Google Analytics som verktyg för användbarhetsutvärdering av mobila applikationer

Niklas Sjöberg & Filip Svensson

2015

Master's Thesis

Department of Design Sciences
Lund University

Abstract

In recent years, the market for mobile applications has continuously grown in size, and appears to continue to grow. This means greater competition among app developing companies as the users expect applications with high usability and a great user experience. To assess these properties the companies can perform usability evaluations. However, practicing traditional usability evaluation methods can be both expensive and time consuming. Thus, they may not be feasible for all companies.

This thesis explores the potential for using Google Analytics as a tool to identify usability issues as well as to evaluate user experience by looking at user data from three mobile applications, all with similar functionality. This included identifying which usability metrics that were both relevant in this context and could be applied to the data collected by Google Analytics. It is generally recommended that you combine two or more usability evaluation methods to get the best results. Therefore, a heuristic evaluation and user testing was also conducted. The results from Google Analytics were compared to the results from the two evaluation methods.

An extensive literature study resulted in five usability metrics; task success, time-on-task, errors, efficiency, and learnability. These were applied to the data collected by Google Analytics. As a whole, the results provided insufficient evidence to recommend Google Analytics as a tool for a comprehensive usability evaluation. However, there were indications of situations were the tool could prove useful. Both heuristic evaluation and user testing identified usability issues that were not detected by Google Analytics. As a result, it is recommended that either of the two methods should serve to supplement the results from Google Analytics.

Sammanfattning

Under de senaste åren har marknaden för mobila applikationer kontinuerligt vuxit i storlek, och den fortsatta utvecklingen pekar i samma riktning. Således ökar konkurrensen mellan app-utvecklande företag då användarna förväntar sig applikationer med hög användbarhet och en bra användarupplevelse. För att utvärdera just dessa egenskaper kan företagen utföra användbarhetsutvärderingar. Detta är dock inte genomförbart för alla företag då traditionella metoder kan vara både kostsamma och tidskrävande.

Detta examensarbete undersöker möjligheterna att med hjälp av Google Analytics identifiera användbarhetsbrister i mobila applikationer samt utvärdera dess användarupplevelse. Detta innefattade att identifiera usability metrics som kunde appliceras på användardata insamlad med hjälp av Google Analytics. I studien användes tre mobila applikationer med liknande funktionalitet. Tidigare studier har visat att man bör kombinera två eller fler metoder för användbarhetsutvärdering för att uppnå bästa resultat. Därför utfördes också en heuristisk utvärdering samt användartester vars resultat jämfördes med resultaten från Google Analytics.

En omfattande litteraturstudie ledde till fem stycken usability metrics: task success, time-on-task, errors, efficiency och learnability. Dessa applicerades på insamlad användardata från Google Analytics. Sammantaget pekade resultatet på att det fanns situationer då Google Analytics kan visa sig vara användbart, men gav inte tillräckligt med stöd för att rekommendera verktyget för en omfattande användbarhetsutvärdering. Både heuristisk utvärdering och användartester identifierade användbarhetsbrister i applikationerna som Google Analytics inte kunde påvisa. Därav är det rekommenderat att Google Analytics används i kombination med någon av dessa metoder.

Förord

Detta examensarbete har genomförts vid institutionen för designvetenskaper på Lunds Tekniska Högskola i samarbete med Smart Refill under våren 2015.

Vi skulle vilja rikta ett stort tack till Smart Refill, och framförallt vår handledare Jesper Ekberg, för all hjälp samt möjligheten att utföra examensarbetet på företaget.

Från Lunds Tekniska Högskola vill vi tacka vår handledare Joakim Eriksson för hans råd och feedback under arbetets gång.

Slutligen vill vi tacka Fredrik Lindholm för att han ställde upp och agerade utvärderare under den heuristiska utvärderingen samt alla personer som tog sig tid och deltog i användarstudien.

Innehåll

Figurer

Tabeller

1 Introduktion

Detta examensarbete är utfört hos Smart Refill AB som utvecklar och driver smarta mobila tjänster. De tar hand om hela utvecklingsprocessen från början till slut och har höga krav på användbarhet. Med hjälp av Google Analytics har de under en tid samlat på sig en stor mängd användardata från deras applikationer. Denna studie ämnar att undersöka hur insamlad användardata kan användas för att utvärdera användbarheten hos mobila applikationer.

1.1 Bakgrund

Marknaden för mobila applikationer växer ständigt. Google Play och iTunes App store är de två största marknadsplatserna och enligt [1] fanns det i slutet av 2014 över 1,2 miljoner applikationer på respektive marknadsplats. Denna siffra ser ut att fortsätta öka och i takt med det så ökar också konkurrensen mellan app-utvecklande företag. Det blir allt viktigare att skapa applikationer som fungerar och ger en bra användarupplevelse. Att användbarhet är en viktig del för att uppnå detta är känt sedan länge.

En applikations användbarhet kan utvärderas på flera olika sätt där de vanligaste metoderna är att antingen utföra tester med riktiga användare eller låta en användbarhetsexpert utvärdera applikationen. Dessa metoder har sina för- och nackdelar. Det är inte säkert att det finns en användbarhetsexpert inom företaget och att hyra in en kan vara kostsamt. Att utföra tester med användare kan vara tidskrävande och det är inte helt lätt att hitta rätt testpersoner. Därmed lämpar sig inte dessa metoder för alla företag vilket gör att det finns ett behov av billigare och mindre tidskrävande metoder.

Det finns idag flertalet verktyg som kan samla in användardata och presentera statistik över hur en mobil applikation används. Det populäraste verktyget är Google Analytics som finns implementerat i ungefär 12,5 % av alla

Android-applikationer på Google Play [2]. Data som samlas in är bland annat antalet användare, varifrån användarna kommer, vilka skärmar de besöker, hur lång tid de spenderar i applikationen med mera. Med Google Analytics så studeras användarna indirekt och alla som använder applikationen bidrar med data. Detta leder till lägre kostnader och mindre tidsåtgång i jämförelse med mer traditionella utvärderingsmetoder då det enda som krävs är att automatiskt insamlad data analyseras i efterhand. En av nackdelarna med denna typ av datainsamlingmetod är att insamlad data endast är kvantitativ. Därmed går det inte att ta del av användarnas subjektiva åsikter kring användarupplevelsen, vilket tester med användare kan bidra med.

Google Analytics och liknande verktyg har länge använts av marknadsavdelningar för att se var användare kommer ifrån och vilka som uppfyller utsatta mål, men få studier har gjorts kring möjligheterna att utvärdera en applikations användbarhet med hjälp av Google Analytics (och liknande verktyg för den delen).

1.2 Syfte och mål

Syftet med examensarbetet är att undersöka möjligheterna till att använda Google Analytics för att utvärdera användbarheten och användarupplevelsen i mobila applikationer. Ett av målen med projektet var att ta fram metoder för att med hjälp av Google Analytics kunna ta fram värden för att beräkna användbarhet. Dessa värden skulle vara enkla och lättbegripliga i den meningen att man inte nödvändigtvis skulle behöva vara en användbarhetsexpert för att kunna utföra beräkningarna och tolka resultatet.

Följande är projektets frågeställningar:

- Vilka slutsatser går att dra kring användbarhet och användarupplevelse med hjälp av användardata från Google Analytics?
- Är insamlad data tillräcklig för att kunna dra värdefulla slutsatser eller behöver den komplementeras med andra metoder och i så fall vilka?

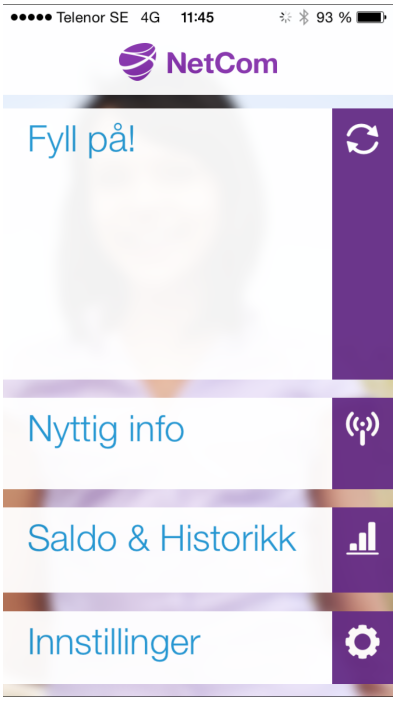
Det fanns aldrig några förhoppningar om att kunna göra en fullständig utvärdering av en mobil applikation endast med hjälp av Google Analytics. Målet var i stället att presentera metoder för att på ett enkelt och billigt sätt kunna ta fram tydliga indikationer på användbarhetsbrister, och därefter undersöka vilka andra metoder som kan användas för att vid behov komplettera de resultat som ges av Google Analytics.

1.3 Beskrivning av applikationerna

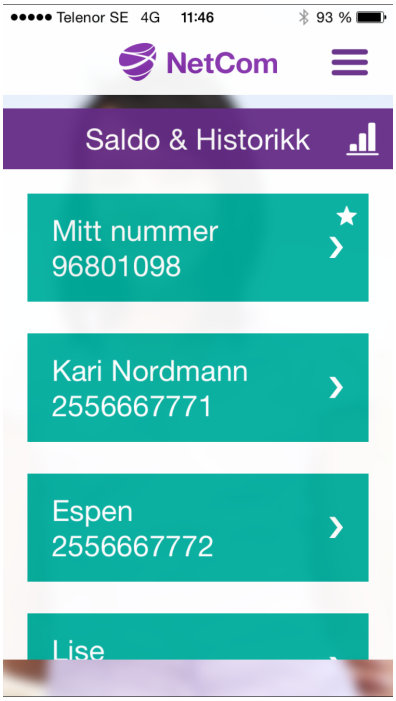
De tre mobila applikationerna som använts i fallstudierna är Ladda Refill, Netcom Påfyll och 3Fyll På. De tillhör alla samma kategori av applikationer och samtliga erbjuder funktionalitet för att ladda ett eller flera kontantkort. De har alla liknande laddningspremier som innefattar både samtal, meddelande och mobilsurf. Dessutom finns det funktionalitet för att kontrollera saldo på ett eller flera kontantkort.

Figur 1.1: Startskärmen för Ladda Refill.

Figur 1.2: Skärmen för Mina nummer i Ladda Refill.

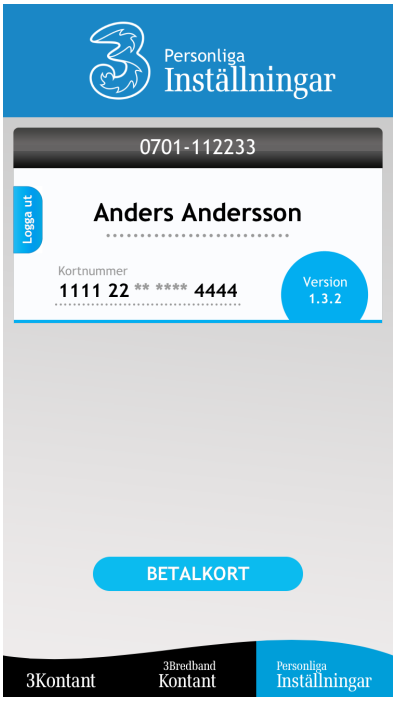


Figur 1.3: Startskärmen för Netcom Påfyll.



Figur 1.4: Skärmen för Mina nummer i Netcom Påfyll.

Figur 1.5: Startskärmen för 3Fyll På.



Figur 1.6: Skärmen för Mina nummer i 3Fyll På.

1.4 Fokus och avgränsningar

Tidsramarna för studien räckte inte till för att undersöka flera olika typer av applikationer. Därmed utfördes tre fallstudier som behandlar tre mobila applikationer av samma typ, alla utvecklade av Smart Refill. Resultatet är därför av begränsad generalitet. Ambitionerna har dock varit att försöka ta fram så allmänna metoder som möjligt. Främst om hur användardata insamlad med hjälp av Google Analytics kan användas vid utvärdering av mobila applikationer.

1.5 Arbetets upplägg

Examensarbetet och dess arbetetsgång delades upp i fem faser, se figur 1.7. Rapportens upplägg följer till stor del arbetsgången. Notera att även om rapporten är skriven på svenska så används engelska termer på flera ställen, detta för att det i sammanhanget saknas en vedertagen eller lämplig översättning till det svenska språket.

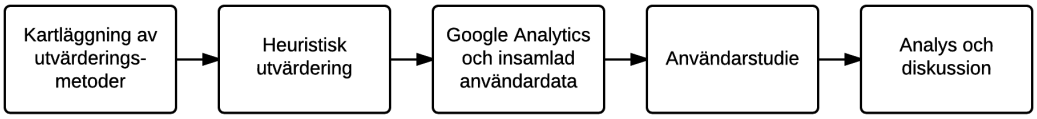
Kartläggning av utvärderingsmetoder
: Kartläggningen utfördes för att hitta relevanta teorier inom användbarhet, användarupplevelse och metoder för användbarhetsutvärdering av mobila applikationer som arbetet kunde baseras på. Metodik och resultat för denna fas presenteras i kapitel 3.

Heuristisk Utvärdering
: Den heuristiska utvärderingen av applikationerna påbörjades innan studien av Google Analytics och insamlad användardata. Den tjänade två syften, dels tog den fram teorier om möjliga områden i applikationerna där användbarhetsproblem kunde uppstå, och dels för att undersöka om heuristisk utvärdering är ett bra komplement till Google Analytics. Metodik och resultat för denna fas presenteras i kapitel 4.

Google Analytics och insamlad användardata
: I dennas fas studerades implementationen av Google Analytics i applikationerna samt hur och vilken användardata som kunde utläsas. Besluten om vilken typ av data som var relevant för att utvärdera applikationernas användbarhet baserades på resultaten från kartläggningen i fas ett. Metodik och resultat för denna fas presenteras i kapitel 5.

Användarstudie
Precis som med den heuristiska utvärderingen var syftet med användarstudien att undersöka hur väl den kompletterade resultatet från studien av Google Analytics. Metodik och resultat för denna fas presenteras i kapitel 6.

Analys och diskussion
Den sista fasen av examensarbetet. Här analyseras, jämförs och diskuteras resultaten från föregående faser. Även arbetets metodik diskuteras. Resultatet av detta arbete presenteras i kapitel 7.



Figur 1.7: Arbetets upplägg

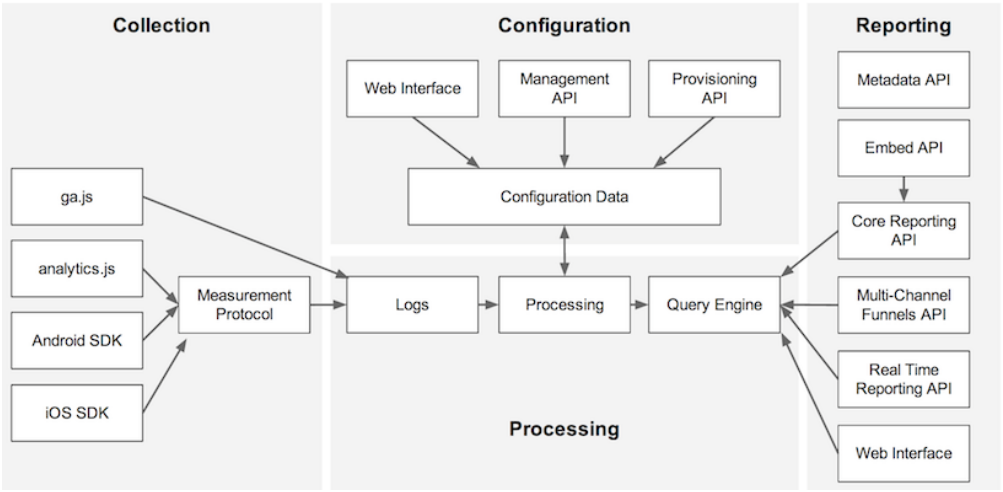
1.6 Arbetets upplägg

Under arbetet med examensarbetet har författarna ständigt arbetat sida vid sida. Det har varit en ständig dialog mellan författarna och samtliga beslut har varit ömsesidiga. Båda författarna har arbetat gemensamt under alla fem faser som beskrivs i avsnitt 1.5 och har en text skrivits av en av författarna har den alltid granskats av medförfattaren.

2 | Introduktion till Google Analytics

Google Analytics är ett analysverktyg som spårar hur användare interagerar med innehållet på en webbsida, en mobil webbsida eller i en mobil applikation. Det finns två versioner av Google Analytics, en gratisversion och en premiumversion. Premiumversionen är en utvidgning av gratisversionen och är mer lämpad för stora företag. I gratisversionen lagras upp till tio miljoner träffar i månaden som sparas totalt tjugofem månader bakåt i tiden. Premiumversionen lagrar istället upp till en miljard träffar i månaden och data sparas tre år bakåt i tiden, dessutom ger den tillgång till Googles support.

