

Institutionen för datavetenskap
Department of Computer and Information Science

Examensarbete

**Design och implementation av webbenkäter:
kvalitet, svarsfrekvens och underhåll**

av

Per Jonsson

LIU-IDA/LITH-EX-G--14/049--SE

2014-06-10



Linköpings universitet

Linköpings universitet
Institutionen för datavetenskap

Examensarbete

Design och implementation av webbenkäter: kvalitet, svarsfrekvens och underhåll

av

Per Jonsson

LIU-IDA/LITH-EX-G--14/049--SE

2014-06-10

Handledare: Johan Åberg

Examinator: Peter Dalenius

Design och implementation av webbenkäter: kvalitet, svarsfrekvens och underhåll

Per Jonsson

SAMMANFATTNING

En webbapplikation för analys och administration av webbenkäter har designats och implementerats. Dess syfte är att maximera svars kvalitet och svarsfrekvens samt att vara underhållbar. Uppdragsgivaren Ericsson Linköping har utfärdat kravspecifikationen för applikationen. Hänsyn har tagits till aspekterna webbenkätdesign och underhållbarhet av mjukvara. Underhållbarhetsmodeller för mjukvara med tillhörande metriker, samt designmodeller och rekommendationer för webbenkäter har studerats. Arbetets bidrag till dessa områden är en praktisk modell som tillämpar rådande forskning, i form av en webbapplikation. Applikationen har testats mot modeller och rekommendationer för underhållbarhet och enkätdesign. Applikationen uppvisar hög grad av analyserbarhet, förändringsbarhet och testbarhet, men inte stabilitet. Effekten av enkätdesignen har inte utvärderats. Modellen för underhållbarhet kan klarlägga orsak och verkan i mjukvarusystem och bidra till utveckling av programvara med hög kvalitet.

INLEDNING

I detta kapitel motiveras behovet av praktisk tillämpning av forskning kring webbenkäter och underhållbarhet. Syftet med arbetet förklaras och forskningsfrågor presenteras.

Motivering

Arbetet i rapporten angriper problem från två olika forskningsområden: Webbenkätdesign och underhållbarhet av mjukvara. Behovet av god webbenkätdesign aktualiseras genom den stora marknad som webbenkäter potentiellt har, men som drabbats av sjunkande svarsfrekvenser [6, 13, 14]. Behovet av att kunna underhålla webbapplikationer kommer att växa om utvecklare för skrivbordsbaserade applikationer väljer att övergå till att skapa molnbaserade webbapplikationer. De kommer då att kräva en annan typ av underhåll.

Webbenkäters utformning

Det är väsentligt att studera hur webbenkäter bäst utformas för maximering av svars kvalitet och svarsfrekvens, för att de fortsatt ska kunna vara relevanta. Design av webbenkäter är ett omfattande forskningsområde och relativt nytt i förhållande till traditionella enkäter. Respondenters interaktion med webbenkäter skiljer sig från de traditionella enkäterna. Webbenkätdesign skiljer även ut sig som delmängd av det mer generella begreppet webbdesign, genom egna konventioner för interaktion. Webben erbjuder en ny typ av mer självständig och dynamisk interaktion

med enkäter. Men respondenters beteende förändras också över tid, då vissa typer av webbenkäter blir etablerade och tillvänjning sker. När beteenden förändras måste forskningen följa med. Rådande forskning ger rekommendationer och uppmanar till praktisk tillämpning av dem [20].

Underhållbarhet av mjukvara

Till skillnad från forskningen kring webbenkäter finns inte etablerade konventioner kring underhållbarhet av mjukvara eller vad begreppet i praktiken ska innebära [3]. Det gäller i synnerhet för webbapplikationer, som skiljer sig i arkitektur från objektorienterade system som är de traditionella målen för underhållbarhetsmodeller. Praktisk tillämpning är då en väg fram mot en standardisering av begreppet underhållbarhet.

Syfte

Att implementera en webbapplikation för att generera, administrera och analysera webbenkäter med enkätdesign för maximering av svars kvalitet och svarsfrekvens. Att även optimera applikationens underhållbarhet genom att tillämpa modeller och principer för underhållbarhet av mjukvara.

Frågeställningar

- Hur kan ett webbenkätsystem designas för att maximera svars kvalitet och svarsfrekvens?
- Hur kan webbenkätsystemet implementeras för att få en hög grad av underhållbarhet?

Avgränsningar

I Bilaga 1 tillhandahålls en kravspecifikation på systemet från uppdragsgivaren Ericsson Linköping. Arbetet som utförts i rapporten omfattas enbart av designen av enkätmodellen samt att bygga systemet enligt goda underhållbarhetsprinciper. Det vill säga: Systemet ska vara lätt att använda och lätt att underhålla. Resten av systemet och Ericssons krav kommer att utvecklas och slutföras utanför ramen för examensarbetets omfattning.

TEORI

I detta kapitel görs en systematisk genomgång av forskning kring webbenkäter och deras design, samt underhållbarhet av mjukvara.

Design av webbenkäter

[20] går systematiskt igenom aktuell forskning inom området design av webbenkäter och jämför den ibland även med forskning kring traditionella enkäter. Problematik för

enkäter i olika moder behandlas: vetenskapliga enkäter, frivilliga enkäter såväl som obligatoriska enkäter. Forskningsrön kring de olika enkättyperna är inte alltid jämförbara sinsemellan. Svarsfrekvensen i en enkät kan till exempel i ett frivilligt läge påverkas av incitament som belöningar. Sådana incitament är inte lika framgångsrika i ett scenario där ett företag eller en liknande organisation genomför en enkät bland anställda. Där kan istället e-post-påminnelser med länkar till enkäten få en avgörande betydelse. Förutom en genomgång av forskning presenterar [20] även rekommendationer för utformning av webb-enkäter och uppmanar till pragmatisk prövning av dessa. Riktlinjerna återfinns sammanfattade under rubrikerna nedan.

Reducering av fel på grund av uteblivna svar

- E-post-meddelanden med länkar till undersökningen bör skickas ut, helst kompletteras med brev eller vykort.
- Oönskade överraskningar bör minimeras, såsom oväntat långsamma framsteg på grund av att enkäten varit svårare eller längre än väntat.
- Att sondera efter uteblivna och ej ifyllda svar kan delvis vara till hjälp.

Reducering av observationsfel och avvikelser

Delas in i fyra områden.

