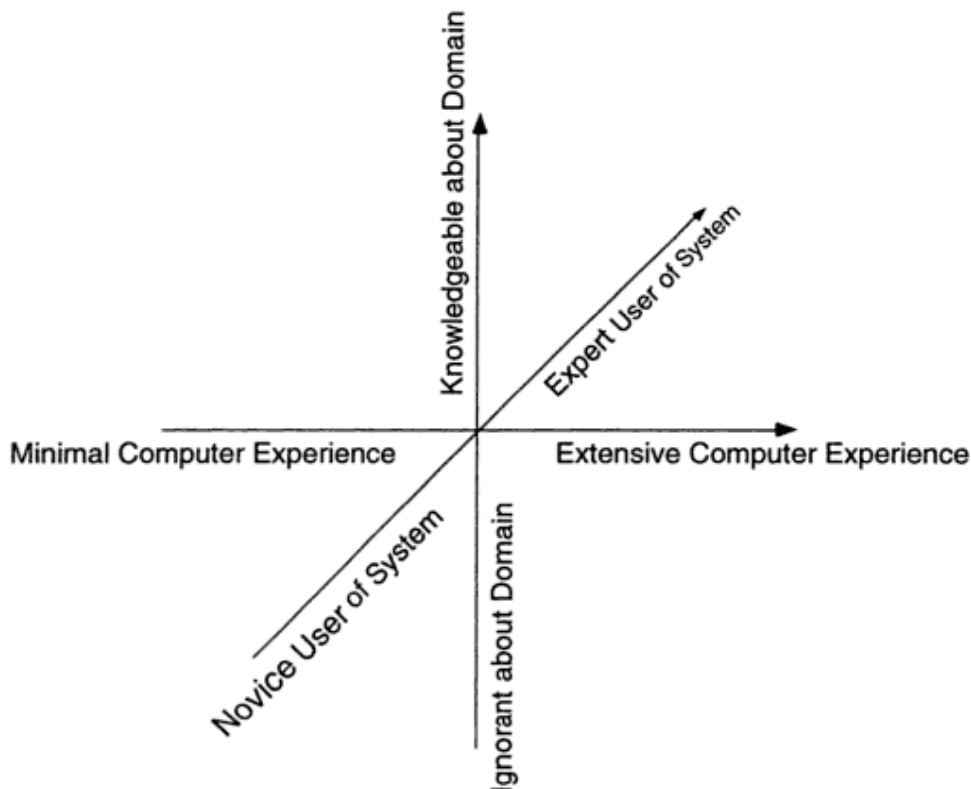
De flesta forskare är överens om att det finns två typer av utvärderare: experter och användare (Rubin, 2011). Expertutvärderingar har en bra överblick av programvaran enligt accepterade användbarhetsprinciper. Det finns också en sorts "dubbel" specialist, det vill säga en expert inom användbarhet som också är expert på den speciella teknik som används av programvaran, i det här fallet ASP.net. Programmeraren i detta projekt ligger i den andra kategorin eftersom allt från utveckling och testning hanteras.

Observation, undersökningar och experimentell utvärdering innebär förekomsten av användare eller som de benämns i denna rapport, testpersoner. Deltagarna måste vara representativa för målgruppen då det gäller att bedöma i vilken grad en produkt uppfyller specifika användbarhetskriterier.

En viktig fråga för användbarhet är användarnas individuella egenskaper och skillnader. Det finns tre viktiga faktorer där användarnas erfarenheter skiljer.

- Erfarenhet av systemet.
- Erfarenhet av datorer i allmänhet.
- Erfarenhet inom uppgiftens problemområde.

Nedan, i **figur 3**, visas användarkuben som Nielsen beskriver i sin bok^[7].



Figur 3: Användarkuben^[7]

Dessa tre faktorer beskrivs som "användarkuben" (Nielsen, 1993). Det gäller att ha testpersoner i så många olika kombinationer som möjligt av dessa tre för att få en så pass komplett bild som möjligt av ändringar som behöver ske. Detta är inte alltid så väldigt enkelt att lösa och man får göra en uppskattning efter de testpersoner man har haft tillgång till.

Testerna bör även ta upp användarens prestation och tillfredsställelse. Prestationen motsvarar åtgärder av deltagarens beteende, med fokus på aspekter som "effektivitet av användning". Detta kan mätas i bland annat tid för att utföra en specifik uppgift, antal felaktiga menyal, antal besök på menyn, antal negativa kommentarer.

Det finns även något som kallas för preferensdata som utgör åtgärder av deltagarens åsikt eller tankeprocess. Rubin (2011) påpekar några aspekter för att mäta nytta av produkten. Hur väl produkten levde upp till förväntningarna, användarvänlighet, hur svår produkten var att lära sig, tillgänglighet, hjälprutor, hur grafiskt tydligt det var.

Det finns flera tester för att utvärdera kundnöjdhet. WAMMI^[13] (Website Analysis and Measurement Inventory) är den mest intressanta för detta projekt för att den är designad för att bedöma hemsidor. WAMMI beskrivs närmare i **kapitel 3.7.1, WAMMI**, då mycket av användartestandet i projektet är baserat på denna metod.

Det är viktigt att kunna skapa ett nytt instrument som är specifikt för testerna som är utsatta i projektplanen. Det är viktigt att ta hänsyn till att de uppgifter som skall samlas in bör baseras på de angivna problemanalyser och testmål som presenteras i projektet.

3.7.1 WAMMI

Enligt WAMMIs officiella hemsida^[13] beskrivs det som det bästa verktyget för bedömning av användarupplevelsen som dina besökare faktiskt får på din hemsida.

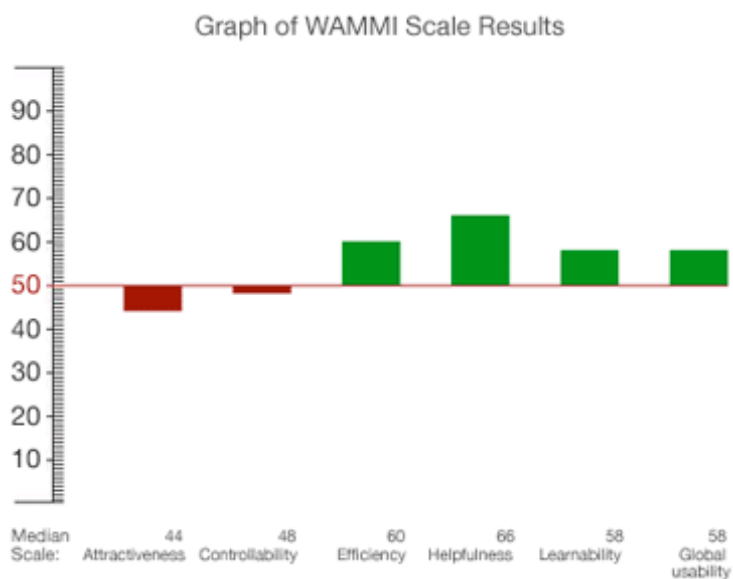
WAMMI är uppbyggt av 20 specifikt utvalda frågor som experter har forskat fram gällandes att få ut så bra information som möjligt om en hemsidas användarvänlighet enligt användarna eller testarna. Frågorna tillåter besökarna eller testarna att säga vad de verkligen tycker, istället för att ha idéer som kastas på dem som en del testmetoder gör. Varje fråga representerar en viktig del av och alla är nödvändiga för att fånga hela skalan av användarupplevelse.

När de forskade fram WAMMI såg de att det finns olika viktiga kategorier att dela upp frågorna i. Kategorierna de delade upp användbarhet i var attraktivitet, styrbarhet, effektivitet, hjälpsamhet och lärbarhet. Dessa kan direkt översättas till hur bra sidan är designad med fokus på att användarna enkelt kan hitta var de vill göra uppgiften de hade tänkt. Och även hur tydligt det är man utför sagt uppgift och om det krävs hjälprutor som beskriver tydligt precis vad som ska göras utan att misstolkas.