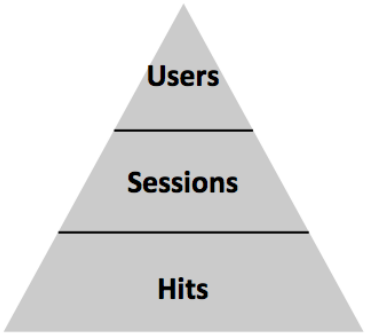
Google Analytics består av fyra stycken huvudkomponenter; insamling, konfiguration, bearbetning och rapportering. Insamlingskomponenten samlar in data över hur användarna interagerar med systemet. Dessa data sparas i loggar som behandlas av bearbetningskomponenten med hänsyn till de inställningar som analytikern (användaren av Google Analytics) gjort i konfigurationskomponenten. Färdigbehandlad data presenteras sedan i rapporter med hjälp av rapporteringskomponenten [3].



Figur 2.1: Figur över de fyra huvudkomponenterna i Google Analytics. Bilden är hämtad från https://developers.google.com/analytics/images/platform/platformOverview.png (11 mars 2015).

2.1 Användare, sessioner och träffar

Spårningsdata som kommer från hemsidor, mobila hemsidor eller mobila applikationer kan delas in i en allmän hierarki bestående av användare, sessioner och träffar [4], se bild 2.2.



Figur 2.2: Den allmänna hierarkin av användare, sessioner och träffar. Bilden är hämtad från http://cutroni.com/blog/2014/02/05/understanding-digital-analytics-data/ (11 mars 2015).

2.1.1 Träffar

En träff är den mest granulära typen av data i ett analysverktyg. För varje träff skickas data från användaren till Google Analytics. Det finns många olika typer

av träffar att samla in beroende på vilket analysverktyg som används. Nedan beskrivs de vanligaste typerna av träffar i Google Analytics.

Sidvisningar/skärmvisningar - En sidvisning (skärmvisning för mobila applikationer) mäts när en användare tittar på eller besöker en webbsida på webbplatsen. En sidvisning är en av de mest grundläggande parametrarna inom digital analytics och kan bland annat användas för att se vilka sidor som besökts mest, beräkna antalet sidvisningar per besök och genomsnittlig tid på en specifik sida. Sidvisningar är automatiskt implementerade i Google Analytics.

Händelser - En händelse inträffar när en användare interagerar med en interaktiv komponent på hemsidan eller i den mobila applikationen, det kan till exempel vara när en användare klickar på en knapp eller en länk. Händelser används för att mäta hur ofta användare utför en viss handling och måste implementeras manuellt på hemsidan eller i applikationen av utvecklarna.

Händelser kan delas in i fyra stycken komponenter:

- **Kategori** - ett namn som används för att gruppera objekt. Vanligtvis används samma kategorinamn flera gånger för relaterade element i gränssnittet som ska grupperas under samma kategori.
- **Åtgärd** - används normalt för att ange vilken typ av händelse eller interaktion som ska spåras för ett specifikt objekt.
- **Etikett** - även kallat händelsenamn, används för att ge ytterligare information om händelser som ska spåras. Både kategori och åtgärder måste anges för händelser medan etiketter är valfria.
- **Värde** - den sista komponenten och är precis som etiketter valfri att använda för händelser. Till skillnad från föregående kategorier är värde ett heltal och inte en sträng, och används således för att tilldela händelser ett numeriskt värde [5].

För att förtydliga hur ovanstående komponenter kan användas ges följande exempel: Google Analytics finns implementerat i en mobila applikation som bland annat kan spela upp ett antal olika videor, samt att det i videospelaren finns möjlighet att starta eller stoppa videon. För att registrera händelser kring hur användarna interagerar med varje enskild video skulle en implementation av händelser kunna se ut på på följande sätt:

- **Kategori:** video
- **Åtgärd:** spela upp
- **Etikett:** namnet på videon som spelas upp

- **Värde:** tiden det tog att ladda videon

E-handelstransaktioner - Med e-handelstransaktioner går det att mäta antalet transaktioner som genomförts och dess intäkter. Precis som händelser är detta något som utvecklarna manuellt måste implementera på hemsidan eller i applikationen [4].

2.1.2 Sessioner

En session är en period av användarinteraktion och kan ses som en samling av träffar som kommer från samma användare. När Google Analytics upptäcker att en användare inte längre är aktiv så avslutas sessionen och nästa gång användaren blir aktiv så startas en ny session. I standardinställningnarna för hemsidor så avslutas sessionen när en användare varit inaktiv i 30 minuter. För mobila applikationer är denna tid 30 sekunder. Det finns möjlighet för utvecklarna att ändra denna tid till en som passar den specifika hemsidan eller mobila applikationen. Tiden mellan den första träffen och den sista träffen för en session beräknas som den totala sessionstiden [4].

2.1.3 Användare

I Google Analytics är alla användare anonyma och "identifieras" med hjälp av ett anonymt nummer eller en anonym sträng som skapas första gången en användare upptäcks, det vill säga om det inte redan finns en identifierare på användarens enhet. Den anonyma identifieraren skickas med varje träff till Google Analytics och på så sätt kan träffar grupperas och skapa sessioner [4]. Hur den anonyma identifieraren lagras hos användaren skiljer sig en del mellan hemsidor och mobila applikationer, vilket beskrivs i avsnitt 2.2.

2.2 Spårning i Google Analytics

Google Analytics kan samla in data från flera olika plattformar med hjälp av olika typer av spårningstekniker. Spårning av webbsidor skiljer sig en del från spårning i mobila applikationer. Nedan beskrivs kortfattat hur det fungerar i de två fallen.

2.2.1 Spårning av webbsidor

De två vanligaste metoderna för att samla in data från webbplatser är antingen via sidtaggar eller via loggfiler på serversidan. En sidtagg är JavaScript-kodavsnitt som placeras på alla webbsidor på en webbplats. Sidtaggen samlar in data via besökarens webbläsare och skickar sedan dessa till en avlägsen server. Denna teknik kallas för ”client-side data collection”. Den andra tekniken innebär att en webbserver samlar in data genom att logga dess egna aktivitet och spara data i loggfiler. Detta kallas för ”server-side data collection” [6].

Google Analytics använder sig främst av den förstnämnda tekniken och sidtaggen brukar kallas för Google Analytics Tracking Code (GATC). Som nämndes ovan måste GATC inkluderas på alla webbsidor på webbplatsen som ska spåras. När en besökare ansluter till en webbsida med Google Analytics installerat så identifierar GATC ett antal attribut så som besökarens anonyma identifierare, geografiska position, vilken typ av webbläsare som används, vilket operativsystem, hur många gånger besökaren tidigare varit på sidan med mera. GATC antingen skapar eller uppdaterar (beroende på om användaren besökt sidan tidigare) också ett antal förstaparts-cookies som lagras lokalt. I dessa cookies lagras information om besökaren som bland annat innehåller den anonyma identifieraren. Dessa data skickas sedan till Google Analytics servrar i form av en sidvisning som indikerar att en besökare har besökt en viss webbsida. Google Analytics har även stöd för andra typer av data som händelser (används för att spåra en viss händelse, till exempel en knapptryckning), transaktioner (används för att spåra e-handelstransaktioner) med mera [7].

Idag kan större delen av de smartphones som finns på marknaden spåras med hjälp av GATC på samma sätt som beskrevs ovan. Dock finns det några undantag, äldre telefoner som inte kan hantera varken JavaScript eller cookies måste spåras på ett annat sätt. Google Analytics implementeras då istället på serversidan och låter servern logga dess egen aktivitet, detta medför att det endast går att spåra sidvisningar [6].

2.2.2 Spårning av mobila applikationer

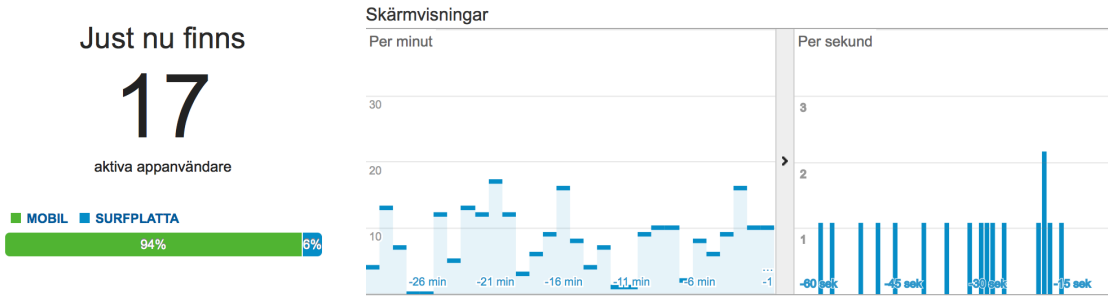
Att spåra användandet i mobila applikationer fungerar inte riktigt på samma sätt som att spåra webbsidor men det finns vissa likheter. Precis som i webbspårning så lagras en anonym identifierare på enheten, men istället för att använda sig av cookies så sparas data i en databas. I grund och botten fungerar databasen likadant som en cookie, den skapas först när användaren installerar den mobila applikationen på sin enhet och tas bort när användaren tar bort applikationen

[4]. Eftersom mobila applikationer inte renderar några HTML-sidor går det inte att lägga till GATC i applikationen för att samla in data. Istället går det med hjälp av Google Analytics software development kits (SDK) att skapa virtuella sidvisningar och händelser. Det finns i dagsläget två versioner av SDK, en till Android och en till iOS [6].

Det är inte säkert att användare alltid har en internetuppkoppling när de interagerar med den mobila applikationen och i Google Analytics går det att spåra användare även om de är offline. All data som sparas när en användare är offline läggs i en kö och nästa gång användaren har en internetuppkoppling så skickas all data i kön på en och samma gång. Tidsstämpeln för data i kön blir den tid som dessa skickas till Google Analytics (nästa gång användaren får en internetuppkoppling) och inte tiden för när dessa spårades [8].

2.3 Realtidsfunktionen

Realtidsrapporter gör det möjligt att bevaka aktiviteten på en webbplats eller i en mobil applikation med bara några sekunders fördröjning. Rapporterna uppdateras löpande. Detta gör det möjligt att se hur många personer som för tillfället besöker webbplatsen, vart de befinner sig, vilka sidor de tittar på, vilka händelser de utlöser, och vilka målkonverteringar som har skett. Se figur 2.3 för en exempelvy. Mobilträffar bearbetas i grupp för att spara batteritid. Det medför att det kan förekomma fördröjningar. Bearbetningen sker vanligtvis inom några minuter. [9]



Bästa appversioner:

	App-version	Aktiva användare
1.	2.1.2	9
2.	2.1.1	8

Bästa aktiva skärmar:

	Namn på aktiv skärm	Aktiva användare	
1.	Phones	6	35,29%
2.	Amounts	3	17,65%
3.	TopUpPIN	3	17,65%
4.	CustomerLogIn	2	11,76%
5.	TopUpPhone	2	11,76%
6.	CustomerRegistration	1	5,88%

Figur 2.3: Överblick över realtidsfunktionen. Visar antal aktiva användare, vilka skärmar de besöker för tillfället samt vilka appversioner de använder.

3 | Kartläggning av utvärderingsmetoder

3.1 Syfte och metod

I början av arbetet utfördes en omfattande kartläggning med fokus på användbarhet, användbarhetsprinciper, användarupplevelse och metoder för analytisk utvärdering av mobila applikationer. Ett av syftena med kartläggningen var att definiera och skapa förståelse för nyckelbegrepp inom dessa ämnen, det andra syftet var att få insikt kring tidigare utförd forskning. Detta gjordes i förstahand genom att konsultera konferensartiklar och vetenskapliga artiklar.

Kartläggningen rymmer följande frågeställningar:

- Vad innebär användbarhet? Vilka områden inom användbarhet är relevanta för mobila applikationer?
- Vilka manuella metoder finns för att utvärdera användbarhet och användarupplevelse?
- Vilka principer har tidigare tagits fram inom ämnena användbarhet och gränssnittsdesign?
- Vilka utmaningar finns det vid användbarhetsutvärdering av mobila applikationer?
- Har det tidigare gjorts forskning på utvärdering av användbarhet med hjälp av Google Analytics eller liknande analysprogram?
- Vilka möjligheter finns det att mäta användbarhet med hjälp av kvantitativ data?

En central parameter för urval av artiklar och böcker har varit att välja litteratur som ofta citeras i annan litteratur, då detta speglar hur erkänd en viss studie är

inom det givna området. Om nypublicerade artiklar hittats har beskrivningen av dessa granskats så ingående och så objektivt som tiden tillåtit, vilket är en bra strategi enligt Bell [10].

Relevanta delar av resultatet från kartläggningen samt svaren på frågeställningarna presenteras under avsnitt 3.2 i detta kapitel.

3.2 Resultat

3.2.1 Användbarhet

Användbarhet är ett kvalitetsattribut som beskriver hur lätt det är för en användare att interagera med ett användargränssnitt. ISO har tagit fram en standardisering (ISO 9241-11) där användbarhet definieras som:

> *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [11]*

Medan Jakob Nilsen i sin bok "Usability Engineering" [12] definierar fem kvalitetskriterier för att utvärdera hur bra användbarhet en produkt har.

Learnability - Hur lätt är det för förstagångsanvändare att utföra enklare uppgifter i systemet?

Efficiency - När användarna väl lärt sig hur systemet fungerar, hur lätt är det då för dem att utföra uppgifter?

Memorability - Hur lätt är det för användaren att komma ihåg hur man använder systemet, även om man inte använt det på ett tag?

Errors - Hur många fel gör en användare, hur allvarliga är felen, och hur lätt är det att rätta till dem?

Satisfaction - Hur tillfredställande är det att använda systemets design?

3.2.2 Användarupplevelse

Användarupplevelsen syftar till intrycket användare får under tiden som systemet används. Detta är ett totalintryck som inte enbart innefattar själva gränssnittet och kan delvis kopplas till hur bra användbarhet en produkt har. Däremot

behöver inte hög användbarhet medföra en bra användarupplevelse och vice versa, även om de är två sammanfallande koncept [13].

Det ska dock nämnas att användarupplevelse är ett något vagt begrepp med många olika definitioner, vilket också påvisats av Law et al [14]. I deras undersökning lyfts ISO's definition (ISO 9241-210) fram som den som bäst beskriver den generella bilden av användarupplevelse. Följande är ett utkast från definitionen:

> *"A person's perceptions and responses that result from the use or anticipated use of a product, system or service" [15]*

3.2.3 Användbarhetsproblem

Det finns ingen allmänt vedertagen definition i litteraturen av vad som är ett användbarhetsproblem och det är således inte helt lätt att definiera. Tullis och Albert [16] menar att det inte finns någon simpel definition och att det är lättare att ge exempel på problem. De tar bland annat upp följande exempel:

- allt som som förhindrar att det går att slutföra en uppgift
- allt som skapar någon typ av förvirring för användaren
- att användaren inte upptäcker något som bör upptäckas
- att utföra fel handling
- att användaren inte förstår navigeringen

Lavery et al [17] väljer att definiera ett användbarhetsproblem som:

> *"A usability problem is an aspect of the system and/or a demand on the user which makes it unpleasant, inefficient, onerous or impossible for the user to achieve their goals in typical usage situations"*

De menar också att det går att dela upp ett användbarhetsproblem i fyra delar: en *orsak* till problemet, en möjlig *uppdelning* av användarens interaktion, ett *utfall*, och till sist den *kontext* som allt sker i.

3.2.4 Principer för användbarhet och gränsnittsdesign

I användbarhetssammanhang refereras det nästan alltid till en eller flera av de personer som tagit fram principerna i detta avsnitt, och det är inte ovanligt att det är just dessa principer man refererar till. Donald Norman, Ben Shneiderman

och Jakob Nielsen har alla haft betydande roller när de gäller utvecklingen av användbarhet och gränssnittsdesign.

3.2.4.1 Normans designprinciper

Norman beskriver i sin bok ”The Design of Everyday Things” [18] sex stycken designprinciper som idag anses som grundpelare för att förstå varför vissa system är mer användbara än andra:

Affordance
Affordance ger användaren ledtrådar om hur ett element ska användas. Exempel på detta är en knapp som i sin utformning uppmuntrar till att trycka eller en spak som uppmuntrar till att dra. Detta kan användas i gränssnittsdesign genom att utforma element som knappar så att de exempelvis ser klickbara ut.

Visibility
Att ha visuellt tydliga funktioner underlättar för användaren att veta vad det finns för funktionalitet i systemet. Det ger även stöd till användare vid utförandet av uppgifter som kräver flera steg då en tydlig placering av element hjälper användare förstå vad som ska ske i nästa steg.

Mapping
Mappning beskriver relationen mellan ett element och dess funktion. Mappning kan vara naturlig, som att något rör sig uppåt om man trycker på en knapp med en pil uppåt. Mappning kan också vara inlärd, som att röd färg indikerar stopp.

Feedback
Feedback innebär att användaren förses med information om vad som hänt och vad utfallet av en handling blev. Detta hjälper användare att förstå att en handling är utförd. Feedback kan till exempel ske i form av ljud vid en knapptryckning.