1. Grundläggande designbeslut

- En design med sidindelning är att föredra framför en design med scrollning.
- Ljus bakgrund är att föredra, grafik i bakgrunden bör undvikas
- Ett rutnätsystem (eng. *grid system*) till enkätfrågorna kan inverka negativt på svars kvalitet och svarsfrekvens.
- Standardkonventioner bör användas: Fet stil på typsnittet till texten på frågetiteln, normal stil för svarsalternativen, samt versaler eller kursiv stil för emfas.
- Det bör försäkras att inmatningsverktyget matchar den avsedda funktionen: att exempelvis radioknappar används när enbart ett (1) svar krävs.
- Den översta vänstra delen av en webbsida är mest visuellt framträdande, den viktigaste informationen bör placeras där.
- Frågor bör vänsterjusteras och numreras om de är ett flertal så att respondenten kan få en uppfattning om hur många de är.
- Inmatningsfältet (t.ex. radioknapp, kryssruta) för ett svarsalternativ bör placeras till vänster om dess motsvarande etikett. Alla svarsalternativ bör placeras i samma rad eller kolumn, och upplägg med multipla kolumner bör undvikas.
- En "nästa"- och en "föregående"-knapp längst ner på varje sida bör inkluderas, där "föregående"-knappen görs mindre framträdande.

2. Enkätdesign

- Bilder bör helst undvikas där de kan påverka respondentens tolkning av frågan.
- De visuella ledtrådarna och hjälpmedlen bör vara konsekventa med den gradering som avses. Om en skala med jämna avstånd mellan punkter används bör skalan vara skalenlig i alla sammanhang, det vill säga likadan i olika webbläsare och enheter. Mittpunkten bör alltid vara konceptuellt neutral.

- I så stor mån som möjligt bör svarsalternativ av typen "Vet ej" eller "Har ingen åsikt" undvikas, eller separeras visuellt från de konkreta svarsalternativen "Ja", "Nej", med flera.
- Färger och siffror kan leda till missförstånd i sammanhang där de inte har konkret betydelse. Om siffror används, bör räknetalen från 1 till n nyttjas.
- Varje punkt på en skala bör förses med en textetikett.

3. Interaktiva tillägg

- Definitioner som förklarar tekniska begrepp bör erbjudas och det bör vara lätt att komma åt dem. De kan visas automatiskt om respondenten verkar ha svårigheter, eller visas i höjd med frågetexten.
- Om det upptäcks att en respondent går relativt snabbt igenom enkäten, kan den uppmanas att sänka tempot.
- Att indikera framsteg i enkäten med hjälp av förloppsindikatorer (eng. *progress bars*) fungerar enbart om respondenten uppfattar det som att den gjort oväntat stora framsteg.
- Automatisk kontrollräkning kan hjälpa respondenten med frågor som involverar beräkning – som att uppskatta procentsatser.

4. Självadministrering och kognitiv belastning

- När en webbenkät upplevs ha god säkerhet och uppfyller krav på anonymitet underlättar det insamling av känslig data, när respondenten fyller i enkäten i en privat miljö.

Förloppsindikatorer

I likhet med [20] har [21] intagit ett kritiskt förhållningssätt till förloppsindikatorer i webbenkäter. Graden av bortfall har jämförts mellan respondentgrupper som i sina enkäter hade tillgång till förloppsindikatorer, med grupper utan sådana. Uppfattningen att sådana verktyg minskar svarsbortfall bör ifrågasättas enligt [21], som har identifierat en rad väsentliga skillnader mellan webbenkäter och andra typer av datorapplikationer. Behovet av en indikator är berättigat när användaren frivilligt kommit till en punkt där den väntar på att få en uppgift utförd inom programmet, som att spara en fil i en ordbehandlings-applikation. Men i fallet med en webbenkät där användaren själv utför den tidskrävande uppgiften (att fylla i formulären) finns det kanske inte något motiv annat än tvång att slutföra den. Då kan indikatorn få motsatt effekt.

[5] har visat att en indikator kan förhöja användares upplevelse av enkäten, men även orsaka det motsatta beroende på om användare upplever indikatorn som en besvikelse på grund av för högt ställda förväntningar på ett snabbare förlopp. Det har även studerats om sättet som förloppsindikatorn visualiseras på kan förhöja effekten den har på användaren. [10] har lagt fram riktlinjer som föreslår hur en indikator kan designas för att bli mer användarvänlig: Visa uppskattad tid för enskilda uppgifter, dela upp en uppgift i logiska partitioner, visa förloppsindikator för både uppgiften och hela enkäten i stort, ta hjälp av målvisualisering om det finns en belöning, samt att tillhandahålla ytterligare detaljer om uppgiften och användarens framsteg inom kort räckhåll.

Utskick och uppföljning

En metod för att skapa respondentunderlag består av utskick med inbjudningar och uppmaningar till deltagande. [9, 14] har undersökt effekten av personliga enkätupplevelser på svarsfrekvens, fullföljning och deltagande. Det motiveras av en dominerande trend bland enkäter – inte enbart webbenkäter – där svarsfrekvenser sjunker. Personliga upplevelser skapas till exempel genom att anpassa enkätfrågor dynamiskt beroende på hur respondenten svarar, eller genom personliga inbjudningar via e-post. Det har påvisats att en personlig hälsning har mest inflytande när avsändaren är auktoritär (eng. *power source*) [9]. Om målet är att respondenter ska fullfölja hela enkäten bör inte för många personifierade påminnelsemeddelanden skickas ut, mer än tre eller fyra meddelanden totalt rekommenderas inte [14].

Svarsfrekvens

[13] har mot bakgrund av sjunkande enkätsvarsfrekvenser ifrågasatt vad som skulle kunna vara en acceptabel svarsfrekvens, samt hur svarsfrekvensen enskilt påverkar en undersöknings kvalitet. Ett problem är att svarsfrekvens kan definieras på många olika sätt och inte ensamt bör utgöra ett mått på kvalitet. Ett annat problem är en ökande ovilja hos respondenter att svara. Måttet svarsfrekvens bör kompletteras med mer detaljerade fakta om deltagare i undersökningen, bland annat hur de ansvariga har försökt öka deltagande, för att påvisa validitet.

För att å andra sidan försöka påverka svarsfrekvensen istället för att mäta den annorlunda har [6] presenterat praktiska förslag på hur svarsfrekvens kan påverkas. En webbenkäts livscykel delas in i fyra stadier: Skapande, leverans, (respondentens) fullföljande, och returnering (efterbehandling). Enkätskapare bör se över hela livscykeln, ta hänsyn till empirisk forskning och analysera egna resultat baserat på forskning. Därtill har [6] efterlyst en jämförelse mellan etablerade mjukvarulösningar för att se vilka som genererar högst svarsfrekvens.

Underhållbarhet av mjukvara

Underhållbarhet av mjukvara (eng. *software maintainability*) är ett begrepp vars närmast standardiserade definition återfinns i en delmängd till ISO-standard ISO/IEC 9126^{1 2 3 4}. Den har nyligen ersatts av den reviderade versionen ISO/IEC 25010:2011⁵. Den forskning och de

¹ ISO, ISO/IEC 9126-1: Software engineering - product quality - part 1: Quality model, Geneva, Switzerland, 2001.