Tjänsten kretsar kring ett frågeformulär och en internationell databas. En rapport genereras i slutet av utvärderingsperioden. Besökarens upplevelse mäts genom att be besökaren till webbplatsen får chansen att jämföra sina förväntningar mot vad de faktiskt hitta på webbplatsen. Genom att ställa ett antal ytterligare frågor, lite mer detaljerad information, om vilka typer av besökare (ålder, kön, datorvana), varför de besöker hemsidan och hur de tror att det kan förbättras.

Besökarna får fylla i enkäten och skicka in sina svar. Hela processen tar bara några minuter, och svarsfrekvensen är normalt runt 10-20%. När tillräckligt många användare har svarat (någonstans mellan 40 och 200 användare rekommenderas) tas länken ner.

Nedan, i **figur 4**, visas ett exempel på hur resultatet ur ett WAMMI-test kan se ut. Detta kallas en webbplatsprofil.



Figur 4: Webbplatsprofil^[13]

Den viktigaste delen av rapporten är din webbplatsprofil, som innehåller de sex kategorier som togs upp tidigare. En genomsnittlig poäng är 50 och en perfekt poäng är 100.

3.7.2 Hur valdes testpersonerna?

För att hitta personer som kunde ta upp så många olika delar av användarkuben som möjligt så fick testpersonerna väljas ut väldigt varsamt. Det fanns en databas med användare på hemsidan redan som kunde användas och även personligt valda personer från programmeraren. Detta kunde vara nära vänner eller en distant kontakt som uppfyllde en viss målgrupp.

3.7.3 Exempel på användbar och ej användbar feedback

Det är svårt att generalisera vad som räknas som användbar information och vad som ej räknas som användbar information. Av denna anledning tas nedan upp ett exempel på kommentarer, muntligt eller skriftligt, för båda.

Jag insåg ej att man skulle trycka på en knapp för att skriva ut.

Detta är ett perfekt exempel på användbar kommentar från testerna. Detta indikerade till exempel att antingen så borde det stå med i beskrivningen på sidan att man skulle trycka på utskriftsknappen när man ställt in rätt parametrar på sidan. Eller så borde knappen göras mer synlig i form av att byta färg, lägga mer avstånd från den och det andra på sidan.

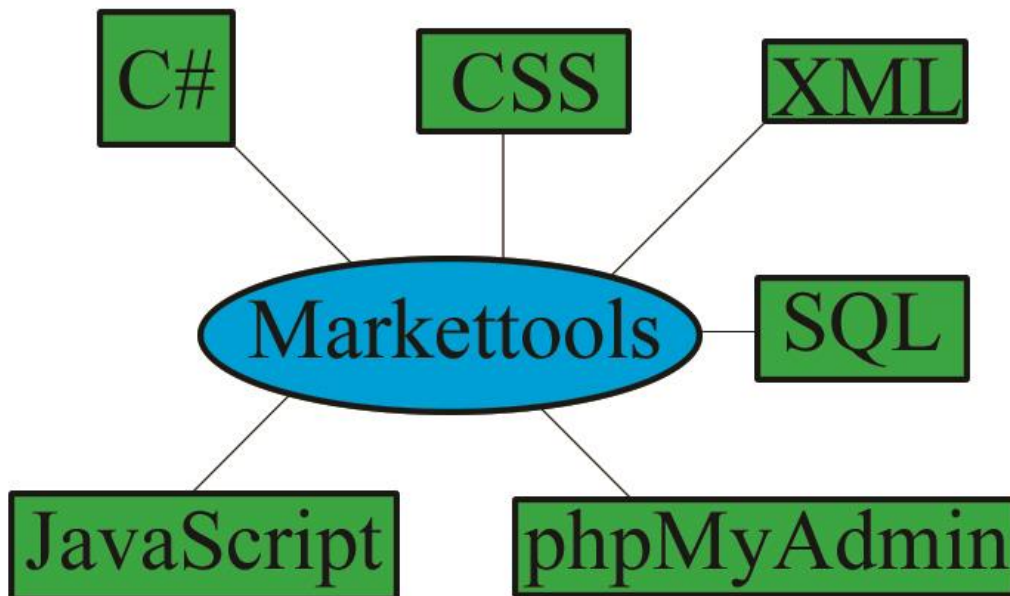
Jag tycker inte om färgen på sidan.

Detta är ett exempel som är relativt oanvändbart då detta ämne är väldigt subjektivt. Dock så är den här kommentaren bara värdelös om denna är den enda av samma slag. Om det finns ett flertal liknande kommentarer blir denna genas mer relevant.

4 Teknisk Bakgrund

För att kunna förstå hur www.markettools.se är uppbyggd och hur de olika delarna fungerar tillsammans krävs en kort genomgång av ASP.net^[16] och alla dess delar.

Nedan, i **figur 5**, visas de olika komponenterna som hemsidan består av för att ge en förenklad syn över vad det innebär.



Figur 5: De olika komponenterna på hemsidan

4.1 Vad är ASP.net?

ASP.net är en plattform designad för att utveckla dynamiska hemsidor. Det utvecklades även med syfte att ha ett enkelt sätt att använda flera olika programmeringsspråk tillsammans, då de uppfyller olika delar av en bra hemsida. En bra hemsida i det här fallet är en bra balans av vad som ska köras på servern, på klientens egen dator eller mobilenhet och vad som ska döljas helt från användarna av bland annat säkerhets och sekretesskäl.

ASP.net bygger på Common Language Runtime* (CLR), vilket är vad som gör att programmerare kan skriva ASP.net-sidor med hjälp av vilket språk som helst som har stöd. De språk som använts i projektet kommer kort att beskrivas nedanstående och ge en bild över hur de integrerar med varandra i de olika delarna av hemsidan.

4.1.1 C#

C# är ett objektorienterat* programmeringsspråk som till exempel Java*, C++*, php eller Python*. Det utvecklades av Microsoft Corporation i samband med skapandet av ASP.net och är baserat på C++. Till stor del liknar det Java, som under många år har varit ett av det mest använda programmeringsspråket. Anledningen till att det används så mycket är att det är ett väldigt enkelt språk att sätta sig in i och används ofta som det första språket man lär sig i utbildningar om programmering för att enkelt sätta sig in i hur tankesättet vid programmering generellt är.

En viktig sak att nämna är att C# kräver ett CLR* för att kunna köras. CLR tar hand om minnesallokering, optimerar körning, skräphanterar, tar hand om exceptions* och mycket mer.

4.1.2 Databaser och SQL

En databas är precis vad det låter som, en samlad plats där all data (information) finns tillgänglig. I detta fall är informationen inloggningsuppgifter, olika auktionsobjekt som är aktiva för tillfället eller köp som har avslutats för att ge ett par exempel. Men det som skiljer en databas från ett dokument där all information finns är hur det är relaterat till varandra för att ta upp så lite plats som möjligt och göra sökningar effektiva. Vi kan ha en person som får ett speciellt ID-nummer när han skapar sin inloggning på hemsidan, detta ID-nummer är kopplat till ett bud han lägger på en auktion.

SQL (Structured Query Language) är ett programmeringsspråk speciellt utformat för att göra sökningar i eller bygga en databas. SQL inkluderar insättning, hämtning, uppdatering och borttagning av data och mycket mer. SQL är överlägset det mest använda frågespråket idag när det handlar om databaser.

4.1.3 Javascript

Javascript är ett programmeringsspråk som används för att göra hemsidor mer interaktiva*. Det är oftast använt som en del av webbläsare, vars implementationer tillåter skript* för att interagera med användaren, styra webbläsaren, kommunicera asynkront* och ändra innehållet på sidan som visas. Trots sitt namn så har Javascript och Java väldigt lite med varandra att göra. Syntaxen för Javascript kommer istället från språket C*.

Fördelen med Javascript är att det inte konstant kräver en massa nerladdningar från hemsidan, utan körs på användarens dator (förutsatt att webbläsaren stödjer Javascript). Support för Javascript finns inbyggt i alla stora webbläsare* och är igång automatiskt tills man själv stänger av det.