Consistency
Människor lär sig nya saker genom att upptäcka mönster. Inlärningskurvan blir lägre om användaren av ett system kan applicera mönster som sedan tidigare redan är kända. Genom att använda sig av liknande mönster och liknande element för att utföra liknande uppgifter när ett gränssnitt designas kan man öka systemets användbarhet.

Constraints
Constraints uppnås genom att införa restriktioner i ett system.

Restriktionerna kan till exempel medföra att ingen felaktig data kan matas in eller att vissa händelser inte kan ske.

3.2.4.2 Shneidermans åtta gyllene regler för gränssnittsdesign

I sin bok ”Designing the user interface: Strategies for effective human-computer interaction” [19] nämner Ben Shneiderman åtta principer för gränssnittsdesign som han anser vara applicerbara på de flesta interaktiva system. Han beskriver också att principerna måste tolkas och anpassas för varje specifik designdomän samt att de kan ses som en bra startpunkt för personer som utformar webbsidor och mobila applikationer.

Strive for consistency
Gränssnittet bör hållas konsekvent och därmed bör sekvenser av handlingar i liknande situationer hållas konsekventa. Identisk terminologi bör användas i prompter, menyer och hjälpskärmar. Detsamma ska gälla för färger, typsnitt och layout.

Cater to universal usability
Ta hänsyn till olika användares behov vid design av gränssnitt. Att lägga till funktioner som ger förklaringar till nybörjare och möjligheten för experter att själva lägga till genvägar kan öka användarnas syn på systemets kvalitet.

Offer informative feedback
För varje användarhandling ska systemet erbjuda någon typ av återkoppling. För mindre och mer frekventa handlingar kan återkopplingen vara måttlig, medan för större och ovanliga handlingar bör återkopplingen vara mer framstående.

Design dialogs to yield closure
Sekvenser av handlingar bör delas upp i grupper så att de har en början, en mitt och ett slut. För att ge användaren en känsla av tillfredsställelse och en indikation på att en ny grupp av handlingar ska påbörjas bör slutförandet av varje grupp av sekvenser ha någon typ av informativ feedback.

Prevent errors
I så stor utsträckning som möjligt bör systemet designas så att användarna genom sina handlingar inte kan göra några allvarliga fel. Om användaren skulle göra något fel bör gränssnittet erbjuda en enkel och konstruktiv felhantering.

Permit easy reversal of actions
För att minska användarnas oro vid användning av produkten samt uppmana till utforskning av obekanta alternativ bör handlingar i så stor utsträckning som möjligt vara reversibla.

Support internal locus of control
Erfarna användare har ofta ett starkt behov av att känna att de kontrollerar gränssnittet och att det är gränssnittet som svarar till deras handlingar.

Reduce short-term memory load
Begränsningar av människans informationsbehandling i korttidsminnet kräver att displayer görs enkla. Gränssnitt där användarna behöver memorera information från en skärm till en annan bör undvikas.

3.2.4.3 Nielsens 10 heuristiska principer

Jakob Nielsen [20] har tagit fram följande 10 heuristiska principer för att användas som stöd vid gränssnittsdesign:

Visibility of system status
Systemet ska alltid hålla användare informerade om vad som händer, detta ska ske med hjälp av lämplig feedback inom en rimlig tid.

Match between system and the real world
Systemet ska förmedla information till användaren med hjälp av ord och koncept som användaren är bekant med snarare än systemorienterade termer. Dessutom ska information presenteras på ett naturligt och logiskt vis.

User control and freedom
Användare ska känna att de har kontroll över sina handlingar i systemet. Det ska finnas funktionalitet för att kunna ångra handlingar och återgå till ett tidigare tillstånd.

Consistency and standards
Systemet ska vara konsekvent utformat, användare ska inte behöva undra över om olika ord, situationer eller handlingar betyder samma sak.

Error prevention
Systemets design ska sträva efter att hindra användarna från att kunna begå misstag. Antingen genom att helt eliminera situationer där användare är benägna att göra misstag eller genom att låta användare bekräfta att de verkligen vill utföra en handling innan den sker.

Recognition rather than recall
Användarens minnesbelastning ska minimeras genom att låta objekt, handlingar och inställningar vara synliga i gränssnittet. Användare ska inte behöva komma ihåg information från ett tillstånd i systemet till ett annat. Användarinstruktioner ska vara synliga eller lätta att hitta när de behövs.

Flexibility and efficiency of use
Användare ska själva kunna anpassa delar av systemet. Hastigheten för att utföra uppgifter kan öka hos användare som är vana vid att använda systemet om de får möjlighet att själva anpassa hur handlingar som förekommer ofta ska utföras.

Aesthetic and minimalist design
Dialogrutor ska inte innehålla information som är irrelevant eller som användaren sällan använder. Varje bit av irrelevant information som presenteras konkurrerar om synlighet med den som är relevant.

Help users recognize, diagnose, and recover from errors
Systemet ska presentera felmeddelande som för användare är begripliga, där problemet beskrivs och ett förslag på en lösning ges.

Help and documentation
Ett system som kan användas utan hjälp av dokumentation är idealt men de finns situationer då dokumentation är ofrånkomlig. I dessa fall ska informationen i dokumentationen vara lätt att söka i och inte vara för lång, samt på enklast möjliga vis förklara hur en uppgift utförs.

3.2.5 Return of Investment (ROI)

Ett fungerande gränssnitt och en snygg design spelar mindre roll om användare inte förstår hur systemet ska användas. Ett gränssnitt som är intuitivt ger en ökad kundnöjdhet och produktivitet, vilket i sin tur leder till en lojal kundbas som litar på företagets produkter. Detta ger även synbara effekter kostnadsmässigt.

> *"In a Gartner Group study, usability methods raised user satisfaction ratings for a system by 40 %; when systems match user needs, satisfaction often improves dramatically." [21]*

En term som ofta brukar användas för att beskriva fördelarna med användbarhet är Return of Investment (ROI). I boken "Cost-Justifying usability" [21] nämns följande fördelar för produkter utvecklade både för internt- och externtbruk:

Intern ROI

- Ökad användarproduktivitet
- Minskat antal användarfel
- Minskade kostnader för träning
- Minskade kostnader till följd av att ändringar kan ske tidigt i designcykeln
- Minskad användarsupport

Extern ROI

- Ökad försäljning
- Minskade kostnader för kundsupport
- Minskade kostnader till följd av att ändringar kan ske tidigt i designcykeln

Om producenten redan från början vet vilken funktionalitet som ska produceras, minskar riskerna för att det utvecklas funktioner som är onödigt avancerade eller som inte kommer används. Vilket leder till kostnadsbesparingar i utvecklingsprocessen.

> *"Approximately 63 % of large software projects are over budget and the top four reasons rated as having the highest responsibility were related to usability engineering" [12].*

3.2.6 Metoder för användbarhetsutvärdering

För att kontrollera användbarheten hos ett gränssnitt kan en användbarhetsutvärdering göras. Nielsen [20] menar att det går att dela in metoderna för användbarhetsutvärdering i fyra olika kategorier:

- Automatiska – användbarhetsutvärderingen sker genom att låta gränssnittsspecifikationer köras i ett analyseringsprogram
- Empiriska– tester med riktiga användare utförs
- Formella – exakta modeller och formler används för att beräkna gränssnittets användbarhet
- Informella – baseras på utvärderarnas allmänna kompetens och erfarenhet

Empiriska metoder kallas ibland även för testmetoder medan informella metoder kan kallas för inspektionsmetoder. Hädanefter i rapporten används termerna test-

respektive inspektionsmetoder. Då automatiska och formella metoder förekommer mer sällan har valet gjorts att inte inkludera dessa i studien.

3.2.6.1 Inspektionsmetoder

Användarbarhetsproblem identifieras genom att låta utvärderare kontrollera gränssnittet mot etablerade standarder och är således beroende av utvärderarnas expertis inom området. Utvärderarna är ofta specialister inom användbarhet men de kan även vara utvecklare med kunskap inom gränssnittsdesign. Inga riktiga användare är involverade i inspektionsprocessen. Inspektion används ofta när ett redan befintligt gränssnitt är framtaget och dess användbarhet behöver utvärderas.

Nedan beskrivs två av de vanligare inspektionsmetoderna, heuristisk utvärdering och kognitiv genomgång. Andra inspektionsmetoder som inte tas upp i rapporten är ”formal usability inspections”, ”pluralistic walkthroughs”, ”feature inspection”, ”consistency inspection” samt ”standards inspection”.

Heuristisk utvärdering
Heuristisk utvärdering är den inspektionsmetod som är mest informell och utförs genom att låta experter bedöma om dialoger och andra interaktiva inslag i gränssnittet följer användbarhetsprinciper (även kallat heuristiker) [20]. Vanligtvis så inspekterar och bedömer varje enskild utvärderare gränssnittet själv utefter förutbestämda heuristiker. Dessa heuristiker kan till exempel vara Nielsens 10 heuristiker som beskrivs under avsnitt 3.2.4.3. Under en utvärderingssession inspekteras gränssnittet ofta flertalet gånger. Utvärderarna utför inspektionerna enskilt och när alla inspektioner är gjorda samlas resultaten in. Det är först efter resultaten samlats in som utvärderarna har tillåtelse att kommunicera med varandra [22].

Då det kan vara dyrt att hyra in användbarhetsexperter är det möjligt att låta mindre erfarna personer utföra en heuristisk utvärdering. Studier av Nielsen [23] har däremot visat att experter är mycket bättre på att hitta användbarhetsproblem och presterar således ett bättre resultat. I sina studier nämner Nielsen också något han kallar för dubbelexperter, vilket är personer som är experter både inom användbarhet och inom det specifika området som gränssnittet utvecklas för. Dessa presterar i sin tur bättre än ”vanliga” användbarhetsexperter.

Fördelarna med heuristisk utvärdering är att det är en enkel och billig metod för att utvärdera gränssnitt. Den kräver ingen större planering och kan börja

användas tidigt i en designprocess. Nackdelarna med metoden är att den ibland identifierar problem utan att egentligen ge förslag på hur problemen ska lösas samt är beroende av utvärderarnas nuvarande tankesätt och kunskapsnivå [23].

När en heuristisk utvärdering utförs förekommer problem som är så kallade ”false-positives”, vilket innebär att ett användbarhetsproblem som identifieras under en utvärdering inte visar sig vara ett problem när systemet väl sats i bruk. Problemen har följaktligen ingen påverkan på hur systemet uppfattas eller dess prestanda.

Enligt Nielsen [23] har forskning visat att heuristisk utvärdering hittar en större mängd mindre problem än vad andra metoder gör och att många av dessa inte upptäcks vid testning med användare. Därmed går det att diskutera i vilken utsträckning som dessa ska ses som användbarhetsproblem. Nielsen anser likväl att mindre användbarhetsproblem bör ses som riktiga problem även om de inte upptäcks under användartester. Exempelvis kan inkonsekvent placering av samma information på olika skärmar leda till att tiden det tar för användare att utföra uppgifter ökar med mindre än en sekund, vilket kan vara svårt att upptäcka vid användartester.

Archer [24] menar att personer som utför heuristiska utvärderingar kan vara partiska och eftersom de endast försöker efterlikna slutanvändarna kan de vara benägna att upptäcka problem som visar sig vara false-positives. I hans studie så var förhållandet mellan potentiella problem och de som senare identifierats som verkliga problem 10:1. Vilket innebar att en stor del av de upptäckta användbarhetsproblemen var false-positives.

Lauesen menar i sin bok [25] att ungefär hälften av alla problem som upptäcks under en heuristisk utvärdering är false-positives. Han har skämtsamt kallat det för första användbarhetslagen:

> *Heuristic evalution has only 50 % hit rate.*

Lagen är ganska kontroversiell men Lauesen menar att han och hans kollegor ofta stöter på en stor andel false-positives.

Kognitiv genomgång

Kognitiv genomgång är en uppgiftsbaserad metod där utvärderarna undersöker gränssnittets funktionalitet. Metoden fokuserar främst på kognitiva problem som inlärning (learnability or ease of learning) genom att analysera de mentala processer som krävs av användarna för att utföra vissa uppgifter. För att genomföra en kognitiv genomgång så väljer utvärderaren en specifik uppgift från en större mängd uppgifter som gränssnittet ska stödja. Sedan bestäms en eller

flera sekvenser av handlingar som en teoretisk användare måste utföra för att lyckas med den specifika uppgiften. Utvärderaren undersöker därefter varje steg i dessa sekvenser och bedömer om en hypotetisk användare skulle kunna välja rätt handling för att fortsätta sekvensen. Om utvärderaren skulle identifiera ett problem så ska en anledning till detta anges [26].

Precis som i en heuristisk utvärdering så reflekterar resultatet utvärderarens kompetens inom området, men i en kognitiv genomgång undersöks specifika användaruppgifter snarare än gränssnittet som en enhet. Kognitiv genomgång kan med fördel användas tidigt i designprocessen då den inte kräver någon implementation av gränssnittet [26].

3.2.6.2 Testmetoder

Testmetoder, eller användbarhetstestning som det också kallas, innebär att tester utförs med användare som representerar produktens målgrupp. Vanligtvis utför testpersonerna ett antal förutbestämda uppgifter (testfall) samtidigt som testövervakaren observerar och registrerar resultatet.

Det finns ett stort antal olika typer av metoder för användbarhetstestning, allt från experiment med stora provstorlekar och komplex testdesign till informella kvalitativa studier med endast en testperson. Varje testmetod har sina egna resurskrav och sitt eget syfte, och passar således vid olika tillfällen. Det är inte ovanligt att det i litteraturen används olika termer för att beskriva identiska metoder [27].

Användbarhetstestning kan delas in i fyra olika kategorier: formativa tester, summativa tester, valideringstester (ibland kallade verifieringstester) och jämförelsetester.

Formativa tester används ofta i början av designprocessen för att få återkoppling från användare på koncept och designskisser. De är ofta informella och målet är samla in information som kan användas vid design snarare än att utvärdera gränssnitts användbarhet. Det är inte ovanligt att formativa tester endast utvärderar en liten del av ett gränssnitt och endast involverar ett litet antal testpersoner [28].

Summativa tester är mer formella än formativa tester och ämnar utvärdera ett gränssnitts användbarhet. De är den vanligaste typen av användbarhetstester och utförs vanligtvis tidigt eller halvvägs in i en designprocess, vanligtvis när den fundamentala designen har fastställts [27]. En god experimentell design är viktig för summativa tester och metrics för efficiency och user satisfaction används ofta.

Uppgifter som används i testen motsvarar ofta kärnfunktioner i systemet men även ny funktionalitet kan testas [28].

Valideringstester utförs ofta sent i en designprocess och är avsedda för att mäta ett systems användbarhet mot fastställda användbarhetsstandarder eller för att verifiera att problem som upptäckts i en tidigare utvecklingsprocess har åtgärdats och att inga nya har introducerats [27].

Den fjärde och sista typen av användartester kallas för jämförelsetester och används för att jämföra två eller flera olika designer. De kan användas för att jämföra hela gränssnitt eller små element som knappar och ikoner. Jämförelsetester kan utföras när som helst under utvecklingsprocessen och i samband med att någon av ovanstående tre tester utförs [27].

Nedan beskrivs två av de vanligaste metoderna för att samla in testdata, thinking aloud och observation. Även tre andra datainsamlingsmetoder beskrivs kortfattat, dessa metoder används ofta för att bedöma användbarhet men behöver nödvändigtvis inte användas i samband med testning.

Thinking aloud

Jakob Nielsen skrev i sin bok ”Usability Engineering” [12] från 1993 att ”Thinking aloud may be the single most valuable usability engineering method.” och än idag står han fast vid påståendet. Metoden innebär att testpersonen tänker högt (uttrycker sina tankar i ord) samtidigt som denna utför specifika uppgifter under testets gång. Detta gör det lättare att förstå hur användare ser på systemet och kan ge insikter om varför ett problem finns. Det är exempelvis inte ovanligt att få reda på varför användare gör fel och varför vissa saker är lättare att genomföra än andra. En stor fördel med thinking aloud-metoden är att det går att få in en stor mängd kvalitativ data från en liten mängd testanvändare [29]. Nielsen nämner också att [29] thinking aloud är väldigt robust, det går att få bra fynd även om metoden inte utförs på ett helt och hållet korrekt sätt.

En av nackdelarna med thinking aloud-metoden är att det hela kan uppfattas som en ganska onaturlig situation för testdeltagarna, de flesta personer sitter inte och tänker högt för sig själva [29]. Det finns däremot en variant av thinking aloud kallad för konstruktiv interaktion där två testpersoner utför tester tillsammans. Testsituationen blir då mer naturlig eftersom det är vanligt att berätta för varandra vad man tänker när man löser ett problem tillsammans [22].

Observation

Ett sätt att samla in data är att direkt observera användarna när de utför uppgifter. Detta kan ske både i ett laboratorium och i användarnas naturliga miljö. När användare observeras är det viktigt att inte störa deras arbete, därför bör anteckningar tas på ett så diskret sätt som möjligt. Helst bör observatören praktiskt taget vara osynlig för att garantera normala arbetsförhållanden. I vissa fall kan det vara lämpligt att filma testpersonen för att göra observationsprocessen mindre påträngande. Nackdelen med att filma är att det tar ungefär tio gånger längre tid att analysera en videoinspelning än vad det tar att utföra testet. I många fall kan det därför vara bättre att utföra tester utan inspelning för att istället få tid till tester med flera personer [22].