² ISO/IEC TR 9126-2: Software engineering - product quality - part 2: External metrics, Geneva, Switzerland, 2003.

³ ISO/IEC TR 9126-3: Software engineering - product quality - part 3: Internal metrics, Geneva, Switzerland, 2003.

⁴ ISO/IEC TR 9126-4: Software engineering - product quality - part 4: Quality in use metrics, Geneva, Switzerland, 2004.

modeller kring underhållbarhet för mjukvara som existerar tycks mest cirkulera kring den förstnämnda standarden.

Mätning av mjukvaras underhållbarhet: SIG-modellen

Utifrån ett upplevt behov av att kunna använda en mer tillfredsställande modell för mjukvarukvalitet än den i ISO/IEC 9126 har [7] utifrån några av dess principer utvecklat den så kallade *SIG Software Maintainability Model* (nedan benämnd SIG-modellen). Den har tagit fasta på principerna anpassningsbarhet (eng. *adaptability*), förändringsbarhet (*changeability*), stabilitet (*stability*), och testbarhet (*testability*) – som ett system ska kunna uppvisa för att vara underhållbart, med hjälp av specifika metriker (eng. *metric*: ett mått på en mjukvaruegenskap) som indikatorer. Modellen fungerar även som motvikt till ett mer etablerat underhållbarhetsmått: *Maintainability Index (MI)* [4], som ger en numerisk bedömning på ett systems grad av underhållbarhet baserat på källkodens utformning, med hjälp av en formel. Problemet som lyfts fram av [7] är dock att MI inte klarlägger orsak och verkan.

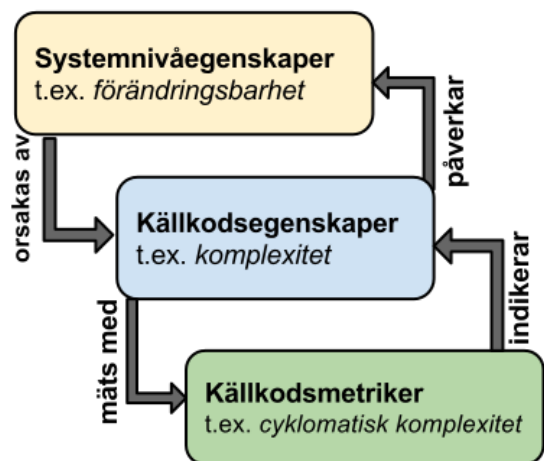
I SIG-modellen har man valt att bygga analysen utifrån systemets källkodsegenskaper, oberoende av programmeringsspråk eller arkitektur, med enkla mätmetoder som klarlägger orsak och verkan. Modellen består av systemegenskaper som kartläggs mot källkodsegenskaper och metriker. Det illustreras i Figur 1, där egenskapen komplexitet hos källkoden påverkar egenskapen förändringsbarhet på systemnivå. I andra steget fastställs en eller flera källkodmetriker som mäter källkodsegenskaper; exempelvis cyklomatisk komplexitet som mäter egenskapen komplexitet. Källkodsegenskaperna som bedöms i SIG-modellen är volym, komplexitet per enhet, duplicering, enhetsstorlek och enhetstestning. Med *enhet* avses den minsta enhet kod som enskilt kan testas och exekveras, till exempel en funktion. Egenskaperna på källkodsnivå influerar underhållbarhetsegenskaper på systemnivå enligt Tabell 1. Varje egenskap bedöms sedan på en 5-gradig skala.

Volym, det vill säga ett systems storlek, påverkar dess analyserbarhet och kräver större ansträngning att underhålla. Detta mäts enklast genom att räkna antal rader kod i systemet.

Komplexitet påverkar analyserbarhet och testbarhet i ett system och mäts med cyklomatisk komplexitet på varje enskild enhet, genom en formel som analyserar kontrollflödet genom ett program [12]. På så vis går det att betygsätta risknivån för varje enhet.

Med *duplicering* menas antalet kodkloner, det vill säga kodblock som förekommer oförändrade på fler än en plats i

⁵ ISO/IEC 25010:2011: Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models.



Figur 1: SIG-modellen. Systemegenskaper orsakas av källkodsegenskaper som mäts med källkodsmetriker. Anpassad från [7].

källkoden. De kan mätas genom strängmatchning.

Storleken på *enheterna* av kod i ett system påverkar dess underhållbarhet. Stora kodenheter påverkar analyserbarhet och testbarhet negativt. Måttet bestäms genom antal rader kod per enhet.

Enhetstestning påverkar testbarheten, stabiliteten och analyserbarheten positivt. Detta mäts genom så kallad code coverage (hur stor del av koden som testerna täcker) samt antalet *assert*-satsar i testfallen.

När alla egenskaper har betygsatts beräknas en totalsumma för systemets underhållbarhet. På systemnivå går det sedan att spåra vad som är orsak och verkan.

Empiriskt stöd för underhållbarhetsmodeller

SIG-modellen har prövats av [11], som undersökt hur dess betygssystem korrelerar med hastigheten som underhållspersonal åtgärdar problem i mjukvara med. En majoritet av måtten i modellen stämde överens med graden av underhållbarhet. SIG-modellen har även fått utgöra underlag för certifiering för mjukvaruprodukter [2].

[1] har utgått ifrån ISO 9126 på ett liknande sätt men tillämpat objektorienterade metriker och kompletterat modellen med informationsutvinningsmetoder och algoritmer för att söka efter mönster, samband och trender. ISO 9126 har i en annan studie kompletterats med bayesiska nätverk, en form av beslutsdiagram [8].

Ifrågasättande av underhållbarhetsmodeller och metriker

I motsats till de som propagerar för underhållbarhetsmodeller menar [3] att det inte går att precis definiera vad underhållbarhet av ett mjukvarusystem innebär, för trots etableringen av begreppet underhållbarhet råder inte konsensus i forskningen om hur begreppet ska tolkas. Metaforiskt medhåll har givits av [16], vilka inte funnit

Källkodsegenskaper

Underhållbarhetsegenskaper	Källkodsegenskaper				
	Volym	Komplexitet per enhet	Duplicering	Enhetsstorlek	Enhetstestning
Analyserbarhet	X		X	X	X
Förändringsbarhet		X	X		
Stabilitet					X
Testbarhet		X		X	X

Tabell 1. Källkodsegenskaper kartlagda mot systemegenskaper [7].

evidens för effektivitet i tekniker och modeller som påstås förutspå underhållbarhet i mjukvara. [18] menar att underhållbarhetsmetriker åtminstone måste kompletteras med en evaluering på systemnivå (vilket görs i SIG-modellen). Val av fel metriker i fel kontext kan inverka menligt på ett systems underhållbarhet.