4.1.4 CSS och HTML

CSS, Cascading Style Sheets, är ett så kallat "stilmallsspråk". Detta innebär att språket används för att beskriva utseendet och formateringen av en hemsida som är skriven i ett annat språk så som Javascript, HTML* eller XHTML*. CSS spelar en väldigt stor roll för specifikation av webben, och nästan alla hemsidor använder stilmallar i CSS för att beskriva sin hemsida.

HTML är själva strukturen på hemsidan, till exempel hur olika delar ser ut, var knappar ska sitta, vilken text som står med. Olika delar av hemsidan delas upp av olika "taggar". Ett exempel på en "tag" är <H1>, vilket säger att det är en rubrik. Samtidigt så bestämmer CSS hur texten är formaterad (typsnitt, fetstil, kursiv, storlek...) eller marginalen mellan startpunkten tills var texten ska placeras.

CSS är främst avsedd för att möjliggöra separation av dokumentets innehåll från dokumentets utseende. Denna separation kan förbättra innehållets tillgänglighet, ge mer flexibilitet och kontroll för att ha ett gemensamt tema på hela hemsidan så att det känns mer professionellt. För att gå tillbaks till exemplet med <H1> som nämndes ovan, så skriver man i CSS-dokumentet att all text som är markerad med en <H1>-tag skall vara i en speciell storlek. Detta betyder att alla ställen då denna "tag" används kommer att ha exakt samma storlek. Så istället för att skriva det varje gång man har en "tag" (risk att skriva fel, mer kod...) så finns

allt samlat på ett ställe och ändringar i design blir mycket enklare att göra för att få ett enhetligt utseende.

4.2 php

PHP (Hypertext Preprocessor) är ett skriptspråk avsett för webbutveckling. PHP-kod kan enkelt blandas med HTML. PHP-kod är vanligtvis behandlat av en PHP-tolk, som genomförs med ett CGI* som gör det körbart. Efter PHP-koden tolkas och exekveras så skickar webbservern sedan ut informationen till användaren.

Det som skiljer PHP från Javascript är att koden som utförs faktiskt körs på servern och inte användarens dator. Användaren får bara ut resultatet av ditt skript och har ingen möjlighet att se hur ditt skript är utformat. Detta är speciellt viktigt i fall då det gäller inloggning och liknande delar av en hemsida för hemsidan och databasens säkerhet och sekretess.

4.2.1 phpMyAdmin

phpMyAdmin är en gratis programvara skriven i PHP, avsedd för att hantera administrationen av MySQL över webben. Den används bland annat för att hantera databaser, tabeller, kolumner, relationer, index, användare och behörigheter. Allt detta kan utföras via användargränssnittet, medan användarna av hemsidan fortfarande inte har direkt tillgång till databasen. Detta är ytterligare ett säkerhetslager för att skydda hemsidan från att bli hackad*.

4.3 Grundsystemet

Hemsidan är utvecklad så att den ska vara så generisk som möjligt. I databasen ligger information som berör olika kunders version av hemsidan. Detta för att exakt samma kod ska kunna användas för olika kunder. Hemsidan anpassas istället genom att lägga in olika information, exempelvis textsträngar i databasen.

Utan databas får du den generiska hemsidan. Annars läses information för en särskild kund in från databasen.

Vi hade bara en kund, varpå jag tror att det är hårdkodat i master-filen. Det finns dock en lösning där som kollar på URL:en för att bestämma vilken kund som ska läsas in.

Det är inte jag som konstruerat systemet såhär. Tanken är att det ska vara lätt att sälja hemsidan till så många som möjligt utan att behöva göra något jobb, men det gör att det är svårare att anpassa enskilda sidor.

- Mikael Karlsson, tidigare programmerare i projektet.

Det som Mikael syftade på var de olika logotypbilderna, standardtexter för förstasidan, avslutade auktioner och information om försäljaren bland annat som fanns i databasen och inte hårdkodat till varje sida. För att lägga till ytterligare funktioner för specifika kunder får man självklart i deras kopia på servern gå in och lägga till koden. Så om ytterligare kunder vill ha samma funktioner så gäller det att hålla koll på var denna har lagts till och om det bara är att slänga in koden eller ej.

4.4 Uppgiftsbeskrivning

Här kommer en liten sammanfattning av vårt projekt som vi skulle vilja ha hjälp med som ett examensarbete till dig.

Vi har byggt ett Internetbaserat auktionssystem som heter MarketTools utvecklat i C# och ASP.net. Systemet behöver uppdateras och byggas ut så vår första riktiga kund kan använda systemet fullt ut.

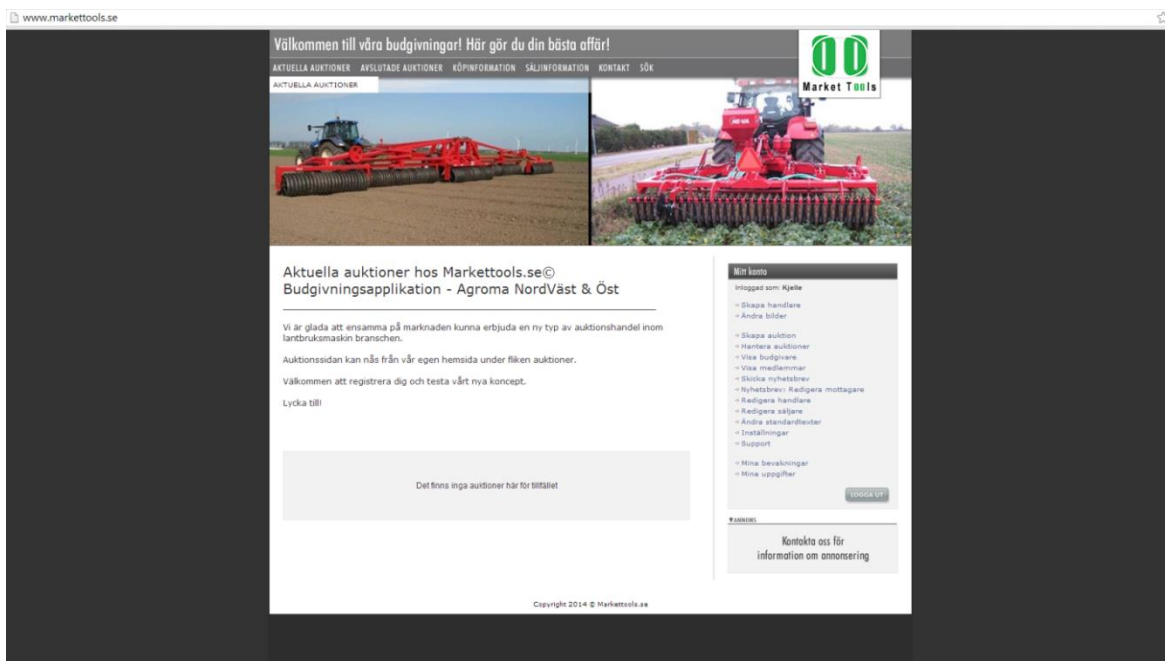
Exempel på funktions som behöver läggas till eller förändras:

- Utskriftsfunktion - läggas till
- Sökfunktion - förbättras
- Ändra layout på förstasidan
- Annonsfunktionen göras lättare att administrera – läggas till
- Dokumentera systemet – läggas till

Med vänliga hälsningar
Kjell Ivarsson, VD

Det var en tydlig lista av Kjell med arbetsuppgifter, vilket var bra, eftersom det alltid fanns något att göra och examensarbetet blev mer flexibelt. Om man skall börja nerifrån så fanns det dokumentation för vissa delar av koden gömt i en fil. Annonsfunktionen uppdaterades, vilket bara var en snabb uppdatering i databasen. Layout på förstasidan var inget som gjordes under projektets gång på grund av tidsbrist och nämns senare i **Framtida utvecklingsmöjligheter, kapitel 6.4**. Sökfunktionen hann inte heller ändras på grund av tid. Projektet fokuserade på att göra en så bra utskriftsfunktion som möjligt och det lades även till en ytterligare för fakturaunderlag som var kopplat till denna.