Frågeformulär och intervjuer

Många aspekter inom användbarhet går bäst att studera genom att fråga användarna om vad de tycker. I detta fall är frågeformulär och intervjuer bra metoder för att ta reda på hur användare använder ett system och vilka delar de gillar och inte gillar. De är båda indirekta metoder eftersom de inte studerar gränssnittet direkt utan endast användarnas åsikter om gränssnittet. I sin bok ”Usability Engineering” är Nielsen [12] tydlig med att det är viktigt att inte alltid ta användarnas åsikter som sanning. Data över användarnas faktiska beteende bör gå före deras påståenden om vad de tror att de gör.

Loggning av faktisk användning

Att logga data innebär att ett loggverktyg automatiskt samlar in data över hur ett system används, se kapitel 2.2 för en närmare beskrivning om hur detta går till i Google Analytics. Vanligtvis används loggverktyg för att samla in information om hur systemet används av slutanvändare efter det satts i bruk, men det kan också användas som komplement under testning för att samla in mer detaljerad information som knapptryckningar, felmeddelanden och vilka funktioner som använts ofta. Att logga användares verkliga användning av ett system kan vara mycket användbart eftersom det visar hur användarna faktiskt använder systemet samt gör det möjligt att på ett enkelt sätt samla in data från ett stort antal användare som arbetar under olika omständigheter [12].

Testmetoders begränsningar

Att utföra tester med användare är ingen garanti för att man i slutändan får en produkt med hög användbarhet eller att produkten blir en succé. Rubin och Chisnell [27] tar upp fyra begränsningar med användartester:

- Testning sker alltid i en artificiell situation. Även om tester sker i laboratorium eller ute på fältet så är det fortfarande bara en skildring av verkligheten och inte en verklig situation i sig själv.
- Testresultat bevisar inte att en produkt fungerar. Även om ett test visar statistiskt signifikanta resultat så bevisar det inte att produkten fungerar. Statistisk signifikans är bara ett mått på sannolikheten att ett resultat inte berodde på slumpen.
- Testdeltagare är sällan representativa för hela målpopulationen.
- Testning är inte alltid den mest lämpade tekniken att använda. Det kan till exempel finnas fall där det kan vara mer lämpligt att utföra en heuristisk utvärdering om man tar hänsyn till kostnad, tidsåtgång och noggrannhet.

3.2.6.3 Inspektion kontra testning

Flertalet studier har gjorts för att undersöka om det är inspektions- eller testmetoder som är effektivast på att hitta användbarhetsproblem. De har båda sina för- och nackdelar. Inspektionsmetoder hittar användbarhetsproblem som förbises av testmetoder samtidigt som testmetoder hittar användbarhetsproblem som förbises av inspektionsmetoder. För att uppnå bästa resultat bör de två metoderna kombineras [30].

I en studie gjord av Karat et al [31] konstaterades det att testmetoder hittar både fler och mer allvarliga användbarhetsproblem än inspektionsmetoder. De ansåg att inspektionsmetoder var ett bra alternativ när det fanns begränsat med resurser. Dessutom ansåg de att test- och inspektionsmetoder kompletterar varandra bra i det avseende att de ger olika resultat då de i en del fall upptäcker olika typer av användbarhetsproblem. Jeffries et al [32] å sin sida fann i sin studie där fyra olika utvärderingsmetoder användes att Heuristisk utvärdering gav bäst resultat.

Tan et al [33] fann att heuristisk utvärdering och testmetoder kompletterar varandra bra och att ingen av metoderna kan ersätta den andra. De anser att båda metoderna behövs i en användbarhetsstudie. Även Jeffries och Desurvire [34] fann att man bör kombinera flera utvärderingsmetoder för att uppnå bästa resultat.

3.2.7 Utmaningar med användbarhetsutvärdering av mobila pekskärmsbaserade enheter

Mobila enheters gränssnitt har tre huvudaspekter som karakteriserar deras gränssnittsdesign: De används i första hand i användarens händer, de används trådlöst och de har stöd för att installera nya applikationer. En aspekt som bör beaktas vid utvärdering av användbarhet på mobila enheter är den lilla skärmstorleken, trots att enheten behöver visa stora mängder information.

Inostroza et al [35] har definierat följande funktioner och egenskaper som gör användbarhetsutvärdering av mobila pekskärmsbaserade enheter till en utmanande process:

- **Mobile context of use:***"All aspects related to the interaction between the user, the system and the environment occur concurrently. Auditive distraction (e.g.: noise) or visual distraction (e.g.: excess or lack of lighting), can disturb the user. A proper evaluation method should consider most of the context of use characteristics."*[35]

- **Small screen size:***"To make information fit in a small display can be not aesthetically pleasing, or even worst, completely illegible. Different tests should be performed, considering distinct screen sizes."*[35]

- **Screen resolution:***"Low screen resolution can degrade the perceived quality of the multimedia data displayed in the device screen. Different screen resolutions may lead to significantly different usability issues."*[35]

- **Limited processing, memory and energy capabilities:***"Some applications require large amounts of memory and/or processing power for graphical support and data processing, which can exceed the device capabilities. The system performance may vary according to available memory, energy and processing power. These issues should be considered in usability evaluations."*[35]

- **Data entry methods:***"Small buttons and labels can reduce the efficiency and effectivity in data entry, reducing the data entry speed and raising the error rate. Evaluations that minimize the impact of data entry methods should be performed (as long as they are not the research focus)."*[35]

3.2.8 Kvalitativ och kvantitativ data

Data kan delas upp i två typer, kvalitativ- och kvantitativ data. Kvantitativ data kan kvantifieras i kategorier eller siffror där analys sker genom statistiska och matematiska metoder. I motsats till kvantitativ data är kvalitativ data svårare att analysera matematiskt och utgörs av ord och beskrivningar. Kvalitativ data kan till exempel bestå av öppna svar i en intervju där den intervjuade personen beskriver känslor eller hur en produkt upplevs. Rorher [36] belyser dock vikten av att inte endast se kvalitativ data som resultatet av öppna frågor i en undersökning. Utan istället dela upp kvantitativ och kvalitativ data utefter hur data samlas in.

Kvalitativa studier generar data rörande attityder gentemot en produkt och användares beteende genom att observera dem under utförandet. Detta gör det möjligt för de som utför undersökningen att ställa frågor eller anpassa studieprotokollet så att det på ett bättre sätt uppfyller dess mål [36].

I motsats till kvalitativa studier där användarbeteende studeras direkt, studeras användare i kvantitativa studier indirekt. Detta görs genom att i efterhand analyserar data rörande attityder gentemot en produkt och användares beteende, oftast genom att studera resultatet av en undersökning eller med hjälp av ett analysverktyg [36].

3.2.9 Usability Metrics

Metrics är ett sätt att mäta eller utvärdera ett bestämt fenomen eller en bestämd sak. Det går till exempel att säga att någonting är längre eller snabbare eftersom det finns möjlighet att mäta eller kvantifiera något av dess attribut, i detta fall längd och snabbhet. För att detta ska fungera krävs det att det finns en överenskommelse om hur saker ska mätas samt en konsekvent och pålitlig metod för utförandet, till exempel är en centimeter alltid en centimeter oavsett vem som utför mätningen. Användbarhetsmetrics är baserade på ett tillförlitligt mätsystem där användning av samma uppsättning av mätningar varje gång någonting mäts alltid ska resultera i jämförbara resultat. Vad är det då som skiljer användbarhetsmetrics från andra typer metrics? De avslöjar någonting om användarupplevelsen och interaktionen mellan användaren och produkten. Framförallt finns det tre krav som alla användbarhetsmetrics måste uppfylla:

- De måste på något sätt vara observerbara, antingen indirekt eller direkt. Till exempel genom att notera att en uppgift genomfördes med framgång eller notera tiden som krävdes för att genomföra en uppgift.

- De måste vara kvantifierbara - de måste kunna omvandlas till ett nummer eller på något sätt kunna räknas.
- De kräver också att saken som mäts representerar någon aspekt av användarnas användarupplevelse, till exempel kan en användbarhetsmetric visa att 90 % av användarna kan slutföra en mängd uppgifter inom en minut.

Användbarhetsmetrics är inget självändamål utan kan snarare användas som hjälp för att ta beslut. De kan hjälpa till att svara på kritiska frågor som:

- Kommer användarna att använda produkten?
- Är den nya produkten mer effektiv att använda än den nuvarande produkten?
- Vilka är de mest signifikanta användbarhetsproblemen med produkten?
- Har det skett förbättringar från föregående designiteration till nästa? [16]

3.2.9.1 Performance metrics

Resultatet av performance metrics räknas ut med hjälp av mätningar som baseras på användarbeteenden. Utöver användares beteenden krävs också en eller flera specifika uppgifter eller mål. Utan specifika uppgifter skulle det inte gå att mäta performance metrics, detta eftersom det inte skulle gå att avgöra om användaren lyckats eller misslyckats med att utföra en uppgift.

För att mäta tiden och ansträngningen som krävs för att utföra en uppgift är performance metrics det bästa alternativet. Om det till exempel skulle visa sig att det tar fyra gånger längre tid än vad man kan förvänta sig för användare att utföra en uppgift så är det en indikation på att det finns utrymme för förbättring av produkten. Performance metrics är inte bara användbart för att påvisa att det existerar ett problem utan kan även användas för att undersöka omfattningen av problemet. Det går till exempel att beräkna hur många användare som kommer ”utsättas” för samma användbarhetsproblem [16].

I huvudsak är performance metrics bra på att svara *vad* snarare än *varför*. Det går att se att det finns problem inom vissa delar av ett system och hur allvarliga dessa är, men för att svara på varför dessa problem uppstår så krävs ofta någon form av komplement. I det här avsnittet kommer följande fem performance metrics att beskrivas:

1. Task success - mäter förmågan hos användarna att slutföra en given uppgift.
2. Time-on-task - mäter hur mycket tid som krävs för att slutföra en given uppgift.
3. Errors - beskriver antalet misstag som görs då användarna utför en uppgift.
4. Efficiency - mäter ansträngningen som krävs för att utföra en viss uppgift.
5. Learnability - mäter hur användarnas prestation förändras med tiden.

Task success
Den vanligaste typen av metric är task success, denna kan i princip användas i alla användbarhetsutvärderingar där det finns definierade uppgifter för testpersonerna att utföra. Task success är också väldigt lätt att relatera till då det inte krävs några detaljerade beskrivningar av mättekniker eller statistik för att förstå innebörden.

När data samlas in för att mäta task success är det viktigt att varje uppgift har ett tydligt definierat slut. Att köpa en produkt, hitta svaret på en specifik fråga eller fylla i ett formulär är några exempel på uppgifter med tydligt definierade slut. Ett exempel på en uppgift som inte har ett tydligt slut är att undersöka olika möjligheter för att pensionsspara.

Binary success är det enklaste och vanligaste sättet för att mäta task success. Detta görs genom att låta utgången av varje försök bli antingen ett lyckat eller ett misslyckat försök. Ett annat sätt för att mäta task success är att använda sig av levels of success, vilket innebär att det sätts värde i att försökspersonen delvis klarar en uppgift. De olika metoderna lämpar sig olika bra beroende på vilken typ av uppgifter som används i försöket samt möjligheten till att sätta upp olika nivåer för att avgöra hur väl en uppgift har utförts [16].

Time-on-task
Time-on-task är ett bra mått på en produkts systemeffektivitet och beskrivs av tiden det tar från det att användaren påbörjar en uppgift till att den avslutas. Tiden det tar för en användare att utföra en uppgift säger mycket om användbarheten hos en produkt, generellt sett så gäller det att en kortare tid per uppgift ger en bättre användarupplevelse. Det tillhör ovanligheten att en användare klagar på att det går för fort att slutföra en uppgift men det finns en del undantag. Ett av dessa är då användaren ska lära sig något under en uppgift, man vill då undvika att användaren tar sig igenom uppgiften för snabbt och

missar delar av det som ska läras in. Ett annat undantag är spel, där det oftast är upplevelsen i sig som är det viktiga och inte hur snabbt en användare slutför det.

I produkter där samma uppgifter utförs upprepade gånger är det extra viktigt att optimera time-on-task för att minska användarnas ansträngning samt öka effektiviteten. En fördel med time-on-task är att det är relativt enkelt att räkna ut kostnadsbesparingarna som kan göras genom att systemeffektiviteten ökar, för att därefter kunna beräkna ROI [16].

Errors
Det är inte ovanligt att errors och användbarhetsproblem ses som samma sak bland användbarhetsspecialister. Detta är dock inte fallet trots att de är nära relaterade. Ett användbarhetsproblem är den underliggande orsaken till ett problem medan ett error kan ses som utfallet av att ett användbarhetsproblem uppstått. I huvudsak är errors felaktiga handlingar som kan leda till att det inte går att slutföra en viss uppgift.

Det är bra att mäta antalet errors för att utvärdera en applikations användarupplevelse. En applikation där användarna inte gör några errors alls kan ses som mycket användbar medan en applikation där det görs många errors kan visa raka motsatsen. Fel kan visa hur många misstag som görs för att utföra uppgifter, var de görs i applikationen samt vilken typ av errors som görs i olika designer. Däremot är det inte lämpligt att mäta errors i alla situationer. Främst är det intressant att kolla på situationer där:

- errors resulterar i att applikationen blir signifikant mindre effektiv att använda.
- ett error resulterar i stora kostnader.
- ett error resulterar i att en handling inte går att utföra [16].

Efficency
Det är inte ovanligt att time-on-task används som ett mått på systemeffektivitet men det finns även andra sätt att mäta ett systems effektivitet. Ett av dessa är att titta på ansträngningen som krävs för att utföra en uppgift. Detta görs vanligtvis genom att mäta antalet handlingar eller steg som krävdes av användaren när en uppgift utfördes. En handling kan specificeras på olika vis beroende på vilket system som utvärderas. Det kan till exempel vara ett klick på en länk eller ett knapptryck på en mikrovågsugn. Varje handling kräver viss ansträngning av användaren vilket innebär att fler handlingar kräver en större ansträngning.

Det finns olika typer av ansträngning och i sammanhanget pratas det vanligtvis om kognitiv och fysisk ansträngning. Den kognitiva ansträngningen innefattar saker som att hitta rätt ställe för att utföra en handling, avgöra vilken typ av handling som är nödvändig samt tolka resultatet av en handling. Den fysiska ansträngningen innefattar istället handlingar som att flytta datormusen eller mata in text via ett tangentbord [16].

Ett annat sätt att mäta efficiency är med hjälp av lostness. Måttet introducerades först av Smith [37] och har sedan blivit ett vedertaget begrepp för att beräkna efficiency [16]. Måttet beskriver hurpass lätt det är för användare att hitta rätt i gränssnittet då denna löser en uppgift. Detta görs genom att använda antalet unika skärmar deltagaren besökt (N), det totala antalet skärmar som besökts (inklusive återbesök på skärmar) (S) och det lägsta antalet skärmar som krävs för att lösa uppgiften (R). Lostness räknas sedan ut med hjälp av nedanstående formel [37]:

$$L = \sqrt{[(N/S - 1)^2 + (R/N - 1)^2]}$$

De ideala resultatet för ovanstående formel är L=0, detta medför i teorin att deltagaren inte alls är vilsen under utförandet av en given uppgift. Smith kunde utläsa från sina observationer att ett värde högre än L=0,5 medförde att försökspersonerna uppträdde vilsna, medan försökspersoner med ett värde på mindre än L=0,4 verkade veta vart de befann sig. För värden mellan 0,5 och 0,4 rekommenderar Smith [37] att individuella bedömningar utifrån observationer bör göras.

Learnability

De flesta produkter har någon form av inlärningskurva. Vanligtvis tar det en viss tid innan användaren har tillräckligt med erfarenhet för att fullt behärska en produkt. Erfarenheten baseras på mängden tid som spenderats med att använda produkten samt antalet handlingar som utförts. Learnability kan mätas genom att titta på hur lång tid och hur mycket ansträngning det krävs för att behärska ett system eller en produkt.

Inlärningen kan ske under kortare perioder med lite eller ingen tid mellan det att användaren interagerar med produkten, alternativt under längre perioder där det kan gå veckor mellan det att användare interagerar med produkten. I det första fallet tenderar användarna att bygga upp en mental modell av hur produkten fungerar. I det här fallet spelar inte minnet så stor roll utan användare fokuserar i stället på att hitta strategier för att maximera sin effektivitet. I det senare fallet

då det går långa perioder mellan användandet av produkten blir det viktigare med minnet [16].

3.2.9.2 Preference metrics

Att fråga användarna vad de tycker om användarupplevelsen och användbarheten på ett system är en metod som ofta används för att samla in data till en utvärdering. Ett samlingsnamn för data som samlas in genom att fråga användarna om information är preference metrics.