METOD

I detta kapitel beskrivs först hur metoden för att implementera systemet tekniskt har utförts. Därefter beskrivs hur webbdesignen har implementerats, samt ansträngningar för att göra systemet underhållbart.

Implementation

Arbetet har utförts i en samarbetsform med agila inslag kunnat ändra inriktning utifrån kundens önskemål. Kundens önskemål har varit en av de yttre faktorerna som styr hur mycket av den avsedda designen som tillämpats. Detta har dock inte påverkat implementationen så att avsteg från de tilltänkta underhållbarhetsaspekterna [7] har behövt göras.

Uppdragsgivaren har inte begärt att systemet ska skyddas av sekretess. Därför har hela systemet öppen källkod. Källkoden har versionshanterats med hjälp av verktyget Git och publicerats på versionshanteringsportalen GitHub⁶.

Webbapplikationramverket Meteor⁷ har varit det verktyg som systemet har implementerats i och som byggt den körbara webbapplikationen.

Applikationen har testats i webbläsarna Google Chrome (version 34), Internet Explorer (version 9, 10, 11) och Mozilla Firefox (version 28, 29), men främst utvecklats med Google Chrome som primär webbläsare.

⁶ GitHub. <https://github.com/>

⁷ Meteor. <https://www.meteor.com/>

Som CSS-ramverk till presentationslagret av applikationen har Twitter Bootstrap⁸ använts och justerats för att passa Ericssons grafiska profil, med hjälp av CSS-preprocessorn Less⁹. Den ger möjlighet att ställa in variabler för färger och typsnitt, med mera. Mallsystemet för HTML som använts har varit Jade¹⁰, som dynamiskt genererar HTML-kod under körning. All den egna programmeringslogiken har skrivits i CoffeeScript¹¹, vars körmiljö förkompilerar koden till JavaScript.

Design

För att besvara hur ett webbenkätsystem kan designas för att maximera svars kvalitet och svarsfrekvens har designbeslut tagits med stöd i rekommendationer från [20], som listats i punktform i teorikapitlet. Besluten har utgått från de listade punkterna och pragmatiskt anpassats (som [20] rekommenderar) till kundens krav samt vad som har varit tidsmässigt realiserbart inom examensarbetets omfattning; det har därmed inte varit möjligt att ta hänsyn till alla riktlinjer.

För att exemplifiera tillvägagångssättet med kontroll mot den ovan nämnda punktlistan, har en sådan funktion som uppmaning till respondenter att sänka hastigheten inte implementerats, på grund av tidsbrist. Kontrollräknings-funktionalitet har inte implementerats på grund av att kunden inte önskat en sådan funktion. Rutnätsystem och förloppsindikatorer har inte implementerats på grund av att de inte är rekommenderade. Däremot har ljus bakgrund och staplade svarsalternativ i rader implementerats på grund av att de är rekommenderade. På dylikt vis har arbetet fortlöpt.

Enkätdesign har utgjort en delmängd av systemet, resten av systemet har tillämpats enligt allmänna designprinciper, influerat av [17, 19]. I de delar som omfattar webb-enkätdesign har hänsyn tagits till de riktlinjer som har lagts fram av [20]. Ericsson har interna riktlinjer som styr hur webbsidor som representerar Ericsson får se ut. Det rör sig om färger, typsnitt, struktur, interaktionsdesign, med mera. Dessa har dock inte motsagt riktlinjerna som [20] föreslagit, och därmed i stor grad fått styra utseendet på den implementerade webbdesignen.

Metoden för att implementera designen programmatiskt har varit att skapa mallar i språket CSS. Med CSS har den avsedda designen tillförts HTML-dokument, som strukturerat den avsedda formen.

Underhållbarhet

För att besvara hur webbenkätsystemet kan implementeras för att få en hög grad av underhållbarhet har hänsyn tagits till SIG-modellens [7] fyra systemegenskaper analyser-

barhet, förändringsbarhet, stabilitet och testbarhet, under utvecklingsfasen. Detta genom ansträngningar att hålla nere volymen på systemet, undvika upprepning av kod, omfaktorer stora funktioner och kodstrukturer, samt modularisera kod och separera kodansvar.

Twitter Bootstrap, Less, Jade och CoffeeScript har valts för att minska komplexiteten och totalvolymen på systemet samt öka läsbarheten, för att följa SIG-modellens riktlinjer [7]. Databasen har implementerats i MongoDB¹², som är en JavaScriptbaserad databas. Detta har gjort att hela applikationsstacken använder samma programmerings-språk. Avsikten med det har varit att öka analyserbarheten och stabiliteten i systemet som helhet.

Med hänsyn till underhållbarhetsegenskaperna från SIG-modellen [7] har källkoden löpande omfaktorerats för att undvika duplicering, stora kodblock och komplexitet. Detta har gjorts ofta, för att bryta ner kod i mindre funktioner. Kodansvar har separerats genom en mappstruktur som visar tydligt vad varje kodfil har för uppgift i systemet. Koden har försetts med kommentarer som tydliggör ansvar.

Utvärdering

Design

Designen har utvärderats mot de rekommendationer från [20] som listats i kapitlet Teori, genom att undersöka hur många av punkterna som tillämpats samt hur och varför de tillämpats eller inte tillämpats. Designen redovisas sedan genom skärmdumpar från den utvecklade webb-applikationen, i kapitlet Resultat.

Underhållbarhet

Implementationen har utvärderats mot SIG-modellen [7]. De fyra systemegenskaperna analyserbarhet, förändringsbarhet, stabilitet och testbarhet har uppskattats genom att bedöma de beroende källkodsegenskaperna volym, komplexitet per enhet, duplicering och enhetsstorlek. Räkning av antal rader kod har mätt volym, beräkning av cyklomatisk komplexitet har mätt komplexitet, räkning av kodkloner har mätt duplicering och antal rader kod per enhet har mätt enhetsstorlek.

De rader källkod i systemet som uppmäts har exkluderat kod som tillförts via importerade bibliotek och dylikt. Blankrader eller bortkommenterade rader har inte räknats. Metoden som istället valts har varit att jämföra hur mycket volymen på systemet minskar procentuellt sett, med CoffeeScriptkod som ersättning för JavaScript. På så sätt har metriken blivit relaterad till en proportion.

Cyklomatisk komplexitet har beräknats genom ett verktyg som analyserar den JavaScriptkod¹³ som programmerings-logiken skrivits i.