Nedan, i **figur 6**, visas en överblick av hemsidans startsida för att ge en idé om vad examensarbetet bygger på.



Figur 6: www.markettools.se [15]

4.5 Första anblick

För att få en inblick i det som redan fanns tillgängligt på hemsidan så gav Kjell Ivarsson, VD på Diflex AB, en snabb genomgång om hemsidan och vad man kunde göra. Auktionssystemet var klart, det mesta på hemsidan var klart. Utöver ett par funktioner och en total avsaknad av dokumentation från tidigare programmerare så fungerar hemsidan.

Efter denna genomgång gavs tillgång till koden, 13 000 rader kod, i sju programmeringsspråk som programmeraren ej tidigare använt och fick lägga lite mer än de första två veckor av projektet på att bara börja förstå dessa språk och hur de samverkade i koden. Eftersom det är exakt samma kod som ligger uppe på hemsidan, som fungerar, så tänktes det att den ska gå att köra lokalt på laptop utan några större problem. Så var ej fallet. Det fanns en del buggar och varningar i koden redan och en i princip total avsaknad av kommentarer i koden gjorde det svårt att lista ut vad som gick att ändra utan att totalt förstöra koden i övrigt.

4.6 Omdirigering till lokal databas för utveckling

Ett av de första problemen som påträffades var att sidan var designad generiskt för att kunna utgöra underlag för så många olika versioner som möjligt. Syftet med detta är att kunna sälja vidare auktionssystemet till mer än en kund utan att behöva göra några stora ändringar i ASP.net-koden. Istället sker de flesta av ändringarna i databasen som hämtar all information om den specifika sidan.

Tanken var först att ha en lokal kopia av allt som ligger uppe på servern för att enkelt kunna jobba med utvecklingen på en privat laptop utan tillgång till internet, detta för att förenkla användbarhetstesterna som nämns i **kapitel 3, Användbarhetstester**. Detta var något som tog oväntat lång tid att lösa då mjukvaruproblem uppstod vid installationer av rekommenderade program att använda. I detta fall var det Microsoft Visual Web Developer som är ett program utvecklat av Microsoft för att programmera i ASP.net. Koden nedan visar referensen till databasen i sidans kod som behövde hittas och ändras, men det var inte allt som behövde ändras.

Nedan, i **figur 7**, visas en bit av programmeringskoden i XML som kopplar databasen till hemsidan.

```
<connectionStrings>
  <add name="MySQLConnection" connectionString="DATA
    SOURCE=localhost;DATABASE=markettools;USER ID=user;PASSWORD=password;PORT=3306"/>
</connectionStrings>
```

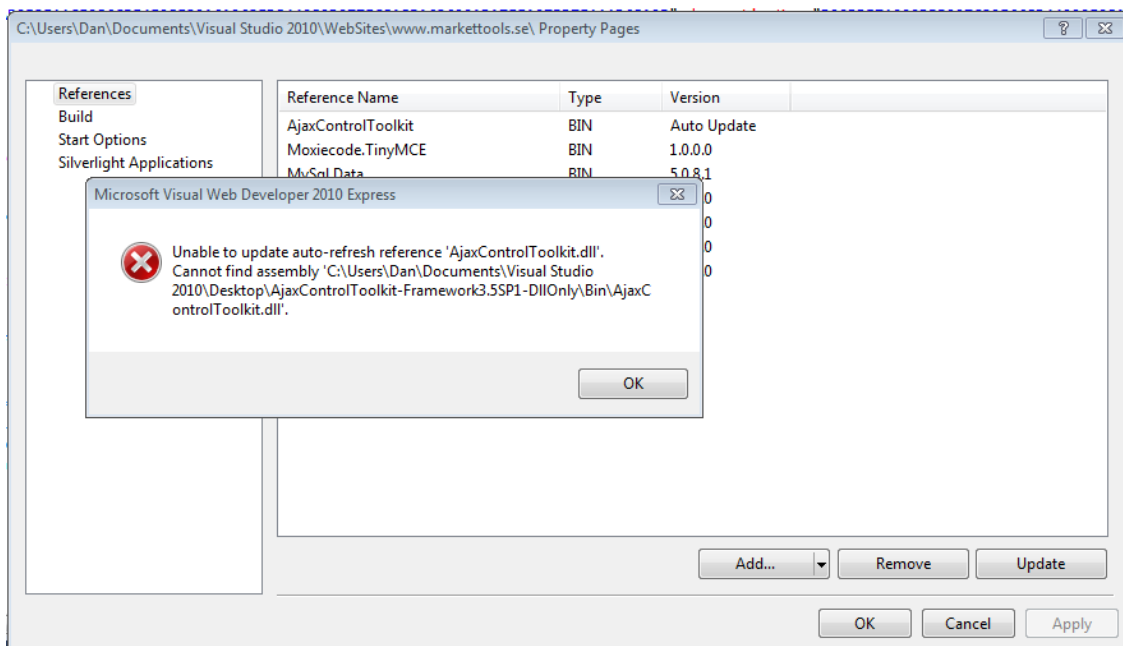
Figur 7: Kod för koppling till databas.

Tillägg i koden ändrades ej när programmet kördes lokalt. Detta kunde vara på grund av att databasen inte omdirigerats korrekt eller även att systemet var uppbyggt på ett annat sätt än vad som först var förstått. Den andra delen av lösningen var att faktiskt skapa en kopia av databasen i MySQL och få ut en referens därifrån. Efter detta problem var löst kunde utvecklingen äntligen påbörjas.

4.7 AjaxToolKit

Ajax (asynkron Javascript + XML) är en grupp inbördes relaterade tekniker för webbutveckling som används på klientsidan för att skapa asynkrona webbapplikationer. Med Ajax kan webbapplikationer skicka data till och hämta data från en server asynkront (i bakgrunden) utan att störa displayen av och beteendet hos existerande sida.

Nedan, i **figur 8**, visas ett exempel på ett problem som uppstod under överföring från server till lokal hantering.



Figur 8: Referens till AjaxToolKit fungerade ej efter kopiering.

Lösningen till problemet på bilden ovan var att gå in i projektet i Solution Explorer → Property Pages → References → AjaxToolKit → Update.
Om detta ej fungerade så behöve referensen ändras i koden manuellt.

5 Resultat

Beskrivning av bedömning finns i **kapitel 3.3, Bedömning av testmetoder**.

- Svarsprocent.

Hur stor andel av testerna som får in svar jämföras mot hur många tester som utfördes eller efterfrågades. Alla testmetoder kan inte garantera att man får in svar av varje person och kräver därmed mer tid. Eventuellt ges inte hela bilden, då man kanske missar en specifik målgrupp som kanske är den mest tekniskt utmanade gruppen.

1 på skalan betyder att 10% av de tillfrågade gav svar och 10 betyder att 100% av de tillfrågade gav svar.

- Konkret feedback.

Är informationen som testet resulterar i något som kan användas direkt för en programmerare eller behöver det formuleras om eller tolkas om för att ge en förståelse av problemet alls?

1 på skalan betyder information som ej kunde användas av programmeraren och 10 på skalan betyder att testaren beskriver precis vad problemet är och har gett förslag på en bättre lösning.