Hur frågor ska formuleras för att få bra data är inte helt självklart. Frågorna kan presenteras på många olika sätt, där ett alternativ är att använda sig av bedömningsskalor. Ett annat alternativ är att presentera en lista med egenskaper där användarna väljer de egenskaper som enligt dem själva bäst beskriver systemet. Relevanta egenskaper att ställa frågor kring skulle kunna vara systemets utseende, terminologi eller navigering. En annan möjlighet är att använda sig av öppna frågor som "Vad tyckte du var bäst med systemet?".

Vilken metod som är bäst att använda beror helt på vilken typ av system som utvärderas och vilka resurser som finns tillgängliga. En muntlig redogörelse är till exempel lättast för deltagarna i undersökningen, men insamling och behandling av data blir mer krävande för observatören. Krävs det många deltagare i undersökningen är det därför bättre att använda sig av formulär med bedömningsskalor.

Preference metrics gör det möjligt att få en bild av hur användarna uppfattar systemet, hur de interagerar med det och vilka känslor det framkallar. I många situationer kan användarnas reaktioner gentemot systemet vara det som är viktigast för skaparna. Det behöver inte göra något att det tar lång tid för användare att utföra en uppgift om det är en positiv upplevelse som gör de nöjda med produkten [16].

Tillfällen då data bör samlas in
Data kan samlas in under olika tillfällen under utvärderingen. De två bästa tillfällena för att samla in data är precis efter en uppgift utförts samt i slutet på undersökningen då alla uppgifter utförts, båda alternativen har sina fördelar. Att samla in data efter varje uppgift kan hjälpa till med att peka ut uppgifter och delar av systemet där det finns problem, medan en mer djupgående utfrågning i slutet av undersökningen kan ge en bra bild av användarens helhetsbild av systemet.

Studier har visat att personer som tillfrågats personligen eller via telefon tenderar att ge mer positiv feedback än om de hade blivit tillfrågade via ett anonymt webbformulär. Detta kallas “the social desirability bias”, de tillfrågade personerna har en benägenhet att ge svar som får dem själva att framstå som bättre eller svar som inte gör utvärderaren besviken. Felaktig resultat kan därför undvikas genom att låta deltagarna vara anonyma. Kräver undersökningen att deltagarna är identifierbara kan till exempel observatören lämna rummet om ett formulär ska fyllas i [16].

3.2.10 Automatiserade användbarhetsutvärderingar

Följande arbeten har studerats mer ingående för att ge stöd till detta arbete. De har bidragit genom att dels presentera olika metoder för användbarhetsutvärdering och dels genom att presentera verktyg, både för insamling av data och för att automatisera processen med att utvärdera denna.

Ivory och Hearst [38] har i en uppmärksammad rapport granskat olika metoder för att utvärdera användbarhet. Fokus ligger på att analysera metoder för användbarhetsutvärdering och hur delar av dessa kan automatiseras. De belyser även problemet med att resultatet från en utvärdering av samma gränssnitt kan skilja sig beroende på vem som utfört utvärderingen. I rapporten dras slutsatsen att vid användning av automatiserade tester kan viktig kvalitativ och subjektiv information förbises. Samtidigt framkommer det att automatiserade tester kan vara ett bra alternativ vid jämförelse av olika designer samt i kombination med metoder som användbarhetstestning och heuristisk utvärdering. Andra fördelar som nämns är minskade kostnader av icke automatiserade tester samt ökad konsekvens i utvärderingsresultaten.

Få arbeten har gjorts med fokus på mobila applikationer men Burzacca och Paternò [39] har undersökt möjligheten att utvärdera användbarheten av en mobil applikation i en miljö där utvärderarna och användarna är separerade i form av tid och/eller rum, även kallat remote evaluation. Framförallt ligger fokus på olika aspekter som bör adresseras vid remote evaluation, samt verktyg för att underlätta arbetet med att analysera loggdata från mobila applikationer. I artikeln beskrivs följande tre huvudaspekter som bör adresseras vid remote evaluation:

- vad som kan loggas,
- hur informationen som samlats in kan behandlas,
- hur användardata kan presenteras för att kunna analyseras av utvärderare

och designers.

I en studie som gjordes av Hilbert och Redmiles [40] undersöktes hur man med hjälpa av händelser i gränssnittet kunde utläsa användbarhetsinformation. Händelserna är definierade som rörelser och klick från muspekaren samt tangentnedslag. Studien resulterade i slutsatsen att det kan vara svårt att utvärdera källan till en gränsnittshändelse utan att känna till relaterade händelser. Däremot kunde informationen vara användbar för att svara på frågor som "hur ofta gör användare X" eller "hur ofta sker Y" vilket kan vara intressant då utvecklare och designers kan se vilken påverkan förändringar i gränssnittet har.

4 | Heuristisk Utvärdering

4.1 Metod

Innan arbetet med att analysera användardata från Google Analytics påbörjades gjordes en heuristisk utvärdering av de mobila applikationerna. Den heuristiska utvärderingen tjänade två syften, dels gav den en överblick över antalet användbarhetsbrister som fanns i applikationerna samt inom vilka områden dessa befann sig, och dels användes den för att undersöka om en heuristisk utvärdering är ett bra komplement till den analys som gjordes av användardata från Google Analytics.

Den heuristiska utvärderingen genomfördes genom att låta tre personer med kännedom om gränssnittsdesign och användbarhetsutvärdering gå igenom en checklista med frågor och kriterier, där två av de tre personerna är författarna av denna rapport. Valet att låta tre personer genomföra utvärderingen baseras på en rekommendation av Jakob Nielsen [41], där han hävdar att fler än 3-5 personer nödvändigtvis inte bidrar med ytterligare information.

Checklistan, se appendix A, togs fram genom att dels använda heuristiker som tagits fram av andra, och dels genom att ta fram egna heuristiker för att täcka in alla områden på de applikationer som utvärderats. Merparten av de studier där heuristiker för mobila gränssnitt tagits fram bygger på de arbete som gjorts av Nielsen och Molich [42] [43]. Dessa har sedan i olika utsträckningar anpassats för respektive system. Stora delar av checklistan som använts i fallstudien har tagits fram av Gómez et al. [44]. Den har modifierats genom att ta bort delar som ansetts vara irrelevanta samt att frågor har ändrats och lagts till. De frågor som lagts till har till största del inspirerats av Nielsen och Molich och därefter anpassats till de specifika applikationerna som berörs i denna rapport.

4.2 Resultat

De tre utvärderarna gick igenom en checklista bestående av frågor med ja/nej svar samt utrymme för att lägga till kommentarer. Den fullständiga checklistan finns presenterad i appendix A. Nedan presenteras de frågor där någon av utvärderarna upptäckt ett problem samt en kommentar till varför det anses vara ett användbarhetsproblem.

Tabell 4.1: Frågorna i tabellen kommer från checklistan i Appendix A, kommentarerna är de som skrivits av utvärderarna under den heuristiska utvärderingen. Tabellen visar också antalet utvärderare som upptäckt problemen. Relaterade frågor har grupperats.

Ladda Refill

Fråga	**Kommentar**
- In multipage data entry screens, is each page labeled to show its relation to others? - Is there any way to inform the user about where they are and how to undo their navigation?	När en laddning görs finns ingen tidslinje som indikerar hur många steg som är kvar för att slutföra uppgiften. Samtliga utvärderare påpekade detta.
- Are all the items in a list on the same page? Are they sorted in an order that matches the needs of the task?	Sorteringen av inlagda telefonnummer är lite oklar. Telefonnummer verkar sorteras efter ordningen som de lagts till. Hade kunnat sorteras i alfabetisk ordning efter namnet som telefonnumret är knutet till. 1 av 3 utvärderare påpekade detta.
- Are low discoverable areas as touch buttons well identifiable? - Are touchable areas sufficiently big? (Research has shown that the best target size for widgets is 1cm x 1cm for touch devices)	Knappen för att redigera tillagda telefonnummer är liten och kan således vara svår att både upptäcka och interagera med. 2 av 3 utvärderare påpekade detta.

- Are users prompted to confirm commands that have drastic, destructive consequences?	Om man tar bort ett inlagt nummer behöver man inte bekräfta detta, vilket kan bli ett problem om man råkar trycka på ta bort-knappen istället för uppdatera-knappen (ta bort-knappen ligger precis under uppdatera-knappen). Samtliga utvärderare påpekade detta.
- Is there sufficient space between buttons that perform different actions? (For example a change button and an erase button)	Knapparna för att uppdatera och ta bort telefonnummer ligger väldigt nära varandra, borde kanske vara ett större mellanrum mellan dessa alternativt att ta bort knappen placeras längst ner på skärmen. Samtliga utvärderare påpekade detta.
- Do data entry screens and dialog boxes indicate when fields are optional? - Are inactive menu items grayed out or omitted?	På första skärmen när man ska ladda kan man antingen ange ett sparat nummer eller ett nummer manuellt, dock så finns det inget förutom texten som tyder på att en av dessa är valfria. Det står heller ingenting på nästa skärm om vilket nummer man försöker ladda. 1 av 3 utvärderare påpekade detta.

Eftersom det visade sig att viss funktionalitet i Google Analytics var felimplementerad i två av de tre applikationerna (se avsnitt 5.1.1) presenteras endast resultatet från Ladda Refill-applikationen. Detta då det i de andra två fallen inte gick att göra en ordentlig jämförelse av resultaten från de heuristiska utvärderingarna med resultaten från analysen av användardata i Google Analytics.

5 | Google Analytics och insamlad användardata

5.1 Metod

5.1.1 Implementationskontroll

Eftersom Google Analytics redan hade implementerats i de mobila applikationerna när denna studie påbörjades togs beslutet att implementationen av Google Analytics samt insamlad data behövde kontrolleras. Först verifierades det att skärmvisningar, som är den mest grundläggande funktionaliteten i Google Analytics, var implementerad korrekt. Detta gjordes med hjälp av realtidsfunktionen och en applikation där en testanvändare var inloggad. Det kontrollerades att de skärmar som var aktiva i applikationen också var aktiva i Google Analytics. Sessioner, händelser och annan funktionalitet som inte kunde verifieras via realtidsfunktioenen verifierades genom att göra en objektiv bedömning av data som fanns tillgänglig i Google Analytics. Eftersom det fanns tre applikationer med liknande funktionalitet kunde jämförelser göras mellan dessa. Detta gjorde det lättare att bedöma reliabiliteten hos data.

5.1.2 Bearbetning av insamlad data

Resultatet från litteraturstudien visade att det var fem performance metrics som var aktuella att undersöka med Google Analytics. För att få data till beräkningarna av de metrics som var aktuella togs ett antal specifika uppgifter fram. För att elicitera uppgifterna utformades ett antal kriterier som uppgifterna skulle uppfylla:

- Uppgifterna skulle gå att applicera på de tre olika applikationerna för att

göra det möjligt att jämföra resultaten.

- Uppgifterna skulle ha både en tydligt definierad början och ett tydligt definierat slut.
- Uppgifterna skulle gå att filtrera med hjälp av Google Analytics segmenteringsfunktion.

Detta resulterade i följande tre uppgifter som sedan användes i mätningarna:

- Genomföra ett köp
- Registrera en användare
- Lägga till ett nytt telefonnummer

Först undersöktes möjligheterna till att få fram data som kunde påvisa om det var möjligt att mäta performance metrics utifrån ovan nämnda uppgifter. Detta gjordes genom att använda Google Analytics filtreringsfunktion där segment för att få fram relevant data skapades. Därefter kunde beräkningar utföras för att ta fram resultat till olika performance metrics.

5.1.2.1 Task Success

Binary success användes under mätningarna av task success. Där ett lyckat utförande av uppgifterna definierats som att ett köp genomförts, en användare registrerat sig eller att ett telefonnummer lagts till. Om någon av de tidigare nämnda kriterierna inte uppfylldes räknades försöket som ett misslyckat försök. Eftersom det inte gick att på ett enkelt och tydligt sätt dela upp de olika uppgifterna i levels of success användes endast binary success.

5.1.2.2 Time-on-task

För time-on-task användes två stycken tillvägagångssätt. I det första fallet mättes time-on-task genom att ta fram den genomsnittliga sessionstiden för användarna som utfört de uppgifter som beskrivs ovan. Att den genomsnittliga sessionstiden användes vid beräkningar av time-on-task medförde att endast användare som utförde den specifika uppgiften togs med i beräkningarna. Det vill säga, personer som öppnade applikationen utförde uppgiften och sedan stängde ner applikationen.

Exempel:
För att genomföra en uppgift krävs det att användaren besöker skärmarna A,B,C och D i inbördes ordning. De användare som tagits med i beräkningarna är endast de användare som startat på skärm A för att sedan slutföra uppgiften genom att besöka skärm B,C,D och sedan stänga ner applikationen.

Ytterligare ett tillvägagångssätt användes för att mäta time-on-task. Där användes de händelser som fanns implementerade i Google Analytics. Mätningarna gjordes för den genomsnittliga sesionstiden då endast en händelse utlöstes som var kopplad till den specifika uppgiften. Användarna kan således ha besökt andra skärmar än de som behövs för att utföra uppgiften men de har inte utlöst några andra händelser.

Exempel:
För att utlösa en händelse krävs det att användaren klickar på en specifik knapp på skärm D. De användare som tagits med i beräkningarna är de som klickat på den specifika knappen på skärm D men inte utlöst några andra händelser i applikationerna. Detta innebär att användarna kan ha gjort i princip vad som helst så länge ingen annan händelse har utlösts.

5.1.2.3 Errors

För att undersöka om det gick att upptäcka errors med hjälp av Google Analytics användes beteendeflödesrapporten. Beteendeflödesrapporten visualiserar vägen användarna går från en skärm till nästa. Denna studerades i ett försök att identifiera tydliga avvikelser i användarnas navigationsmönster. Dessa avvikelser skulle kunna vara:

- Användarna besöker skärmar i en annan ordning än den som är tänkt för att utföra en specifik uppgift.
- Användarna går fram och tillbaka mellan olika skärmar.
- Användarna besöker en viss skärm utan att utlösa några händelser, för att sedan gå vidare till en annan skärm och utlösa händelser där.
- Användarna lämnar applikationen innan uppgifter är slutförda.

5.1.2.4 Efficency

För att mäta applikationernas efficiency användes antalet skärmvisningar som krävdes för att utföra en uppgift. För att identifiera att användarna utfört en viss

uppgift användes samma metod som i det andra fallet under avsnitt 5.1.2.2, med andra ord de sessioner där endast en händelse utlösts. Antalet skärmvisningar enligt Google Analytics jämfördes med det optimala antalet skärmvisnigar som krävdes för att utföra en specifik uppgift. Även möjligheterna till att beräkna lostness undersöktes. Formeln för att beräkna lostness hittas under avsnitt 3.2.9.1.

5.1.2.5 Learnability

Learnability i applikationerna mättes genom att segmentera användarna utefter hur många gånger de använt applikationen. Intervallen valdes ut så att det skulle vara ett någorlunda jämnt antal användare och sessioner i varje segment. I övrigt så användes samma metodik som i det andra tillvägagångssättet för time-on-task för att hämta ut data, se avsnitt 5.1.2.2.

5.2 Resultat

5.2.1 Implementationskontroll

Vid kontrollen av implementationen av Google Analytics i de tre applikationerna visade det sig att Android-versionerna var felimplementerade. Detta medförde att Android-versionerna av applikationerna inte kunde användas i mätningarna.

I iOS-versionen av Ladda Refill hittades inga problem med implementationen av Google Analytics. Däremot visade det sig att skärmvisningar var felimplementerade i Netcom Påfyll samt 3Fyll På. Exempelvis så registrerades startskärmen för de två applikationerna som fyra stycken olika skärmar i Google Analytics. Detta medförde att beteendeflödet inte gick att följa, genomsnittstiderna för de olika skärmarna var inkorrekta samt att antalet skärmar per session inte stämde överens med verkligheten. Däremot var resterande funktionalitet rätt implementerad, vilket innebar att det gick att använda både händelser och totala sessionstider för applikationerna.

Under genomgången av Google Analytics så visade det sig också att Ladda Refill samt Netcom Påfyll hade ett större antal händelser implementerade än 3Fyll På. Detta är inget fel i sig men påverkar hur väl det går att jämföra resultaten mellan applikationerna.

5.2.2 Bearbetning av insamlad data

För att få ett konsekvent resultat har endast data över en förutbestämd tidsperiod avlästs. För att undvika att få resultat från olika appversioner med olika gränssnittsdesign har mätningar endast gjorts på de som vid mättillfället var de senaste appversionerna. Alla mätningar är gjorda på iOS-versionerna av applikationerna.

I implementationen av Google Analytics som fanns att tillgå var tre stycken uppgifter gemensamma för de tre applikationerna. Dessa tre uppgifter var:

- Genomföra ett köp (även kallat en laddning) - Användarna måste bestämma laddningspremie, vilket telefonnummer de vill ladda på, vilket bankkort de ska använda samt ange sin pinkod. Dessa saker görs på olika skärmar och kräver således flera steg.
- Registrera en ny användare - Precis som i föregående uppgift så kräver denna uppgift att man anger information på flera skärmar och kräver således också flera steg.
- Lägga till ett nytt telefonnummer - För denna uppgift anges all information på en skärm och kräver således endast ett steg.