⁸ Bootstrap 3. <http://getbootstrap.com/>

⁹ Less. <http://lesscss.org/>

¹⁰ Jade: Node templating engine. <http://jade-lang.com/>

¹¹ CoffeeScript. <http://coffeescript.org/>

¹² MongoDB. <http://www.mongodb.org/>

¹³ JSComplexity. <http://jscomplexity.org/>

Komplexiteten har sedan bedömts utifrån SIG-modellens tabell med hjälp av en 5-gradig skala.

Kodkloner har beräknats med exakt strängmatchning, bortsett från blankrader, utifrån block av kod större än 6 rader, och bedömts på en 5-gradig skala.

Rader kod per enhet har beräknats och bedömts utifrån risknivå på en 5-gradig skala – ju fler stora enheter desto större risk och lägre betyg – genom att undersöka hur många block som består av 6 rader kod eller fler.

En sammanställning har slutligen genomförts med ett medeltal för det totala systemet när alla betyg räknats samman.

RESULTAT

I detta kapitel presenteras den slutliga implementationen med illustrativt fokus på den interaktiva designen. Designen och underhållbarheten utvärderas därefter.

Implementation

Systemets versionshanterade källkod finns tillgänglig på GitHub¹⁴. Applikationen installeras och körs med hjälp av Node.js, MongoDB och Meteor som beroenden. Systemets underhållbarhet utvärderas under rubriken Utvärdering.

Design

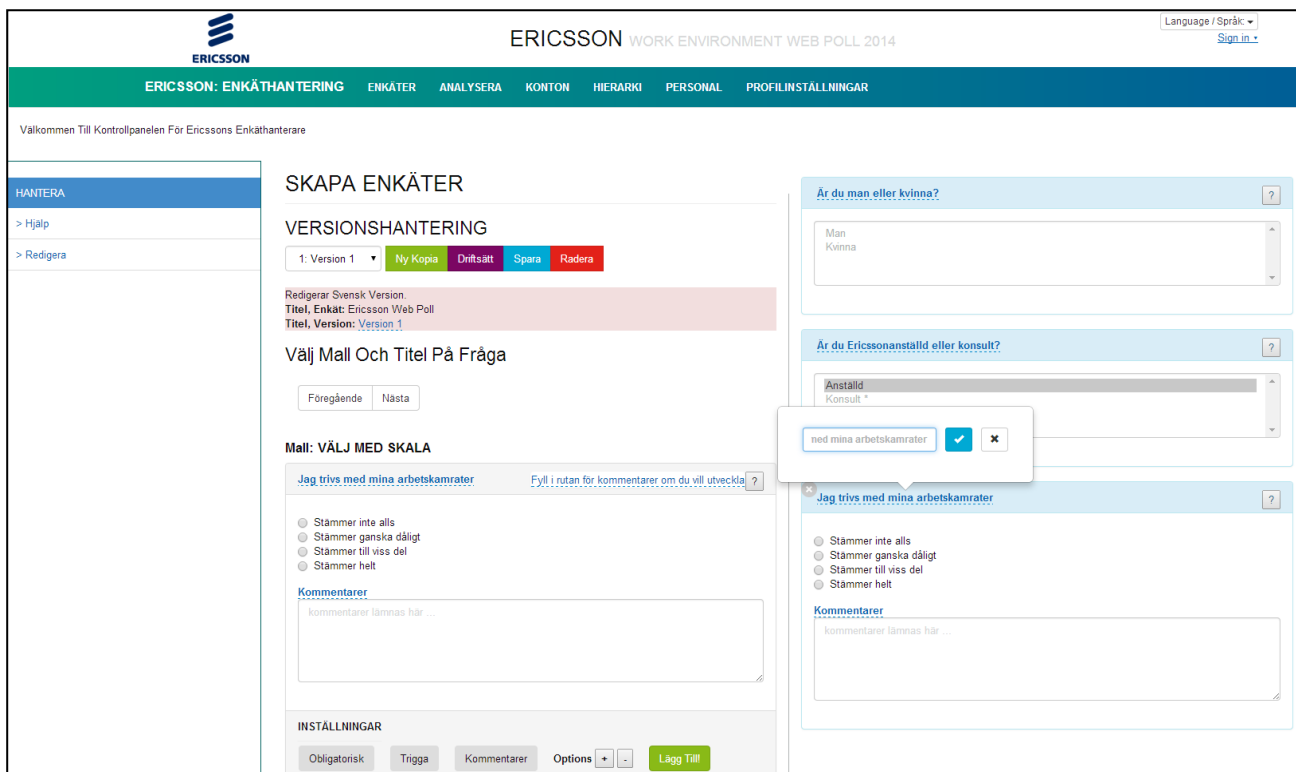
Följande rekommendationer från [20] har implementerats:

- Ljus bakgrund används, utan bakgrundsgrafik.
- Inget rutnätsystem används. Flerkolumnupplägg undviks.
- Förloppsindikatorer undviks.
- Standardkonventioner för typsnitt på frågor och svar.
- Inmatningsverktyg matchas mot funktionalitet, samt placeras till vänster om etiketter.
- Mest information har lagts på vänster sida.
- Frågor vänsterjusteras.
- Skalor är konsekventa.
- Bilder undviks.
- Definitioner som förklarar tekniska begrepp erbjuds.

Följande har inte implementerats, antingen på grund av tidsbrist eller att de inte önskats av kunden:

- E-post-funktionalitet för påminnelser och utskick.
- Sondering efter uteblivna svar och långsamt tempo.
- Design med sidindelning istället för scrollning.
- Automatisk kontrollräkning.

Det finns punkter som ligger utanför utvecklarens kontroll; om kunden implementerar frågor av typen ”Vet ej”, eller om enkäten kommer upplevas ha god säkerhet vid drift.



Figur 2: GUI. Gränssnitt för att redigera skapade enkäter i systemet. Till vänster mallhantering, till höger genererad hierarki.

¹⁴ Pollonium. <https://github.com/perjo927/Pollonium>

Figur 2 visar hur gränssnittet för att skapa och redigera enkäter ser ut. Till vänster kan administratören välja mellan olika mallar för att skapa frågor. Det går att göra frågor obligatoriska, det går att välja ut svarsalternativ som genererar fritextrutor, eller enbart bifoga en fritextruta. De olika mallarna som kunden önskat är flersvar, svarsalternativ med skala och fritext. Alla relevanta fält är redigerbara: Titel på enkäten, titel på enkätversionen, titel på frågan, hjälptext (eng. *tooltip*), svarsalternativ, med mera. Till höger i vyn kan administratören sedan rangordna frågor samt åter redigera dem eller ta bort dem. Kunden begärde även versionshantering, vilket innebär att man kan spara eller kopiera en enkät till en ny version. Man kan även redigera enkäten på i två olika språklägen: engelska och svenska.