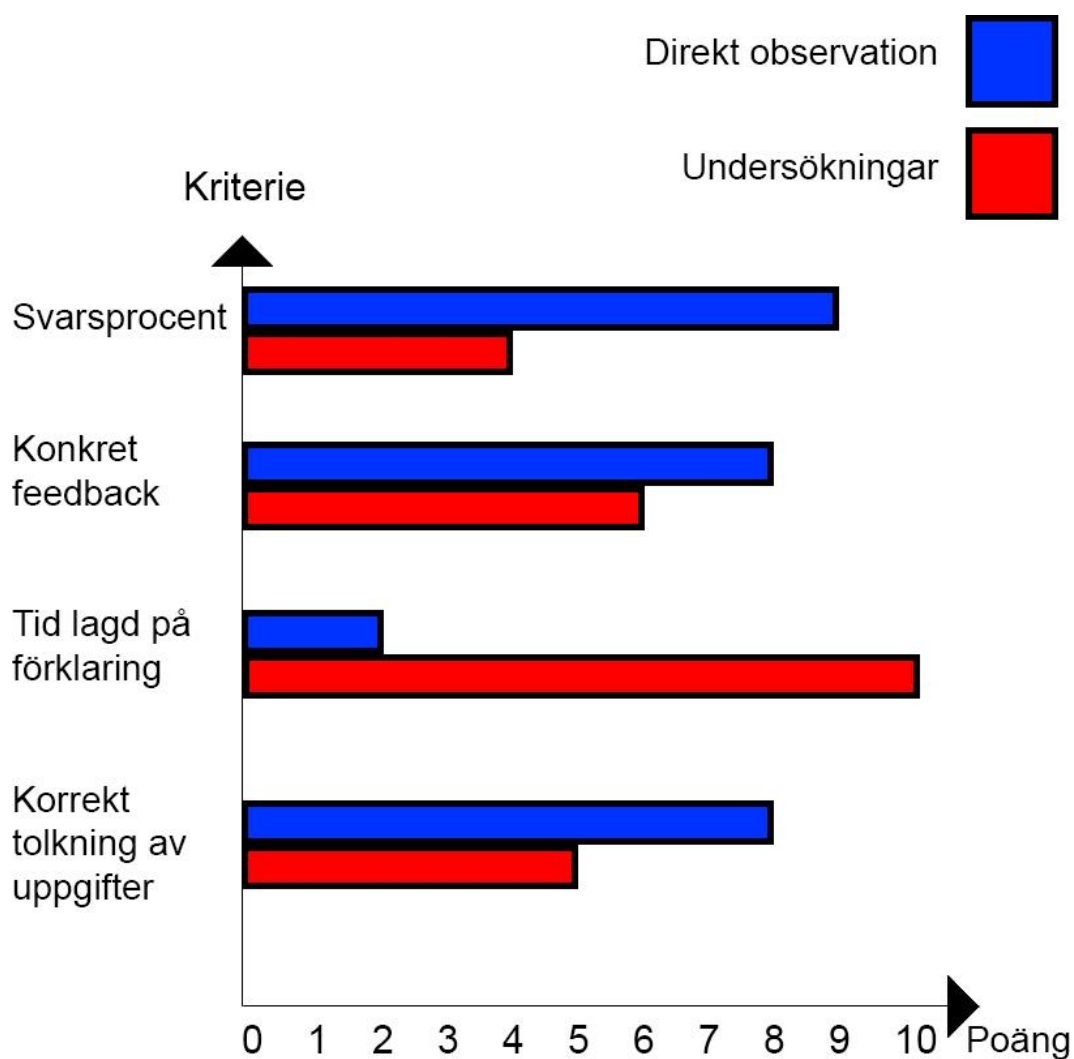
- Tid lagd på förklaring.

1 på skalan betyder att det krävdes väldigt mycket tid för att förbereda testerna i allt från struktur om av hur detta skulle se ut till genomförandet. 10 på skalan betyder att det krävdes väldigt lite tid.

- Korrekt tolkning av uppgifter.

1 på skalan betyder att personen har gjort en tolkning av testerna (bilagorna) men att detta inte har gett ut informationen som programmeraren hoppades på. 10 på skalan betyder att testmetoden inte kräver mycket förklaring och att testet är väl designat.

Nedan, i **figur 9** visas resultatet i form av en graf för att enkelt överblicka styrkorna och svagheter mellan de olika metoderna.



Figur 9: Resultatgraf av testmetoderna

Det finns direkt några väldigt tydliga skillnader mellan de två olika metoderna.

Tiden som valdes för varje metod var 10 timmar som involverade allt från att formulera dokumenten till att tolka svaren och utföra observationerna.

Detta ledde till att det skedde fler undersökningar än direkta observationer eftersom dessa var väldigt mycket mer tidseffektiva, det var bara att skicka ut fler mail. Och sedan uppskatta ungefär hur lång tid det tog att få in och tolka alla resultat från dessa så att det läggs lika mycket tid på båda metoderna för att ge en så bra jämförelse mellan dessa som möjligt.

Den genomsnittliga tiden för den direkta observationen var 30 minuter och sedan tiden för att hitta personer som kunde genomföra testet. Detta ledda till att 18 personer intervjuades jämfört mot 100 personer som mailades ut till i undersökningsmetoden, varav 39 av dessa valde att ta del i undersökningen.

5.1 Direkt observation

Den direkta observationen hade en del fördelar och nackdelar som var följande.

5.1.1 Fördelar

- Det kunde i princip inte uppstå några större missförstånd om vad uppgifterna gick ut på.

Det fanns möjlighet att se till att personen verkligen försökte göra uppgiften och hur denna gjordes tack vare inspelning.

- Det gavs nästan alltid svar på frågorna som ställdes (9/10)

Den största fördelen med den direkta observationen i kombination med att ha bett testpersonen att tänka högt tillsammans med videoinspelning är att man kan gå tillbaks och se om testet så att man verkligen har fått en komplett bild av problemet.

- Den ger ut väldigt konkreta resultat (8/10)

Programmeraren kunde enkelt se var testpersonerna inte gör som det är tänkt eller att det tar längre tid för personen att göra en enkel uppgift.

5.1.2 Nackdelar

- Den direkta observationen var väldigt tidskrävande (2/10)

Det var klart redan innan testerna utfördes att den här sortens test ej kommer att kunna användas på lika många testpersoner som undersökningarna kan.

- Ett av problemen, som försökte lösas i beskrivningen till uppgifterna, var att personerna ibland glömde att tänka högt vilket gjorde att resultatet fick tas från videoinspelningen och tolkas utifrån vad personen gjorde.

Detta kunde avhjälpas genom påminnelser från programmeraren.

- Testpersonen bli stressad av att ha en person som satt bredvid och iakttog denna vilket ledde till vissa misstag som i normala fall inte hade uppstått enligt de flesta personer där detta uppstod.

5.2 Undersökningar

Undersökningarna hade en del fördelar och nackdelar som var följande.

5.2.1 Fördelar

- Undersökningsmetoden har en större potential till svar, med andra ord så kan man få in ett större antal svar (39) på samma tid jämfört med den direkta observationen (18).

Detta kan leda till en mer korrekt bild av den genomsnittliga personen som kommer att använda givet system.

- Överlägsen när det handlar om hur tidskrävande metoderna är (2/10 poäng)

Den absolut största skillnaden mellan dessa metoder är hur mycket tid som krävs för att skapa testerna och utföra testandet.

5.2.2 Nackdelar

- Dålig svarsprocent (4/10 poäng)

I verkligheten så är det väldigt svårt att få alla personer att svara och ge den sorts feedback som man är ute efter då personen sitter själv och tolkar uppgifter och frågor kanske på ett annat sätt än vad som är avsett.

- Sämre andel av konkret feedback (5/10 poäng)

Programmeraren kunde inte lika lätt, relativt till den direkta observationen, se var testpersonerna inte gör som det är tänkt eller att det tar längre tid för personen att göra en enkel uppgift.

6 Slutsats

- *Vilken metod av användartestning ger den mest konkreta feedbacken ur en programmerares perspektiv?*

Ur en programmerares perspektiv handlar allt om att konkret få ut informationen om var problemen ligger, vilket den direkta observationen ger väldigt mycket bättre än undersökningarna. Men med undersökningarna är det möjligt att man får in fler svar, samtidigt som risken finns att dessa inte ger konkreta svar på de frågor som programmeraren söker svar på. Detta kan resultera i information som inte kan användas vilket i sin tur inte medför att sidan blir mer användarvänlig eller till och med sämre än utgångsläget.

6.1 Direkt observation

Detta ger en klar bild av vad problemet är och sedan är det upp till programmeraren att lista ut en lösning utifrån denna oftast väldigt konkreta data som det finns tillgång till efter testet. Men samtidigt är denna metod extremt tidskrävande och behöver aktivt arbetas med under hela processen för att få ut tillräckligt mycket information för att veta vad som behöver ändras.