Den tredje uppgiften, lägga till ett nytt telefonnummer, gick inte att applicera på applikationen 3Fyll På. Detta eftersom det inte fanns någon implementerad händelse för uppgiften, men eftersom det var en väsentlig funktion och händelser fanns implementerade i de två andra applikationerna så inkluderades uppgiften i undersökningen.

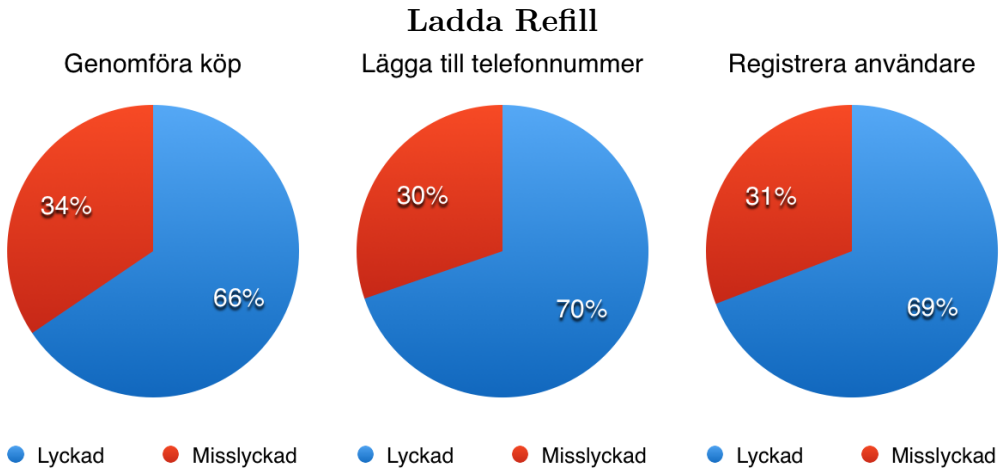
Data som var möjlig att utläsa för dessa uppgifter presenteras nedan. Då de tre applikationerna har ett varierande antal användare baseras uppgifterna på ett varierande antal sessioner.

5.2.2.1 Task Success

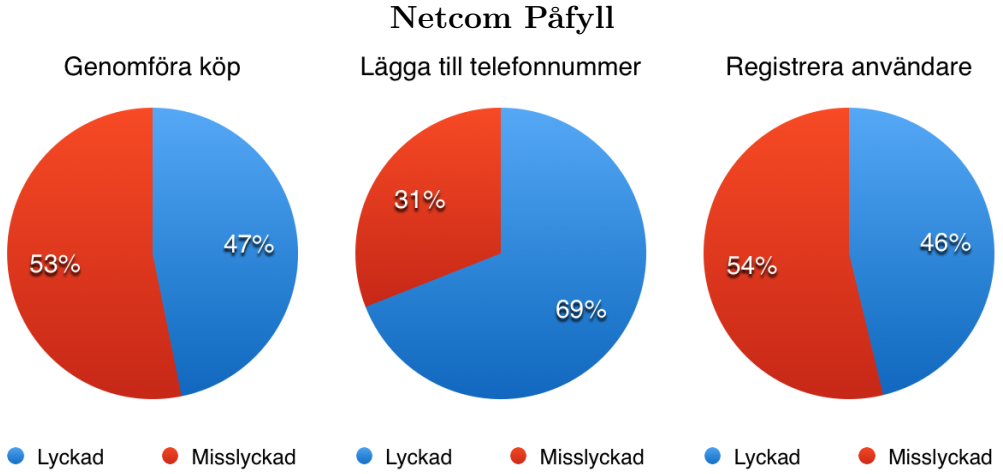
I figur 5.1-5.3 visas resultatet av data som gått att utläsa med hjälp av Google Analytics för de tre olika uppgifterna kring task success. I implementationen av Google Analytics fanns det händelser som utlöstes varje gång en användare klickade på den sista knappen som krävdes för att utföra en uppgift och således baseras data på detta. En genomförd uppgift kan därför endast ses som lyckad eller misslyckad. Att en uppgift blir misslyckad innebär att användaren fyllt i data (telefonnummer, kortnummer, pinkod etcetera) som inte är korrekt, alternativt att data gått förlorad under tiden som den skickas till servern. Data

visar inte om användarna påbörjat en uppgift för att sedan avbryta den under utförandet, utan kan endast med säkerhet påvisa att felaktig data mottagits på serversidan.

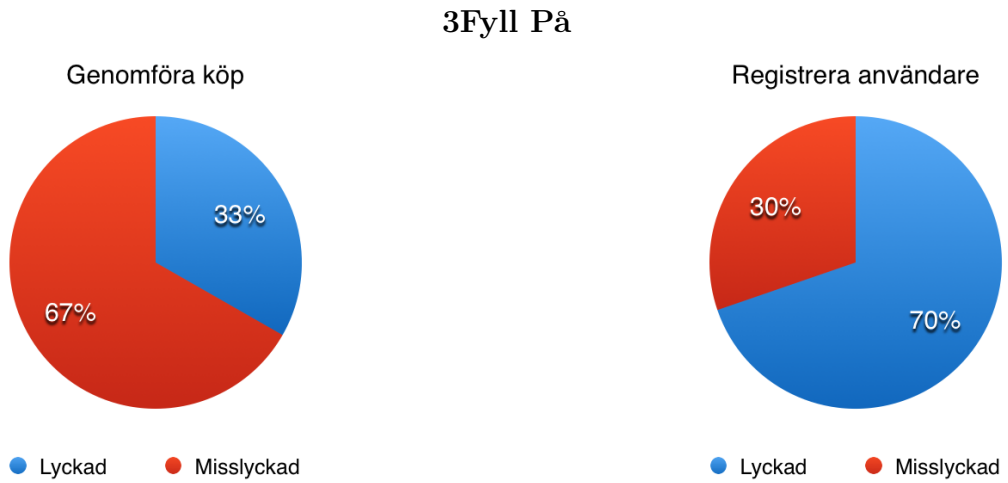
Det saknades en händelse i Google Analytics för att registrera användare i 3Fyll på-applikationen och således kunde inget resultat utläsas för den uppgiften.



Figur 5.1: Visar procentandelen lyckade respektive misslyckade försök för de tre uppgifterna i Ladda Refill. Genomföra köp baseras på 22742 sessioner, Lägga till telefonnummer baseras på 1477 sessioner och Registrera användare baseras på 5328 sessioner.



Figur 5.2: Visar procentandelen lyckade respektive misslyckade försök för de tre uppgifterna i Netcom Påfyll. Genomföra köp baseras på 6353 sessioner, Lägga till telefonnummer baseras på 272 sessioner och Registrera användare baseras på 260 sessioner.



Figur 5.3: Visar procentandelen lyckade respektive misslyckade försök för de tre uppgifterna i 3Fyll På. Genomföra köp baseras på 2102 sessioner och Registrera användare baseras på 1201 sessioner.

5.2.2.2 Time-on-task

I tabell 5.1-5.4 visas den data som gått att utläsa med hjälp av Google Analytics för de tre olika uppgifterna för time-on-task. Data har segmenterats så att endast de sessioner där användarna lyckats genomföra uppgifterna presenteras. För att ta bort extremvärden har ett trunkerat medelvärde där 5 % av ändarna tagits bort använts. Detta innebär att 5 % av de användare som tagit längst tid på sig och 5 % av de användare som tagit kortast tid på sig har uteslutits från medelvärdet. Detta har gjorts på alla uträkningar i detta avsnitt.

Det gick att avläsa time-on-task genom att segmentera data på två olika sätt. I tabell 5.1-5.3 samt figur 5.4 visas resultatet för den första metoden, där har data utlästs genom att titta på de sessioner där endast en specifik händelse som motsvarar en av de tre uppgifterna utlösts. Med andra ord kan användarna ha besökt fler skärmar än vad som krävs föra att genomföra uppgiften men ingen ytterligare händelse har utlösts.

I tabell 5.4 visas resultatet för den andra metoden. I detta fall har data utlästs genom att titta på användare som gått den optimala vägen för att genomföra uppgifterna. Med andra ord så besöker användarna endast de skärmar som krävs för att genomföra uppgifterna och de besöker skärmarna i rätt ordning. När uppgiften är slutförd så lämnar de applikationen. Eftersom skärmvisningar i Google Analytics var felimplementerade i två av applikationerna gick det endast att utläsa ett resultat för Ladda Refill.

Tabell 5.1: Visar den genomsnittliga tiden i sekunder som det tar för användarna att genomföra de tre uppgifterna i Ladda Refill. Genomföra köp baseras på 10323 sessioner, Lägga till telefonnummer på 204 sessioner och Registrera användare på 682 sessioner.

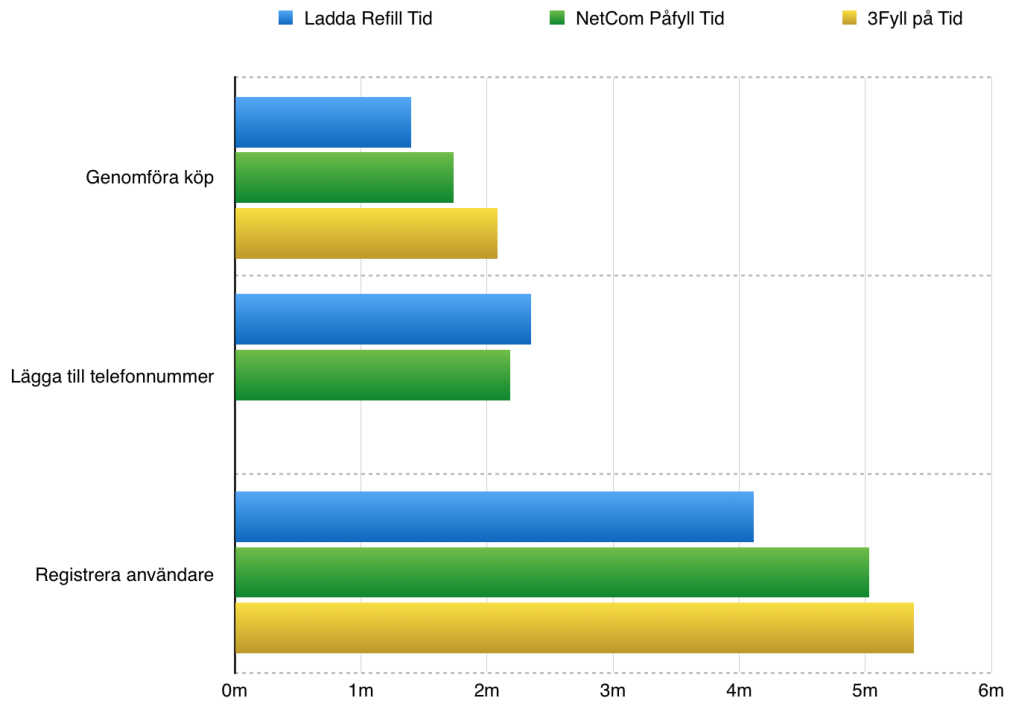
Ladda Refill	
	Tid
Genomföra köp	1m 24s
Lägga till telefonnummer	2m 21s
Registrera användare	4m 7s

Tabell 5.2: Visar den genomsnittliga tiden i sekunder som det tar för användarna att genomföra de tre uppgifterna i Netcom Påfyll. Genomföra köp baseras på 2579 sessioner, Lägga till telefonnummer på 31 sessioner och Registrera användare på 39 sessioner.

NetCom Påfyll	
	Tid
Genomföra köp	1m 44s
Lägga till telefonnummer	2m 11s
Registrera användare	5m 2s

Tabell 5.3: Visar den genomsnittliga tiden i sekunder som det tar för användarna att genomföra de tre uppgifterna i 3Fyll På. Genomföra köp baseras på 641 sessioner och Registrera användare på 548 sessioner.

3Fyll på	
	Tid
Genomföra köp	2m 5s
Lägga till telefonnummer	N/A
Registrera användare	5m 23s



Figur 5.4: Visar den genomsnittliga tiden i minuter som det tar för användarna att genomföra de tre uppgifterna i de tre applikationerna.

Tabell 5.4: Visar den genomsnittliga tiden i sekunder som det tar för användarna att slutföra de tre uppgifterna. Genomföra köp baseras på 2072 sessioner, Lägga till telefonnummer på 52 sessioner och Registrera användare på 29 sessioner.

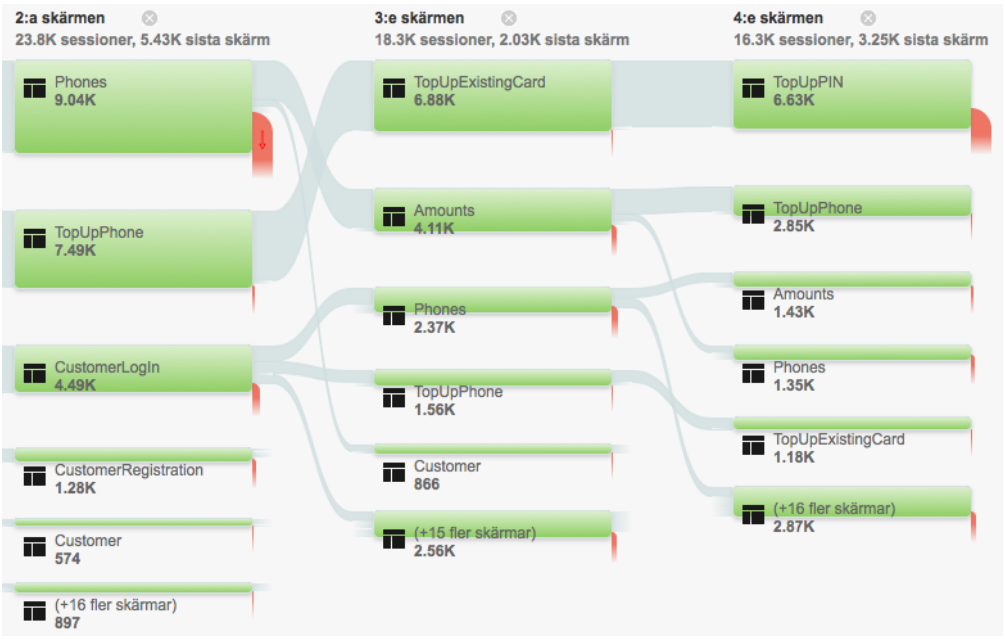
Ladda Refill	
	Tid
Genomföra köp	42s
Lägga till telefonnummer	1m 12s
Registrera användare	2m 13s

5.2.2.3 Errors

I figur 5.5 visas hur en del av ett flödesschema kan se ut. Dessa studerades för att hitta errors i applikationerna. Flödesscheman visar hur användarna rör sig i applikationerna samt vid vilka skärmar de väljer att lämna dem. Detta har studerats för att identifiera avvikelser i användarnas navigationsmönster. Fokus

har framförallt legat på att hitta mönster där användarna besöker skärmar i en annan ordning än den som behövs för att utföra en uppgift, samt vid vilken skärm som användarna väljer att lämna applikationen.

Det visade sig vara svårt att studera och tolka resultaten från ett flödesschema på det sätt som från början hade planerats. Detta resulterade i att endast en intressant avvikelse har kunnat identifierats. I figur 5.5 går det att se att en tredjedel av de användare som påbörjat en laddning lämnar applikationen vid den sista skärmen innan köpet genomförs, samt att nästan inga användare lämnar applikationen på skärmarna innan.



Figur 5.5: Visar en del av flödesschemat för Ladda Refill. I rutan längst upp till höger går det att se att ungefär en tredjedel av alla som besöker skärmen lämnar applikationen vid just denna skärm.

Som nämnts tidigare så registreras skärmvisningar på ett felaktigt sätt i Google Analytics för två av applikationerna, därav har det endast gått att utläsa ett resultat för Ladda Refill.

5.2.2.4 Efficiency

I tabell 5.5 visas resultatet som tagits fram för att beskriva Ladda Refill applikationens efficiency. Från tabellen kan en jämförelse göras mellan det optimala antalet skärmar som behövs för att utföra en uppgift och det genomsnittliga antalet skärmar som användarna besöker per session då de utför

en uppgift. Mätningarna gjordes endast på sessioner då en uppgift utfördes. Användarna utlöste således inte några andra händelser än de som behövdes för att utföra uppgiften.

Tabell 5.5: Antalet skärmvisningar för Ladda Refill applikationen, där optimala antalet skärmar motsvarar de minsta antalet skärmar som behövs för att utföra respektive uppgift. Det genomsnittliga antalet skärmar är antalet skärmar som användarna besöker under en session då respektive uppgift utförs.

	Ladda Refill	
	Optimalt antal	**Genomsnittligt antal**
Genomföra köp	5 Skärmar/ session	7,16 Skärmar/ session
Lägga till telefonnummer	4 Skärmar/ session	6,65 Skärmar/ session
Registrera användare	6 Skärmar/ session	8,14 Skärmar/ session

Som nämnts tidigare så registreras skärmvisningar på ett felaktigt sätt i Google Analytics för två av applikationerna, därav har det endast gått att utläsa ett resultat för Ladda Refill.