Figur 3 visar hur administratören kan förhandsvisa en skapad enkät, för att bilda sig en uppfattning hur den ser ut från en tänkt respondents vy. Det går att återkoppla till punktlistan som följts genom att notera att den mesta informationen placerats till vänster, frågor är vänsterjusterade, bakgrunden är ljus, hjälptext erbjuds, inmatningsverktyg matchar funktionalitet, förloppsindikatorer undviks, skalor är konsekventa, med mera.

Utvärdering

Nedan utvärderas designen i systemet utifrån [20] samt

implementationens underhållsgrad utifrån [7].

Design

Av totalt 22 punkter som från kapitlet Teori har 12 stycken i realiteten tagits hänsyn till i implementationen, genom aktivt urval eller bortval av funktionalitet, baserat på rekommenderat tillvägagångssätt. De övriga 10 punkterna har antingen inte varit önskade av kunden och därmed inte aktuella, eller inte kunnat implementeras på grund av tidsbegränsningen för arbetet. De kan därför komma att implementeras i ett senare skede. Vilka punkter som har berörts listades under den föregående rubriken Design.

Det har inte gått att utvärdera effekt på svars kvalitet eller svarsfrekvens på enkäter som har genererats genom det implementerade systemet, eftersom driftsättningen av enkäten har placerats bortom omfattningen för arbetet. Därför stannar designutvärderingen vid konstaterandet att mer än hälften av rekommendationerna från kapitlet Teori har implementerats i praktiken.

Underhållbarhet

Andelen CoffeeScriptkällkod som motsvaras av kompilerad JavaScriptkod har räknats ut på ett genomsnitt av 1000 rader JavaScriptkod. Blankrader och kommentarer har inte räknats med. Resultatet visar att för varje 1000 rader JavaScriptkod har 707 rader CoffeeScriptkod skrivits. Det innebär att CoffeeScriptkod sparar in i storleksordningen 30

The screenshot shows the 'ERICSSON ENKÄTHANTERING' (Survey Management) interface. The top navigation bar includes 'ERICSSON ENKÄTHANTERING', 'ENKÄTER', 'ANALYSERA', 'KONTON', 'HIERARKI', 'PERSONAL', and 'PROFILINSTÄLLNINGAR'. The main content area displays a preview of a survey question: '1. Är du man eller kvinna?'. Below the question is a dropdown menu with options 'Man' and 'Kvinna'. The interface also shows a version selector '1: Version 1' and a 'HANTERA' sidebar with options for 'Hjälp' and 'Redigera'. The question is followed by two more preview questions: '2. Är du Ericssonanställd eller konsult?' and '3. Jag trivs med mina arbetskamrater', each with a help icon. A legend for the third question shows radio button options: 'Stämmer inte alls', 'Stämmer ganska dåligt', 'Stämmer till viss del', and 'Stämmer helt'.

Figur 3. Förhandsvy av genererad enkät.

% på totala antalet rader läsbar källkod. Det får anses vara en stor vinst, särskilt ju större systemet växer i storlek, och bedöms därmed med högsta betyg.

En cyklomatisk komplexitet på 8 % beräknades, vilket enligt [7] renderar i högsta betyg.

32 stora enheter per 66 totala enheter detekterades. De 66 enheterna innehåller dock anonyma funktioner, som också bör räknas som enheter i sig. Räknas anonyma funktioner med blir totalsumman 107 enheter. Det ger en 30-procentig risk, vilket renderar näst högsta betyg.

Inga kodkloner större än 6 rader har detekterats, vilket ger högsta betyg.

Ingen enhetstestning finns för systemet vilket ger lägsta betyg.

Bedömningskalan består av följande enheter:

(++), (+), (o), (-), (--)

där (++) är högst, (--) lägst.

För varje systemegenskap har ett medelbetyg räknats ut, för alla de källkodsegenskaper som haft ett kryss på motsvarande rad. Efter att uträkningen har gjorts har det gått att för varje systemegenskap avläsa den totala bedömningen i kolumnen längst till i höger i den sammanställda tabellen (se Tabell 2). Det går i tabellen direkt avläsa att systemet är i hög grad förändringsbart, till viss grad testbart och analyserbart, däremot inte stabilt i någon grad. Detta går att spåra tillbaka till avsaknaden av enhetstester.

Källkodsegenskaper

	Volym	Komplexitet per enhet	Duplicering	Enhetsstorlek	Enhetstestning	
	++	++	++	+	--	
Analyserbarhet	X		X	X	X	o
Förändringsbarhet		X	X			++
Stabilitet					X	--
Testbarhet		X		X	X	o

Tabell 2. Total bedömning av systemegenskaper.

DISKUSSION

I detta kapitel diskuteras först resultatet av arbetet, därefter metoden för att skapa systemets enkät-design samt metoden för att skapa underhållbarhet i systemet.

Resultat

Design

En design har implementerats med hänsyn till både Ericssons riktlinjer och rekommendationer för webbenkäter [20]. Kopplingen till punktlistan i kapitlet Teori torde vara tydliggörande för en bedömare, om implementationen följer god enkät-design eller inte. En sådan tydlig mappning, kompletterat med skärmdumpar, har även till viss grad skapat objektivitet i studien.

Den uteblivna utvärderingen av effekten av den implementerade designen på svars-kvalitet och svars-frekvens är en stor brist i studien och hade kunnat åtgärdas med ett större tidsomfång till arbetet, eller en snävare avgränsning av studiens område. Det kan konstateras att arbetet med designen i studien har resulterat i en fungerande tillämpning, men som inte går att utvärdera tydligt utifrån frågeställningen i kapitlet Teori.

Underhållbarhet

Det har gjorts avsteg från SIG-modellen [7] när den har behövt anpassning till studiens kontext. Det visar dels på att modellen har flexibilitet, vilket kan vara av godo, men nackdelen är att liknande studier kan bli svåra att jämförbara sinsemellan, särskilt om metriker justeras och bedöms godtyckligt. Exempelvis räknades enbart antal rader CoffeeScriptkod i studien. Att räkna antal rader Jadekod hade blivit problematiskt då koden som genereras inte är konstant, utan dynamisk.

Det kan ändå konstateras att ett tydligt resultat har kunnat presenteras för systemets underhållbarhet, vilket kan ses i Tabell 2, där orsak och verkan synliggörs.

Metod

Replikerbarhet

Metoden som har använts för att implementera enkät-designen i systemet – där en punktlista följs under implementeringsfasen och som följs upp efteråt – torde vara i hög grad replikerbar. Rekommendationerna som gavs av [20] var tydliga och därmed enkla att följa. De lämnar även utrymme för flexibilitet med hänsyn till tillämpningens kontext.

De programmeringsspråk som designen implementerats med har varit konventionella: CSS har kombinerats med HTML. Det bör inte orsaka svårigheter att replikera.