I kontrast till undersökningsmetoden där det fanns en del svar som lämnades blanka och tolkades i detta fall som att personen inte lyckats lösa sagd uppgift eller att frågan inte blivit förstådd helt vilket inte kunde göras något åt utan en uppföljning av svaret vilket inte ingick i beskrivning av detta projekt på grund av att det hade tagit längre tid än vad som var utsatt per metod.

6.2 Undersökningar

Denna metod har definitivt ingen garanti för att du kommer få svar från alla spektrum inom målgruppen för din hemsida. Detta kan leda till väldiga missuppfattningar i båda ändar av testet och kräva ändringar i systemet som egentligen inte behövs, alternativt missa saker som behövde ändras redan från början.

6.3 Sammanställning

Förutsatt att det finns tid i budget och projektplanen för att göra en ordentlig direkt observation på ett tillräckligt stort antal personer inom given målgrupp så kommer detta resultera i en mer användarvänlig hemsida i slutändan även om det betyder mer fysiskt och socialt arbete för personerna som är ansvariga för testandet. Med andra ord är det en mer effektiv metod, tidsmässigt, vilket är extremt viktigt i ett projekt med tydliga deadlines som är extremt vanligt i dagens industri. Att skicka ut ett hundratal mail, få svar på 25-40 % av dessa kräver också mer struktur för sammanställningen av resultatet och gör ofta att man inte kan göra ändringar efter att ha analyserat ett fåtal resultat. Utan det kräver att man analyserar alla resultaten och skriver kommentarer under hela tiden och sedan i slutändan sammanställer det och börjar göra ändringar efteråt. Denna metod ger helt enkelt inte samma bild av verkligheten som den direkta observationen eftersom den gör det lätt att missförstå vad som egentligen är problemet.

Samtidigt får man inte glömma att man får vara mycket mer selektiv när det gäller den direkta observationen och hitta personer i alla olika användargrupper då det utförs mycket färre tester under samma tidsperiod som motsvarande för undersökningarna. Detta kommer självklart att påverka resultatet av testerna som utförs i alla projekt av denna form och bör aldrig ignoreras. När dessa tester utförs måste testledarna, vare sig det är programmeraren själv eller ett team som separat jobbar med just det, måste de se till att en så bred del av målgruppen testas som möjligt. Med andra ord så är det inte garanterat att 200 personer inom målgruppen ger ett bättre och mer konkret resultat än att väldigt försiktigt välja ut 20 eller 30 personer som faktiskt täcker en större del av målgruppen.

6.4 Framtida utvecklingsmöjligheter

Specifikt för sidorna som har användartestats finns det alltid väldigt små, subtila saker som kan förbättras som positionen av knappar eller indrag till sidorna för texten. Men detta hade krävt många fler tester och det hade inte varit ekonomiskt värt det för företaget.

Men när det gäller framtida utvecklingsmöjligheter för hemsidan i sin helhet så har det inte utförts några tester alls till programmerarens vetskap. Hela sidans utseende hade eventuellt ändrats för att vara mer tillfredställande att använda för de flesta användarna. Designen för tillfället är väldigt simpel men börjar kanske få väldigt många funktioner och menyerna, som i och för sig bara rör personerna som håller igång hemsidan, hade kunnat omdesignas helt.

När det gäller testmetoderna så är programmeraren ganska så övertygad redan att den direkta observationen är det mest effektiva sättet att få fram konkret feedback om vad som behöver ändras, läggas till eller tas bort helt eftersom det finns en risk för att det blir för mycket som kan ta en användares intresse på skärmen. Det finns garanterat många fler metoder som hade kunnat testas i en mer utförlig studie som har valts bort i detta examensarbete. En hybrid som kräver mindre tid men ändå får in i princip lika mycket information kan vara bättre i mer tidspressade projekt, men ändå så vill man inte förlora för mycket av kvalitén. Det hade kunnat leda till att man fått utföra ännu fler tester och blivit ännu sämre ur ett tidsperspektiv.

7 Källor

Många olika sorters informationsinsamling användes inom examensarbetet. Allt från litteraturstudier, informationsinsamling från nätet från publikationer till informativa videor när det kom till programmeringsspråken och hur koden skrevs i detta, användes.

7.1 Testmetoder

[1]: Gerry Gaffney. "*Usability Testing*" [Online]

© 1999 Information & Design.

Tillgänglig: <http://infodesign.com.au/wp-content/uploads/UsabilityTesting.pdf>

[2014-08-26]

[2]: Steve Krug. "*Usability test script*" [Online]

© 2010 Steve Krug.

Tillgänglig: <http://sensible.com/downloads/test-script.pdf>

[2014-06-13]

[3]: Steve Krug. *Usability testing on 10 cents a day* [Online]

© 2000 Steve Krug.

Tillgänglig:

http://sensible.com/downloads/DMMTchapter09_for_personal_use_only.pdf

[2014-07-14]

[4]: Oracle. *How to conduct usability testing* [Online]

© 2012 Oracle Corporation.

Tillgänglig: <http://www.oracle.com/webfolder/ux/applications/uxd/assets/faq/how-to-conduct-usability-testing.pdf>

[2014-07-14]

[5]: Kaikkonen, Kallio, Kekäläinen, Kankainen, Cankar. *Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing*. [Online]

© 2005 UPA.

Tillgänglig:

<http://www.cse.chalmers.se/research/group/idc/ituniv/kurser/06/ucd/papers/kaikkonen%202005.pdf>

[2014-08-13]

[6]: Ana Amélia Amorim Carvalho (2014). *Usability testing of educational software: Methods, techniques and evaluators*. [Online]

Tillgänglig: <http://www.lits.dei.uminho.pt/utes.pdf>

[2014-06-13]

[7]: Nielsen, J. *Usability Engineering*.

© 1993 Academic Press.

ISBN: 0-12-518406-9

[8]: Rubin, J. (2011). *Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests – Second Edition*.
© 2008 Wiley Publishing, Inc.
ISBN: 978-0-470-18548-3

[9]: Shackel, B. (1990). *Human Factors and Usability*.
© 1991 Cambridge University Press.
ISBN: 0-521-36570-8

[10]: Booth, P. (1989). *An Introduction to Human-Computer Interaction*.
© 1989 Lawrence Erlbaum Associates Ltd.
ISBN: 0-86377-123-8

[11]: Goknur Kaplan Akilli (2005). *User satisfaction evaluation of an educational website*.
[Online]
Tillgänglig: <http://www.tojet.net/articles/v4i1/4111.pdf>
[2014-08-05]

[12]: Usabilitypartners (2014). *Standarder som hanterar produktanvändningskaraktäristika*.
[Online]
Tillgänglig: <http://www.usabilitypartners.se/om-anvandbarhet/iso-standarder.php>
[2014-10-10]

[13]: WAMMI. *What is WAMMI?* [Online]
© 2013 WAMMI.
Tillgänglig: <http://www.wammi.com/whatis.html>
[2014-08-26]

7.2 Företaget

[14]: Diflex AB (2014). [Online]
Tillgänglig: <http://www.diflex.se/>
[2014-06-13]

[15]: MarketTools. [Online]
© 2014 Market Time AB.
Tillgänglig: <http://www.markettools.se/>
[2014-06-13]

7.3 Programmeringsspråk

7.3.1 ASP.net

[16]: JavaScript Kit (2014), *What is ASP.net?*[Online]
Tillgänglig: <http://www.javascriptkit.com/howto/aspnet.shtml>
[2014-04-19]

7.3.2 C#

[17]: John Sharp. *Microsoft Visual C# 2010 Step by Step*.
© 2010 Microsoft Press.
ISBN: 2009939912

7.3.3 SQL

[18]: Thomas Padron-McCarthy, Tore Risch. *Databasteknik*.
© 2005 Thomas Padron-McCarthy, Tore Risch och Studentlitteratur.
Art.nr: 31838
ISBN: 91-44-04449-6

7.3.4 Javascript

[19]: Arman Danesh. *Javascript in 10 Simple Steps or Less*.
© 2004 Wiley Publishing, Inc.
ISBN: 0-7645-4241-9.