5.2.2.5 Learnability

I tabell 5.6-5.8 samt figur 5.6 visas resultatet av data som gått att utläsa med hjälp av Google Analytics för de tre olika uppgifterna kring learnability. Precis som i första fallet för att utläsa task succcess baseras data på sessioner där användarna endast utlöst en händelse, men de kan ha besökt andra skärmar än de som krävs för att utföra den specifika uppgiften. De har följaktligen endast utlöst totalt en händelse under hela sessionen. Även här har ett trunkerat medelvärde använts för att ta bort extremvärden. Detta innebär att 5 % av de användarna som tagit längst tid på sig och 5 % av de användarna som tagit kortast tid på sig har uteslutits från medelvärdet. Detta har gjorts på alla uträkningar i detta avsnitt.

Oftast registrerar en användare sig endast en gång och således är det inte intressant att titta på learnability för att registrera en ny användare. Därmed finns det inget resultat att presentera för denna uppgift. Att lägga till ett telefonnummer är också en handling som utförs ganska sällan och det fanns inte

tillräckligt med insamlad data för att presentera ett resultat för de fyra intervallen i någon av applikationerna. Således finns det inget resultat att presentera för denna uppgift heller. Därmed presenteras endast data för att genomföra ett köp.

Tabell 5.6: Visar den genomsnittliga tiden i sekunder för att genomföra ett köp i Ladda Refill. Antal sessioner är det totala antalet sessioner som användarna har haft i applikationen vid mättillfället. Intervallet 1-10 sessioner baseras på 3691 sessioner, 11-20 på 1891 sessioner, 21-30 på 1007 sessioner och >30 baseras på 2661 sessioner.

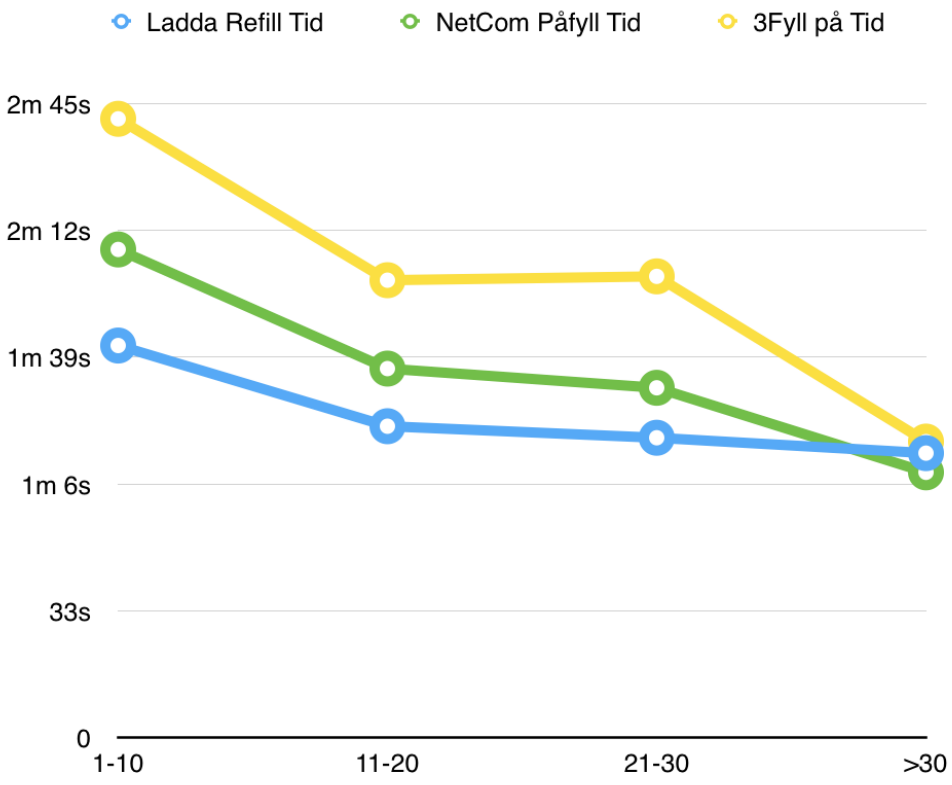
Ladda Refill	
Antal Sessioner	**Tid**
1-10	1m 42s
11-20	1m 21s
21-30	1m 18s
>30	1m 14s

Tabell 5.7: Visar den genomsnittliga tiden i sekunder för att genomföra ett köp i Netcom Påfyll. Antal sessioner är det totala antalet sessioner som användarna har haft i applikationen vid mättillfället. Intervallet 1-10 sessioner baseras på 531 sessioner, 11-20 på 1255 sessioner, 21-30 på 568 sessioner och >30 baseras på 292 sessioner.

NetCom Påfyll	
Antal Sessioner	**Tid**
1-10	2m 7s
11-20	1m 36s
21-30	1m 31s
>30	1m 9s

Tabell 5.8: Visar den genomsnittliga tiden i sekunder för att genomföra ett köp i 3Fyll På. Antal sessioner är det totala antalet sessioner som användarna har haft i applikationen vid mättillfället. Intervallet 1-10 sessioner baseras på 283 sessioner, 11-20 på 157 sessioner, 21-30 på 96 sessioner och >30 baseras på 105 sessioner.

3Fyll på	
Antal Sessioner	**Tid**
1-10	2m 41s
11-20	1m 59s
21-30	2m
>30	1m 17s



Figur 5.6: Visar learnability-kurvan för uppgiften genomföra ett köp för de tre applikationerna. På y-axeln visas tiden i minuter och sekunder, på x-axeln visas intervall för det totala antalet sessioner som användarna haft vid mättillfället.

6 | Användartester

6.1 Metod

Syftet med användartesterna var att undersöka huruvida de på ett bra sätt kunde komplettera det resultat som ges av Google Analytics. Som beskrivs under avsnitt 5.2.1 så saknades viss funktionalitet i implementationen i två av applikationerna, därför togs beslutet att endast utföra användartester på den applikation som hade full funktionalitet, vilket var Ladda Refill. Detta då det ansågs att det inte skulle gå att genomföra en ordentlig jämförelse av resultaten i de två andra applikationerna och att det därför var bättre att lägga all fokus på en applikation.

Användartesterna utfördes med fem testdeltagare. Samtliga var studenter och hade sedan tidigare god vana av mobila applikationer. Alla testdeltagare tillhörde applikationens målgrupp och två av dem hade använt applikationen tidigare. Utöver dessa användes också en testpilot för att genomföra en testgenomgång av testet.

Användartesterna hade för avseende att svara på följande frågor:

- Hur lätt är det för användarna att genomföra de givna uppgifterna?
- Vilken väg tar användarna då de genomför de givna uppgifterna?
- Vilka hinder stöter användarna på då de utför de givna uppgifterna?
- Hur lång tid tar användarna på sig att utföra de givna uppgifterna?
- Hur känner användarna kring tiden det tar att slutföra en uppgift?

Alla testdeltagarna utförde samma uppgifter under testerna. Insamlad data bestod av tiden det tog att utföra uppgifterna, antalet fel som begicks, andelen användare som lyckades/misslyckades med att utföra de givna uppgifterna samt kvalitativ data om användarnas upplevelser vid användning av applikationen.

Under testerna användes metoden ”Within-subjects design” där varje testdeltagare utförde samtliga uppgifter på egen hand. För att minimera effekterna av ”transfer of learning” utförde testdeltagarna uppgifterna i olika ordning. För att alla testdeltagare skulle få samma uppgiftsbeskrivning skrevs uppgifterna ner på ett papper som delades ut vid testets början.

Under testerna användes så kallade “sit-by sessioner”, vilket innebär att testledaren sitter i samma rum som testdeltagaren under testet. Testledaren började med att förklara bakgrunden till testet samt genomförde en kort bakgrundsintervju. Även en observatör var närvarande vid testet, med uppgift att anteckna vad som hände under utförandet. Efter det att testdeltagaren utfört de uppgifter som innefattades av testet genomfördes en intervju. Intervjun bestod av både förbestämda frågor för att ta reda på testdeltagarens subjektiva åsikt kring applikationen, och icke förbestämda frågor för att reda ut varför vissa problem uppstod under testets gång.

Testsessionerna varade i totalt ca 35 minuter. De första 5 minuterna var avsatta till att:

- Notera testdeltagarens bakgrundsinformation.
- Förklara syftet med testet.
- Kort beskrivning av applikationen.
- Förklara hur testet kommer att gå till samt moderatorernas roller.
- Kontrollera om testdeltagaren har några frågor innan testet börjar.

Det tog sedan ca 20 minuter för testdeltagarna att utföra uppgifterna. Dessa utfördes på en mobiltelefon som testdeltagarna blev tilldelade. På mobiltelefonen fanns en förinstallerad testapplikation och alla deltagare utgick från applikationens startskärm. De fick i uppdrag att utföra följande uppgifter:

Uppgift A - Lägga till ett nytt betalkort
Kriterier för lyckat genomförande: Testdeltagarna hittar knappen för att lägga till betalkort och anger korrekt kortnummer, utgångsdatum och CVV-nummer.

Uppgift B - Ta bort ett betalkort
Tillstånd: Det finns ett inlagt betalkort
Kriterier för lyckat genomförande: Testdeltagaren
klickar på knappen för att editera betalkort, väljer ta bort och bekräftar valet.

Uppgift C - Lägga till ett nytt telefonnummer
Kriterier för lyckat genomförande: Testdeltagaren anger rätt telefonnummer och rätt alias.

Uppgift D - Utföra ett köp av en premie med ett inlagt telefonnummer
Tillstånd: Det finns ett antal telefonnummer tillagda i applikationen.
Kriterier för lyckat genomförande: Testdeltagaren hittar rätt premie, väljer rätt telefonnummer och anger rätt pinkod.

Uppgift E - Editera ett redan tidigare inlagt telefonnummer
Tillstånd: Det finns ett antal telefonnummer tillagda i applikationen.
Kriterier för lyckat genomförande: Testdeltagaren hittar editera-knappen och väljer att editera rätt telefonnummer.

Uppgift F - Utföra ett köp av en premie med ett icke inlagt telefonnummer
Kriterier för lyckat genomförande: Testdeltagaren hittar rätt premie, anger rätt telefonnummer och anger rätt pinkod.

Uppgift G - Logga ut ur applikationen
Kriterier för lyckat genomförande: Testdeltagaren hittar knappen för att logga ut från applikation.

Uppgift H - Registrera en ny användare
Kriterier för lyckat genomförande: Testdeltagaren lyckas fylla i rätt information.

Under de sista 10 minuterna intervjuades testdeltagarna. Intervjun berörde följande punkter:

- Vad testdeltagarna tyckte om tiden det tar för att utföra uppgifter.
- Vad testdeltagarna tyckte om användarupplevelsen i allmänhet.
- Testdeltagarnas uppfattning om hur lätt det är att utföra uppgifterna.
- Moment eller element som användarna upplever som problematiska.
- Moment eller element som användarna gillar.
- Hur lätt användarna har för att förstå applikationens funktionalitet.

6.2 Resultat

Under testerna observerades testdeltagarnas interaktion med gränssnittet samt så mättes tiden för att slutföra uppgifterna. Relativt få användarfel upptäcktes

under testerna och i listan nedan presenteras en sammanfattning av observatörens anteckningar från testerna. Medeltiderna för varje uppgift presenteras i tabell 6.1.

Uppgift A - Lägga till ett nytt betalkort
Samtliga testdeltagare klarade utan större problem att navigera till rätt meny. Däremot hade tre av testdeltagarna problem med knappen för att lägga till ett nytt betalkort. Knappen för att lägga till ett betalkort (ett plustecken) är placerat i textfältet för kortnumret, därav trodde testdeltagarna att hela fältet var klickbart och det krävdes således flera försök innan de insåg att endast knappen var klickbar.

Uppgift B - Ta bort ett betalkort
Designen för att lägga till och ta bort betalkort är densamma och vissa av testdeltagarna upplevde således samma problem som för uppgift A. Då testdeltagarna gjorde uppgifterna i olika ordning hade vissa problem med uppgift A och vissa med uppgift B, det berodde helt enkelt på vilken uppgift som testdeltagarna gjorde först.

Uppgift C - Lägga till ett nytt telefonnummer
Samtliga testdeltagare klarade utan problem att navigera till rätt meny. Däremot hade två testdeltagare problem med att hitta knappen för att lägga till ett telefonnummer då det krävs en slide-rörelse före få fram den, se figur 1.2. Samtliga testdeltagare klarade att ange rätt information på rätt ställen.

Uppgift D - Utföra ett köp av en premie med ett inlagt telefonnummer
Tre av testdeltagarna hade problem med denna uppgift då de inte kunde hitta upp/ner-pilarna för att öka respektive minska summorna på laddningspremierna, se figur 1.1. Två av de tre testdeltagarna hittade pilarna på egen hand efter en tids letande. Den tredje testdeltagaren gav efter en stund upp och frågade om hjälp. Ingen av testdeltagarna hade däremot problem med att ange rätt information på rätt ställe.

Uppgift E - Editera ett redan tidigare inlagt telefonnummer
Samtliga testdeltagare navigerade direkt till rätt meny och hittade snabbt rätt telefonnummer, de hade heller inga problem med att hitta knappen för att redigera telefonnummer. Däremot hade två av användarna problem med att klicka på den och det krävdes några försök innan de lyckades. Knappen kan ses i figur 1.2.

Uppgift F - Utföra ett köp av en premie med ett icke inlagt telefonnummer
Samtliga testdeltagare klarade denna uppgift utan problem. De hittade

snabbt rätt laddningspremie och angav rätt information på rätt ställen. Skillnande mellan denna uppgift och uppgift D är att användarna här inte behöver klicka på upp/ner-pilarna eftersom laddningspremien de söker har som standard rätt summa för uppgiften.

Uppgift G - Logga ut ur applikationen
Samtliga testdeltagare navigerade direkt till rätt meny och hade inga problem att hitta knappen.

Uppgift H - Registrera en ny användare
Samtliga testdeltagare hittade snabbt knappen för att lägga till användare och hade sedan inga problem att ange rätt information på efterföljande skärmar.

Tabell 6.1: Visar medeltiden i sekunder som det tog för testpersonerna att utföra uppgifterna.

Ladda Refill	
Uppgift	**Tid**
A	59s
B	31s
C	30s
D	1m 16s
E	22s
F	42s
G	4s
H	1m 10s

Det var endast uppgift D som inte fick 100% task success eftersom en av testdeltagarna behövde fråga om hjälp. Övriga uppgifter kunde testdeltagarna genomföra på egen hand. Samtliga testdeltagare var överlag nöjda med applikationens användarupplevelse och ingen av dem ansåg att någon av uppgifterna tog för lång tid att genomföra. De tyckte gränssnittets uppbyggnad var bra, men det var svårt att få en överblick över vilka laddningspremier som fanns tillgängliga. Några av användarna upplevde även att laddningspremierna

kunde kategoriserats bättre, för att på så sätt skapa en bättre överblick. De som använde applikationen för första gången ansåg att de skulle ha lättare att hitta rätt i gränssnittet nästa gång de använde applikationen.

7 | Diskussion

7.1 Google Analytics och Performance metrics

Att mäta performance metrics för specifika uppgifter med hjälp av data insamlad med Google Analytics kan vara problematiskt. Till stor del beror detta på att det inte går att veta användarnas intentioner när de öppnar applikationen. Det går till exempel inte att veta om användarens tanke när applikationen startas är att endast kolla sitt saldo, att lägga till ett nytt telefonnummer eller genomföra ett köp. Det går heller inte att veta när användaren bestämmer sig för att påbörja en viss uppgift. Möjligheten finns att användaren redan innan applikationen startar bestämt sig för vilka uppgifter som ska utföras, men det skulle också kunna vara så att en uppgift leder till att flera utförs. Användare kan också ändra sig under utförandet och bestämma sig för att göra en annan uppgift. Detta är ett återkommande problem när det gäller mätning av performance metrics i Google Analytics.

7.1.1 Task success

För att veta att en specifik uppgift har utförts, eller att åtminstone ett försök till att genomföra en specifik uppgift har utförts, går det i Google Analytics att använda sig av händelser. I implementationerna för de tre applikationerna så registreras en händelse när användarna klickar på den sista knappen som krävs för att genomföra en uppgift. Vissa av uppgifterna kräver att användarna både besöker och anger information på flera skärmar och det är först när användarna klickar på den sista knappen som all denna information skickas till servern och en händelse får en etikett som antingen motsvarar ett lyckat eller misslyckat försök. Detta innebär att det inte går med hjälp av händelser se när en uppgift påbörjas eftersom de endast motsvarar det absolut sista steget i en uppgift. Detta medför i sin tur att det inte går att se om användarna avbryter ett köp innan den sista

knapptryckningen och således går det heller inte att utläsa levels of success. Det är heller inte möjligt att se specifika skärmar där användarna fastnar. Att se att användare har angett fel information är möjligt men inte var och vilken information som är inkorrekt. För att upptäcka skärmar som är problematiska går det istället att använda funktionen beteendeflöde för att se vid vilka skärmar som användarna lämnar applikationen, alternativt titta på den genomsnittliga tiden som spenderas på varje skärm. Är tiden som spenderas på en viss skärm hög, kan det vara en indikation på att något är fel.