Metoden som har använts för att skapa god underhållbarhet i systemet är i ännu högre grad replikerbar, då det rör sig om en modell [7] med tydliga instruktioner för hur den ska tillämpas. Den kan användas under hela applikationens livscykel. Den vetenskapliga precisionen är hög, med en kartläggning mellan egenskaper på olika nivåer, vilket klarlägger orsak och verkan. Modellen är språkoberoende i avseendet att det är system- och källkodsegenskaper som analyseras, inte språkval. Detta är till fördel för replikerbarheten.

Reliabilitet

Reliabiliteten med avseende på enkätdesignen är till stor del beroende på om studien utformats med eller utan kund. Med en kund som tredje part tillkommer önskemål och företagsriktlinjer. Med självbestämmande kommer större kontroll. Svårigheten i denna studie låg i att balansera kundens önskemål med [20], samt Ericssons egna riktlinjer.

SIG-modellen [7] har visat sig enkel att förstå och tillämpa. Simplicitet är en nyckelfaktor eftersom det inte lämnar utrymme för tolkningskonflikter. Men även [7] har tillåtit fria val av metriker beroende på kontext, vilket gör att studier kan skilja sig åt beroende på val av metriker. Rapportens system implementerades som en webbapplikation, och de kan implementeras på många olika sätt och ändå utföra samma uppgifter. Den stora fördelen med de tabeller och betygsskalor som ges i [7] med en resulterande totalsumma är ändå att det går jämföra studier sinsemellan och på så vis öka reliabiliteten.

Validitet

Att det implementerade systemet är transparent och har öppen källkod torde vara välgörande för tillförlitligheten i studien. Källkoden kan då studeras och applikationens tillämpade design och interaktion går att kontrollera genom att exekvera den publikt tillgängliga applikationen med Node.js och Meteor.

Att effektiviteten i enkätdesignen inte har kunnat undersökas, eller att alla rekommendationer från [20] inte implementerats eller tagits hänsyn till, är förstås en brist i studien.

Till skillnad från vad [3] och [16] anser om det ineffektiva i att mäta underhållbarhet, menar författaren att det är värdefullt att kunna göra en utvärdering på systemnivå, kompletterat med källkodsanalys, som även [18] anser.

De uppmätta underhållbarhetsvärdena har till viss del färgats av en fri tolkning om hur de bäst bör utvärderas, det finns utrymme för mer exakthet i val av mätmetod. Cyklomatisk komplexitet [12] är exempelvis inte ett lättberäknat begrepp och det gäller att ett verktyg finns tillhanda för det valda webbspråket. [7] har även föreslagit att en del av den totala bedömningen ska justeras mot en tabell som beskriver hur produktivt ett språk är. Organisationen som publicerar tabellen ifråga rekommenderar dock inte längre tabellen och publicerar inte nya versioner¹⁵. Dyliga företeelser kan vara problematiska.

Källkritik

Det hade varit intressant att undersöka den reviderade ISO-standarderna för mjukvarukvalitet: ISO/IEC 25010:2011, för att komplettera teorienomgången i studien. ISO 9126 har fått mycket kritik i forskningen som undersökts, det är möjligt att den reviderade standarden har tagit tillvara på

kritiken. Organisationen TIOBE erbjuder exempelvis en kommersiell tjänst¹⁶ som undersöker kodkvalitet baserat på 8 metriker med utgångspunkt från ISO 25010.

Arbetet i ett vidare sammanhang

När ett webbaserat enkätssystem implementeras är det viktigt att kunna säkra anonymiteten för enkätens respondenter, vilket uppdragsgivaren betonat i kravspecifikationen. Att säkra systemet görs lämpligast genom att inte skriva egen kod för säkerhet, eftersom det är svårt att självständigt tänka ut och täppa till alla de säkerhetsluckor som kan uppstå. I det implementerade systemet har istället beprövade tilläggsbibliotek tillämpats, som använts i kritiska situationer. En hundra procentig säkerhetsnivå kan aldrig garanteras, men ansträngningar för detta har likväl gjorts.

SLUTSATSER

De två undersökta modellerna för design och underhållbarhet har implementerats och visat sig tillämpbara, men inte kunnat prövas fullt ut.

Återkoppling till syfte och frågeställningar

Frågan hur ett webbenkätssystem kan designas för att ta maximerad svars kvalitet och svarsfrekvens lämnas till viss del obesvarad. En del av svaret ligger i att tillämpa rådande forskning, vilket har gjorts i studien, men utvärderingen av designens resulterande effekter på respondenter har uteblivit.

Frågan hur ett webbenkätssystem kan implementeras för att få en hög grad av underhållbarhet har besvarats med att en modell kan appliceras – i detta fall [7] – som klarlägger orsak och verkan och används genom utvecklingscykeln. Genom att upptäcka problem på ett tidigt stadium och omfaktorer koden löpande påverkas underhållbarhets-egenskaper positivt. I rapportens arbete skulle design av enhetstester på ett tidigt stadium haft positivt inflytande, vilket modellen klargjorde.

Konsekvenser

Utvärderingen som gjordes i samband med resultatanalysen gjorde att behovet av en omfattande omstrukturering av systemet upptäcktes. Det ledde till en diskussion med uppdragsgivaren där det beslutades att en iteration skulle avsättas till omfaktorer av källkoden. Därmed kunde modellen [7] som prövades i studien komma till praktisk nytta, vilket tyder på en styrka i den. Möjligtvis kan andra som prövar modellen komma att dra liknande erfarenheter.

Framtida arbete

Det skulle ha varit intressant att studera effekten av enkätdesignen på svars kvalitet och svarsfrekvens. Enkätens planerade driftsättning ligger dock bortom räckvidden för

¹⁵ Software Productivity Research. <http://www.spr.com/>

¹⁶ The TIOBE Quality Indicator
<http://www.tiobe.com/content/paperinfo/TIOBEQualityIndicator.pdf>

arbetet i rapporten. Effekten skulle kunna utvärderas med exempelvis A/B-tester, där en testgrupp får den avsedda designen och en kontrollgrupp får en design som går emot riktlinjer. Med de omfattande förberedelser som krävs under sådana omständigheter skulle det kunna vara aktuellt för arbeten på masternivå snarare än kandidatnivå.

Det skulle även vara intressant om metriker för källkodsanalys specifika för webbapplikationer togs fram. Metrikerna i [7] har inte ursprungligen tagits fram för det ändamålet. Ett exempel är dynamiskt genererad kod som skapas av mallsystem (eng. *templating engines*) för HTML-dokument. Metriken för att räkna antal rader kod i [7] tar inte hänsyn till kod som dynamiskt genererar kod.