[20]: Nicholas C. Zakas. *Professional JavaScript™ for Web Developers*.
© 2005 Wiley Publishing, Inc.
ISBN-13: 978-0-7645-7908-0.

7.3.5 CSS

[21]: YouTube (2014). *XHTML and CSS Tutorial* [Online]
TheNewBoston – en respekterad programmerare med över 10 års erfarenhet som gör tydliga videor för utläring av grundläggande kunskaper inom olika programmeringsspråk.
Tillgänglig: https://www.youtube.com/watch?v=cqszz_OfAFQ&list=PLC1322B5A0180C946
[2014-04-20]

7.3.6 php

[22]: YouTube (2014). *Beginner PHP Tutorial* [Online]
TheNewBoston – en respekterad programmerare med över 10 års erfarenhet som gör tydliga videor för utläring av grundläggande kunskaper inom olika programmeringsspråk.
Tillgänglig: <https://www.youtube.com/watch?v=iCUV3iv9xOs&list=PL442FA2C127377F07>
[2014-04-20]

[23]: phpmyadmin (2015). *About*. [Online]
Tillgänglig:
http://www.phpmyadmin.net/home_page/index.php
[2015-02-10]

7.4 Allmän information

[24]: Lunds Tekniska Högskola (2014). *Datateknik 180 HP*. [Online]
Tillgänglig: <http://www.lth.se/utbildning/datateknik180>
[2014-10-07]

8 Terminologi

A:

Adobe Flash Player: Adobe Flash Player är en programvara för visning av multimedia så som videor eller ljudeffekter.

Agil (Agile): Grundtankarna bakom agil systemutveckling bygger på att göra kunden/användaren nöjd med det som utvecklas genom ett mycket nära samarbete under hela utvecklingstiden med täta och regelbundna möten mellan utvecklare och beställare/mottagare. Det agila synsättet anser att det oftare är människor och kommunikation än verktyg och formella dokument som löser problem under utvecklingsarbetet.

Om du vill läsa mer om agila metoder så rekommenderar programmeraren www.agilemethodology.org.

Asynkron: Innebär att ett skeende interagerar med ett annat skeende, men tidsmässigt helt oberoende av det andra skeendet. Motsatsen är synkron, då skeenderna koordinerar med varandra i tiden.

C:

C: Ett generellt programmeringsspråk avsett för strukturerad programmering.

C++: Ett programmeringsspråk likt Java, objektorienterat.

CLR (Common Language Runtime): En virtuell maskin och huvudkomponenten i ASP.net. Det är en implementering av standarden Common Language Infrastructure, som definierar en exekveringsmiljö för programkod.

CGI (Common Gateway Interface): Ett protokoll för hur en webserver kan köra program med argument och hur dessa anropas från en webbläsare via HTTP. CGI är alltså inte ett eget programspråk.

D:

Data: Information.

Databas: En organiserad struktur av data utformat för effektiv tillgång och lagring.

E:

Exception: Ett hanterat felmeddelande från koden.

F:

Fraps: Ett program som används för att spela in vad som händer på skärmen av en dator. Spelar även in ljud från en mikrofon.

H:

Hacking: Inom datasäkerhet är en hacker någon som försöker utnyttja svagheter i ett datorsystem eller datornätverk och får ut information som ska vara skyddad från andra användare så som personuppgifter, lösenord och så vidare.

Hemsidan: www.markettools.se, hemsidan som examensarbetet utfördes på.

HTML: HTML (förkortning för HyperText Markup Language) är ett märkspråk för hypertext och utgör tillsammans med TCP/IP och HTTP den grundläggande standarden för WWW (World Wide Web).

I:

Interaktiv hemsida: Hemsida som reagerar direkt på inmatningar som görs av användaren.

J:

Java: Ett objektorienterat programspråk som konstruerades av Sun Microsystems under 90-talets tidigare hälft.

O:

Objektorienterat programmeringsspråk: Objektorienterad programmering är en programmeringsmetod i vilken ett program kan innehålla en varierande uppsättning objekt som interagerar med varandra. De ursprungliga idéerna i metoden (objekt, inkapsling) uppstod i slutet av 1960-talet i programmeringsspråk för simulering. De byggdes sedan på med arv under mitten och slutet av 1980-talet. Denna metod att programmera anses av förespråkarna vara effektiv och kraftfull, speciellt vid konstruktion av större program, eftersom oönskad påverkan mellan programmets olika delar minimeras. Det är också oftare lättare att återanvända programdelar från ett program i ett annat, eftersom begreppen, objektklasserna, ofta blir generella och användbara i olika sammanhang.

P:

Python: Python är ett programspråk som utformades i slutet av 1980-talet. Python har utvecklats till ett kraftfullt och samtidigt smidigt språk med stöd för flera olika programmeringsparadigmer, bland annat objektorienterad och funktionell programmering.

S:

Skript: Ett skriptspråk är ett tolkat programspråk. Det färdiga programmet kan läsas av programmerare utan några speciella verktyg, men kräver en speciell exekveringsmiljö för att kunna köras.

T:

Table: Är vad databaser är uppbyggda av. Kan exempelvis finnas ett table innehållandes all information om en användare registrerad på hemsidan.

W:

Webläsare: Program som man använder för att komma åt internet. Till exempel Internet Explorer, Firefox, Safari och Google Chrome.

X:

XHTML: XHTML (Extensible HyperText Markup Language) är ett märkspråk och en vidareutveckling av HTML. Tanken är att man med XHTML (liksom med HTML) ska kunna bygga och länka samman dokument som kan visas över webben.

XML: Extensible Markup Language, XML, är ett universellt och utbyggbart märkspråk.

9 Bilagor

9.1 Bilaga 1: Direkt observation

9.1.1 Användbarhetstest

- Öppna webbläsare, rekommendation: Google Chrome eller Firefox.
- Webbläsare bör vara öppen för Google eller annan "Neutral" sida.

Hej, _____. Mitt namn är Dan Andersson, och jag kommer att vandra dig genom denna session i dag.

Innan vi börjar, har jag lite information till dig, och jag kommer att läsa högt för dig för att se till att den täcker allt.

Du har förmodligen redan har en bra uppfattning om varför vi bad dig komma hit, men låt mig gå igenom det igen det en kort stund för att det ska bli helt klart. Vi ber folk att prova att använda en webbplats som jag jobbar på, så att jag kan se om det fungerar som det är tänkt. Den sessionen bör ta ungefär en halvtimme.

Det första jag vill göra klart med en gång är att jag testar hemsidan, inte dig. Du kan inte göra något fel här. Så även om du får en känsla av att du inte förstår, vill jag att du visar detta för att kunna förbättra hemsidan så att så många som möjligt kan förstå den i framtiden.

När du använder webbplatsen, kommer jag att be dig så mycket som möjligt att försöka tänka högt: att säga vad du är tittar på, vad du försöker göra, och vad du tänkte. Detta kommer att vara en stor hjälp för oss.

Dessutom ska du inte oroa dig att du kommer att skada mina känslor. Jag gör detta för att förbättra webbplatsen, så jag behöver höra din ärliga reaktioner.

9.1.2 Frågor

Om du har några frågor, bara att fråga dem.

Jag kanske inte kan svara på dem direkt, eftersom jag är intresserad av hur människor gör när de inte har någon som sitter bredvid dem för att hjälpa till. Men om du fortfarande har några frågor när vi är klara ska jag försöka besvara dem då. Och om du behöver ta en paus på någon punkt, så låt mig veta.

Du kanske har märkt mikrofonen. Med din tillåtelse kommer vi att spela in vad som händer på skärm och vårt samtal. Inspelningen kommer endast att användas för att hjälpa oss att lista ut hur man kan förbättra webbplatsen och det kommer inte ses av någon annan än de som arbetar på det här projektet. Och det hjälper mig, därför att jag inte behöver ta så många anteckningar.

Om du vill, jag ska be dig att skriva en enkel tillstånd blankett för oss. Den säger bara att vi har en tillåtelse att spela in dig, och att inspelningen kommer endast ses av mig.

- Ge dem en inspelningstillståndsformuläret och en penna.
- Medan de underteckna den, starta skärm och inspelningsprogram (Fraps) och börja spela in.

Innan vi tittar på hemsidan, skulle jag vilja be er svara på några snabba frågor.

Först, vad är ditt yrke?

Vad gör du om dagarna?

Nu, ungefär hur många timmar i veckan totalt skulle du säga att du spenderar med att använda Internet, inklusive surfning och e-post, på jobbet och hemma?

Och vad är uppdelningen mellan e-post och surfa-a grov procentandel?

Vilka typer av webbplatser tittar du på när du surfar på webben?

Har du några favoritwebbplatser?

Ok, bra. Vi är klar med frågorna, och vi kan börja titta på hemsidan.

9.1.3 Uppgifter

- Klicka på bokmärket för webbplatsens hemsida.

Först kommer jag att be er att titta på den här sidan och berätta mig vad du gör av det: det som slår dig om det, vars webbplats du tycker att det är, vad du kan göra här och vad det är för.

Bara titta runt och göra lite berättelse.

Du kan bläddra om du vill, men inte klicka på något ännu.

- Låt detta pågå i en eller max två minuter.

Tack. Nu ska jag be er att försöka göra några specifika uppgifter. Jag kommer att läsa var och en högt och ge dig ett tryckt exemplar.

Och återigen, så mycket som möjligt, det kommer att hjälpa oss om du kan försöka tänka högt.

- Ge deltagaren det första scenariot, och läsa den högt.
- Tillåt användaren att fortsätta tills du inte känner för det producerar något värde eller användaren blir mycket frustrerad.
- Upprepa för varje uppgift eller tills tiden tar slut.

9.1.4 Frågor angående test i efterhand

Har du några frågor till mig, nu när vi är klara?

- Stoppa skärm domare och spara filen.

Informationen du har gett oss kommer att vara till stor hjälp för hemsidans användbarhet.

Tack så mycket, du kan nu gå.

9.1.5 Scenario 1

*Jag vill att du ska försöka skriva ut en lista med de avslutade auktionerna mellan den första januari 2014 till dagens datum för alla **avslutade auktioner**.*

- Kontrollera att personen hittar till följande sida:
<http://test2.markettools.se/PrintPage.aspx>
- Kontrollera att personen lyckas välja datum.
- Kontrollera att personen lyckas kryssa i checkboxen för att bara visa avslutade auktioner.
- Kontrollera att personen hittar utskriftsknappen.

9.1.6 Scenario 2

Detta scenario kräver att scenario 1 lyckats, annars så visar du hur detta skulle ha lösts för att kunna göra scenario 2.

*Jag vill att du ska försöka skriva ut ett specifikt datum med titel ” **En överdrivet lång titel för att testa en väldigt specifik sak**” som avslutades någon gång i september 2014. Du skall sedan skriva ut ett specifikt fakturaunderlag för denna.*

- Kontrollera att personen hittar till följande sida:
<http://test2.markettools.se/Invoice.aspx?m=318>
- Kontrollera att personen lyckas hitta objekt med ID 318.
- Kontrollera att personen lyckas hitta utskriftsknappen.

9.2 Bilaga 2: Undersökningar

Hej, du har blivit utvald till användbarhetstestare för www.markettools.se eftersom du passar in i vår målgrupp (20-80 år gammal) och vi skulle uppskatta om du kunde testa och svara på lite frågor angående vår hemsida. Detta är en del av mitt examensarbete på LTH Campus Helsingborg och all feedback är bra feedback.

Jag vill att du ska helst sitta ner och vara relativt avslappnad när du sitter ner och gör testen men det är inget krav att helt avskärma dig från omvärlden under tiden då det ska simuleras som om det görs under en arbetsdag.

Innan vi tittar på hemsidan, skulle jag vilja be er svara på några snabba frågor.

Först, vad är ditt yrke?

Vad gör du om dagarna?

Nu, ungefär hur många timmar i veckan totalt skulle du säga att du spenderar med att använda Internet, inklusive surfning och e-post, på jobbet och hemma?

Och vad är uppdelningen mellan e-post och surfa-a grov procentandel?

Vilka typer av webbplatser tittar du på när du surfar på webben?

Har du några favoritwebbplatser?

Ok, bra. Vi är klar med frågorna, och vi kan börja titta på hemsidan.

Öppna din webbläsare, där jag rekommenderar användning av Google Chrome eller Firefox, men Internet Explorer fungerar även.
Gå in på www.test2.markettools.se och logga in med följande uppgifter i menyn till höger.

Användarnamn: test
Lösenord: test123

Jag ber folk att prova att använda en webbplats som jag jobbar på, så att jag kan se om det fungerar som det är tänkt. Testet bör ta ungefär 30 minuter.

Det första jag vill göra klart med en gång är att jag testar hemsidan, inte dig. Du kan inte göra något fel här. Så även om du får en känsla av att du inte förstår, vill jag att du visar detta för att kunna förbättra hemsidan så att så många som möjligt kan förstå den i framtiden.

Dessutom ska du inte oroa dig att du kommer att skada mina känslor. Jag gör detta för att förbättra webbplatsen, så jag behöver höra din ärliga reaktioner.

- **Först kommer jag att be er att titta på den här sidan och berätta mig vad du gör av det: det som slår dig om det, vars webbplats du tycker att det är, vad du kan göra här och vad det är för. Bara titta runt och göra lite berättelse. Du kan bläddra om du vill, men inte klicka på något ännu.**

Tack. Nu ska jag be er att försöka göra några specifika uppgifter.

9.2.1 Uppgift 1

Jag vill att du ska försöka skriva ut en lista med de avslutade auktionerna mellan den första januari 2014 till dagens datum för alla **avslutade auktioner**.

Lyckades du hitta var man skriver ut avslutade auktioner?

Lyckades du välja rätt datum?

Lyckades du att bara få fram de avslutade auktionerna?

Lyckades du få upp rutan om att skriva ut listan?

9.2.2 Uppgift 2

Denna uppgift kräver att uppgift 1 lyckats.

Jag vill att du ska försöka skriva ut ett specifikt datum med titel ” **En överdrivet lång titel för att testa en väldigt specifik sak**” som avslutades någon gång i september 2014. Du skall sedan skriva ut ett specifikt fakturaunderlag för denna.

Lyckades du hitta var man skriver ut fakturaunderlag?

Lyckades du välja rätt objekt?

Lyckades du få upp rutan om att skriva ut fakturaunderlaget?

Informationen du har gett oss kommer att vara till stor hjälp för hemsidans användbarhet. Tack så mycket, testet är nu slut.

9.3 Bilaga 3: Tidsplan

Kontakt med Diflex AB

Möte på företaget,
introduktion till projekt

Få en grov överblick av
projektet

Studera C# och .NET

Studera möjligheter för
användartester

Få in synpunkter och idéer från
kund

Avsluta dokumentation

Anordna en testgrupp

Val av testmetoder

Programmera

Påbörja examensrapport

Användartester (bilaga 2)
för att finslipa.

Rätta till problem som uppstår vid
ovanstående

Vecka 13

Möte med Christin Lindholm om
examinator och handledare

Skriva frågeställning, tidsplan

Registrera examensarbete

Börja förstå koden och hur
systemet är uppbyggt

Dokumentera det som finns

Göra upp en plan för
användartester

Se vad som rimligt är genomförbart inom
given tidsram av det som föreslagits av
kund och projekt

Skriv utförlig tidsplanering

Testa systemet för buggar som redan exis-
terar och fixa dessa innan mer kod skrivs

Konturläs examensrapport
och rätta eventuella misstag

Fixa de sista buggarna och dokumentera
allt som har gjorts

Skicka in examensrapport till examinator
för godkännande

Vecka 30

Bild 10: Tidsplan