REFERENSER

1. Antonellis, P., Antoniou, D., Kanellopoulos, Y., Makris, C., Theodoridis, E., Tjortjijis, C., & Tsirakis, N. (2007). A data mining methodology for evaluating maintainability according to ISO/IEC-9126 software engineering–product quality standard. *Special Session on System Quality and Maintainability SQM2007*.
2. Bouwers, E., Visser, J., & Van Deursen, A. (2009, September). Criteria for the evaluation of implemented architectures. In *Software Maintenance, 2009. ICSM 2009. IEEE International Conference on* (pp. 73-82). IEEE.
3. Broy, M., Deissenboeck, F., & Pizka, M. (2006, May). Demystifying maintainability. In *Proceedings of the 2006 international workshop on Software quality* (pp. 21-26). ACM.
4. Coleman, D., Ash, D., Lowther, B., & Oman, P. (1994). Using metrics to evaluate software system maintainability. *Computer*, 27(8), 44-49.
5. Conrad, F. G., Couper, M. P., Tourangeau, R., & Peytchev, A. (2010). The impact of progress indicators on task completion. *Interacting with computers*, 22(5), 417-427. Schwartz, M. *Guidelines for Bias-Free Writing*. Indiana University Press, Bloomington, IN, USA, 1995.
6. Fan, W., & Yan, Z. (2010). Factors affecting response rates of the web survey: A systematic review. *Computers in Human Behavior*, 26(2), 132-139.
7. Heitlager, I., Kuipers, T., & Visser, J. (2007, September). A practical model for measuring maintainability. In *Quality of Information and Communications Technology, 2007. QUATIC 2007. 6th International Conference* (pp. 30-39). IEEE.
8. Johnson, P., Lagerström, R., Närman, P., & Simonsson, M. (2007). System quality analysis with extended influence diagrams. In *CSMR 2007 Workshop and Special Session papers*.
9. Joinson, A. N., & Reips, U. D. (2007). Personalized salutation, power of sender and response rates to Web-based surveys. *Computers in Human Behavior*, 23(3), 1372-1383.
10. Kask, T. (2012). Indication of user-driven progress. An evaluation of the impact of progress indicator design on user experience and performance. Department of Applied Information Technology, Chalmers University of Technology.
11. Luijten, B., Visser, J., & Zaidman, A. (2010, March). Faster defect resolution with higher technical quality of software. In *Proceeding of the 4th International Workshop on Software Quality and Maintainability (SQM'10), Madrid (España), 15-18 de marzo de 2010* (pp. 11-20).
12. McCabe, T. J. (1976). A complexity measure. *Software Engineering, IEEE Transactions*, (4), 308-320.
13. Morton, S., Bandara, D. K., Robinson, E. M., & Carr, P. E. A. (2012). In the 21st Century, what is an acceptable response rate? *Australian and New Zealand journal of public health*, 36(2), 106-108.
14. Muñoz-Leiva, F., Sánchez-Fernández, J., Montoro-Ríos, F., & Ibáñez-Zapata, J. Á. (2010). Improving the response rate and quality in Web-based surveys through the personalization and frequency of reminder mailings. *Quality & Quantity*, 44(5), 1037-1052.
15. Myers, B. A. (1985, April). The importance of percent-done progress indicators for computer-human interfaces. In *ACM SIGCHI Bulletin* (Vol. 16, No. 4, pp. 11-17). ACM.
16. Riaz, M., Mendes, E., & Tempero, E. (2009, October). A systematic review of software maintainability prediction and metrics. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement* (pp. 367-377). IEEE Computer Society
17. Scott, B., & Neil, T. (2009). *Designing web interfaces: Principles and patterns for rich interactions*. O'Reilly Media, Inc.
18. Sjöberg, D. I., Anda, B., & Mockus, A. (2012, September). Questioning software maintenance metrics: a comparative case study. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement* (pp. 107-110). ACM.
19. Tidwell, J. (2010). *Designing interfaces*. O'Reilly Media, Inc.
20. Tourangeau, R., Conrad, F., Couper, M. (2013, Maj). *The Science of Web Surveys*. Oxford Press.
21. Villar, A., Callegaro, M., & Yang, Y. (2013). Where Am I? A Meta-Analysis of Experiments on the Effects of Progress Indicators for Web Surveys. *Social Science Computer Review*, 31(6), 744-762.

BILAGA 1

Kravspecifikation

- Årlig enkätundersökning kring arbetsmiljöfrågor
- Implementeras i webbgränssnitt
- Flexibla enkätfrågor som kan ändras över tid
- Anonym datainsamling
- Automatiska påminnelser om respondenter inte svarar
- Datasammanställning med avseende på organisationstillhörighet (sektion/avdelning/site)
- En minsta svarsfrekvens som villkor för resultatdriftsättning
- Enkel driftsättning av resultatpresentation
- Grafisk presentation på olika organisationsnivåer (för både utskrift och projektor)
- Fritext kopplad till enkätfrågor
- Sektionschefer får sin egen presentation, samt överliggande (egen avdelning och site)
- Avdelningschefer får sin egen presentation samt underliggande (egna sektioner) och överliggande (site)
- Åtkomst av presentationer styrt med lösenord / behörighetsnivå
- Chefer bör kunna skriva handlingsplan med åtgärder
- Röstning ska enbart kunna ske 1 gång

BILAGA 2

Enkätöversikt i det implementerade systemet

Language / Språk: [Sign in](#)

ERICSSON WORK ENVIRONMENT WEB POLL 2014

WEB POLL MANAGEMENT SURVEYS ANALYTICS ACCOUNTS HIERARCHY PERSONNEL PROFILE SETTINGS

Welcome To The Ericsson Web Poll Management

MANAGE

- > Help
- > Edit

Edit Surveys

List Of Available Surveys.

New Survey Import New Survey

Label	Year	Versions	Created	Status	Actions
Ericsson web poll	2014	3	Sun, 01 Jun 2014 09:54:31 GMT	OFFLINE	Lock Delete Archive Edit Preview
Experimental web poll	2014	1	Sun, 01 Jun 2014 12:22:52 GMT	OFFLINE	Unlock Delete Archive Edit Preview
Unused web poll	2014	1	Mon, 02 Jun 2014 17:54:35 GMT	OFFLINE ARCHIVED	Preview

Column 1 Label
Li 1

Column 2 Label
Li 1

Column 3 Label
Li 1

Column 4 Label
Li 1

En kompletterande skärmdump som visar en översiktlig administrationssida för alla samlade enkäter i systemet. Det går att importera och skapa nya enkäter. Det går att låsa enkäter (och alla dess versioner) för redigering, ta bort enkäter, arkivera dem (för att kunna jämföra mellan årgångar), förhandsvisa enkäter samt öppna dem för redigering. Det går också att se driftsättningsstatus.



På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extra-ordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: <http://www.ep.liu.se/>

© Per Jonsson