Generellt så definieras en uppgift som misslyckad på något av följande sätt:

- När användarna når en punkt då de själva anser att de inte kan slutföra uppgiften och behöver hjälp.
- Användarna får tre försök på sig att genomföra uppgiften, lyckas inte användarna genomföra uppgiften på dessa försök så anses uppgiften vara misslyckad. Ett försök kan definieras på olika sätt och kan exempelvis vara att användarna trycker på fel knapp eller navigerar fel i applikationen.
- Om den totala tiden det tar för att genomföra uppgiften passerar en tidströskel så ses försöket som misslyckat [16].

I det första fallet har man inte tillgång till användarnas åsikt och på så sätt går det inte veta när de själva anser att de inte kommer att klara av att genomföra uppgiften. I det andra fallet är det svårt att definiera ett försök som är mätbart med Google Analytics. Möjligheten till att mäta det tredje fallet diskuteras under avsnitt 7.1.2. Gemensamt för alla fallen är att användarna befinner sig på distans. Detta medför att det inte finns någon möjlighet att observera vad användarna gör eller ta del av deras åsikter. Därmed kan det vara problematiskt att definiera när en uppgift ska räknas som misslyckad med hjälp av mätningar från Google Analytics.

Det blir problematiskt att mäta task sucess för uppgifter som inte kräver att användaren aktivt interagerar med interaktiva element i gränssnittet. Om exempelvis mätningar för task success ska göras för en uppgift som att kontrollera saldo för ett av användarens inlagda telefonnummer så krävs endast en knapptryckning till rätt huvudmeny. Därefter handlar det om att kunna avläsa rätt saldo till rätt nummer. Att användarna har navigerat till rätt meny går att registrera som en händelse i Google Analytics, men att användaren läst av rätt saldo eller ens kunnat läsa av något saldo kan inte registreras som en händelse eftersom användaren inte interagerar med något interaktivt element.

I implementationen av händelser i Google Analytics så grupperas alla användares sessioner tillsammans. Detta gör att det inte är möjligt att ta fram antalet

gånger en godtycklig användare har utlöst en händelse. På så sätt finns det inget sätt att se om antalet utlösta händelser är jämnt fördelade över alla användare eller om ett fåtal användare har stått för en stor mängd händelser. På så sätt kan ett fåtal användare stå för en stor del av alla misslyckade händelser utan att utvärderaren vet om det.

7.1.2 Time-on-task

Det största problemet med att använda Google Analytics för att mäta time-on-task är att det inte går att veta vad användarna har för intentioner när de startar en applikation. Det går därför inte att avgöra när användarna påbörjar en uppgift, men genom att implementera händelser för alla uppgifter som går att utföra i applikationen kan det möjliga spannet för när en uppgift påbörjas smalnas av. För att få fram den totala tiden för att utföra en uppgift används sessionstiden. Genom att studera användarnas sessionstider då de under sessionen endast utlöst en specifik händelse kan tiden det tar att utföra uppgiften fås. Det går fortfarande inte att veta vad användarna hade för intentioner när de startade applikationen, men eftersom inga andra händelser utlöses går det att antaga att det endast var den specifika uppgiften som skulle utföras. Google Analytics har även stöd för att skapa segment med en sekvens av händelser. Detta gör det möjligt att mäta tiden för uppgifter som kräver att flera händelser utlöses i en förutbestämd följd.

Applikationerna som undersökts hade en varierande grad av händelser implementerade. Tyvärr fanns ingen möjlighet att implementera fler händelser vilket medfört att mätningarna i rapporten inte gjorts under optimala förhållanden. Telia Refill och Netcom Påfyll hade flest händelser implementerade, medan 3Fyll På hade något färre. Detta kan vara en förklaring till varför 3Fyll På har högre tider i mätningarna än de andra två applikationerna. För att öka säkerheten i mätningarna skulle fler händelser behöva implementeras i alla applikationer men framförallt i 3Fyll På.

För att öka säkerheten utifrån de förutsättningar som funnits har användarnas beteenden studerats med hjälp av Google Analytics beteendeflödes-funktion. Där har det varit möjligt att kontrollera så att merparten av de användare som använts i mätningarna följt de mönster som är tänkta att följas för att utföra de specifika uppgifterna. Trots det varierar säkerheten i resultaten för de olika applikationerna och jämförelser mellan dem blir något missvisande.

I det andra fallet för att utläsa mätresultat för time-on-task har data segmenterats så att sessioner där användarna besökt precis de skärmar som

behövts för att utföra en specifik uppgift visas. Under dessa sessioner har användarna besökt skärmarna i rätt ordning och efter uppgiftens slut har de lämnat applikationen direkt. Således har användarna gått den optimala vägen, vilket i sin tur innebär ett minimalt antal skärmvisningar för att genomföra uppgifterna. Även i detta fall används den totala sessionstiden för att få ut tiden för en uppgift men eftersom användarna lämnar applikationen direkt efter att uppgiften är genomförd går det att antaga att man kommer närmare det riktiga värdet för time-on-task. Däremot så tar man endast hänsyn till användare som går precis rätt väg. Vilket förmodligen till stor del bara är erfarna användare som utfört uppgifterna ett antal gånger tidigare.

Ett gemensamt problem för de två metoderna ovan är att de endast tar hänsyn till de sessioner där bara en uppgift har utförts. De tillåter inte att användare utför flera uppgifter vilket gör att en stor mängd sessioner inte tas med i beräkningarna. Har en användare exempelvis lagt till ett telefonnummer och genomfört ett köp under samma session så räknas inte tiden för det genomförda köpet med i resultatet.

Under time-on-task blir det tydligt att Google Analytics kan användas för att jämföra resultat mellan olika applikationer med likvärdig funktionalitet eller mellan olika versioner av samma applikation. Detta gäller generellt sätt för kvantitativ data eftersom dessa går att kvantifiera och analysera matematiskt.

7.1.3 Learnability

För att mäta learnabillity i applikationerna har samma metod som för time-on-task använts, med skillnaden att segmentering också gjorts på det totala antalet sessioner som användarna har haft vid mättillfället. Detta gjorde det möjligt att dela upp användarna i olika grupper beroende på hur många gånger de använt applikationerna, för att sedan kunna studera de genomsnittliga sessionstiderna för de olika grupperna.

Eftersom samma metod använts för att mäta tiderna i både time-on-task och learnability finns också samma problem med säkerheten kring resultaten. Trots det kan man spekulera i förändringarna i sessionstiderna mellan de olika grupperingarna. Från resultatet kan man se att för alla de tre applikationerna så minskar den genomsnittliga sessionstiden då det totala antalet sessioner som användarna har haft ökar. Det går också att se att Telia Refill applikationens kurva i figur 5.6 planar ut tidigare än de andra två applikationernas kurvor, vilket skulle kunna indikera att användarna snabbare lär sig att använda den applikationen.

7.1.4 Errors

Fyra scenarion togs fram för att identifiera errors med hjälp av beteendeflödet i Google Analytics:

- Användarna besöker skärmar i en annan ordning än den som är tänkt för att utföra uppgiften.
- Användarna går fram och tillbaka mellan olika skärmar.
- Användarna besöker en viss skärm utan att utlösa några händelser, för att sedan gå vidare till en annan skärm och utlösa händelser där.
- Användarna lämnar applikationen innan uppgifter är slutförda.

Det visade sig vara svårt att identifiera errors i applikationen genom att använda dessa scenarion och endast en indikation på att ett error uppstått upptäcktes. Detta gjordes genom att studera vilka skärmar som användarna lämnade applikationen från. I figur 5.5 går det att se att i ungefär en tredjedel av alla sessioner där användarna besöker den sista skärmen för att genomföra ett köp så väljer de att lämna applikationen innan köpet är genomfört. Vilket skulle kunna innebära att där finns ett användbarhetsproblem vid denna skärm. I detta fall fanns det däremot en logisk förklaring till varför det skulle kunna vara så. Detta eftersom det är vid denna skärm som användarna anger sin laddkod. Då köp inte är något som användarna genomför dagligen kan det vara lätt att glömma bort sin laddkod och detta medför att användarna lämnar applikationen för att ta reda på vad koden är. Även om det i detta fall inte nödvändigtvis handlade om ett användbarhetsproblem så är det ändå en indikation på att Google Analytics till viss del går att använda för att identifiera errors.

Vanligtvis vid mätningar av errors beräknas det totala antalet errors som användarna gör för varje uppgift. Som beskrevs under avsnitt 7.1.1 så grupperas alla användares sessioner tillsammans vilket gör att det inte går att titta på godtyckliga användares sessioner. Därmed blir det svårt att identifiera hur många errors som har skett under en specifik session.

7.1.5 Efficiency

Att använda sig av en jämförelse mellan antalet skärmar som krävs för att utföra en uppgift och det genomsnittliga antalet skärmar som användarna besöker är en relativt simpel metod för att mäta efficiency. Det går att argumentera för att resultatet i det här specifika fallet inte säger så mycket om applikationen har hög

efficiency eller ej, vilket delvis stämmer. Hade applikationerna haft fler händelser implementerade hade resultatet varit desto mer intetsägande. Detta eftersom användarna inte skulle kunna besöka så många andra skärmar än de som behövs för att utföra uppgiften utan att utlösa andra händelser. Vilket i sin tur leder till att de inte tas med i beräkningarna.

Det fanns också förhoppningar om att beräkna värdet för lostness i de tre applikationerna. Detta skulle göras med formeln som presenteras under avsnitt 3.2.9.1. Det visade sig dock att det i Google Analytics inte gick att ta fram alla parametrar som behövdes och således kunde inget värde tas fram.

7.1.6 Möjliga förbättringar av implementationen

Genom att implementera fler händelser i applikationerna hade resultaten kunnat bli både säkrare och mer omfattande.

Med fler händelser hade det till exempel varit möjligt att identifiera errors på ytterligare ett sätt. Som tidigare nämnts så kräver uppgiften för att genomföra ett köp att användarna besöker flera skärmar. På dessa skärmar så finns det en stäng-knapp som avbryter köpet och en bakåt-knapp för att gå tillbaka till föregående skärm. Det hade varit möjligt att registrera varje knapptryckning som en händelse för att på så sätt få en bättre överblick över användarnas beteende under utförandet av uppgiften. Om användarna skulle klicka oftare på dessa knappar än vad köp genomförs kan det ge indikationer på att något är fel.

Ett annat exempel där fler händelser skulle förbättra resultaten är i mätningar för time-on-task och learnability. Genom att ha fler händelser implementerade hade det varit lättare att försäkra sig om att användarna endast utfört just den uppgift som mätningarna sker på. Anledningen till att det blir lättare med fler händelser implementerade är att det skulle finnas fler händelser som kan utlösas om användarna avviker från det tilltänkta beteendet. Där det tilltänkta beteendet definieras som det som krävs för att utföra en specifik uppgift.

Utöver de metoder som använts i denna studie så erbjuder Google Analytics ytterligare ett alternativ för att mäta tiden det tar för en användare att utföra en uppgift. Funktionalitet finns i Google Analytics för att tilldela värden till händelser [5]. Dessa värden skulle kunna motsvara en tidsenhet och skulle då kunna implementeras så att de mäter tiden det tar för att utföra en uppgift. Med denna metod ges möjligheten att vara mer specifik i sin tidtagning. Detta eftersom större frihet ges till att själv avgöra när tidtagningen ska börja och när den ska sluta. Metoderna som använts i denna studie är beroende av den totala

sessionstiden, detta medför att mätningarna av tiden alltid startar när sessionen startar och slutar när sessionen avslutas.

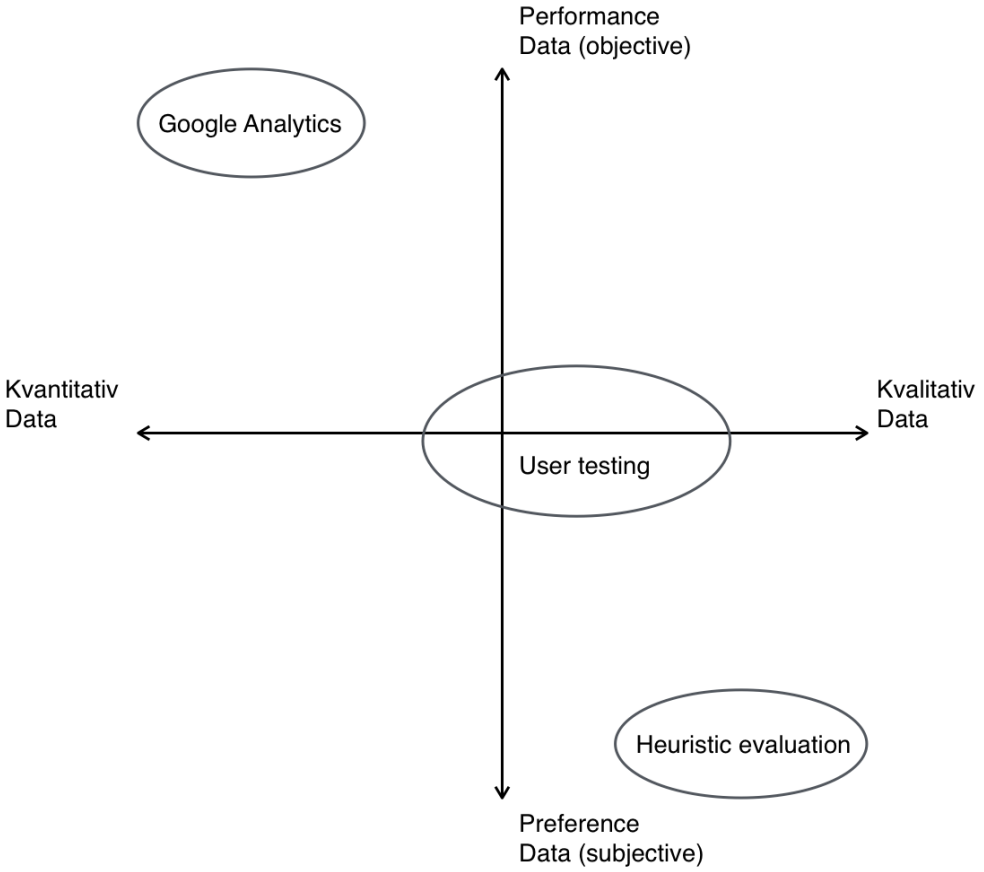
7.2 Jämförelse av metoder för användbarhetsutvärdering

Ett av huvudmålen med examensarbetet var att undersöka vilka metoder som vid behov kunde komplettera resultatet från Google Analytics. För att dra slutsatser kring detta behövdes resultaten från de tre utvärderingsmetoderna jämföras.

Ett av syftena med den heuristiska utvärderingen var att identifiera var i applikationen som det skulle kunna finnas användbarhetsproblem och således var användarna kunde göra fel. Få användbarhetsproblem upptäcktes i den heuristiska utvärderingen och inga av dessa kunde identifieras i Google Analytics. Till exempel så ansåg utvärderarna i den heuristiska utvärderingen att knappar för att uppdatera och ta bort telefonnummer låg nära varandra, vilket skulle kunna leda till att användare råkar trycka på fel knapp. Att detta skulle vara ett problem för användarna kunde inte påvisas med hjälp av Google Analytics.

Resultatet från den heuristiska utvärderingen baseras helt och hållet på utvärderarnas personliga åsikter och data är kvalitativ. Resultatet från Google Analytics baseras å sin sida på användarnas beteende och är kvantitativ. Metoderna står således nästan i motsats till varandra. Detta illustreras tydligt i figur 7.1. Då insamlad data är av två helt olika typer blir dessa svåra att jämföra. Kvalitativ data från den heuristiska utvärderingen är bättre på att svara på varför ett problem uppstår och hur det ska lösas medan kvantitativ data från Google Analytics är bättre svara på frågor som hur många eller hur mycket. Med detta sagt kan metoderna användas tillsammans för att samla in flera olika typer av data. Fördelen med att kombinera dessa metoder är att inga testpersoner behöver närvara eftersom data kan samlas in på distans.

De användbarhetsproblem som upptäcktes under användbarhetsutvärderingen tillhörde alla samma kategori och var relaterade till navigering i applikationen. Testanvändarna hade problem med att antingen hitta eller klicka på objektet de letade efter. Detta berodde antingen på dålig synlighet eller på en kombination av dålig synlighet och dålig affordance i form av klickbarhet. Dessa typer av användbarhetsproblem visade sig vara svåra att upptäcka i Google Analytics. Framförallt beror det på att användbarhetsproblemen i de flesta fall bara sker under användarnas första interaktion med applikationen. När användarna väl hittat knappen eller objektet de letar efter så har de inga problem att hitta dessa



Figur 7.1: Visar hur data insamlad med Google Analytics, användartester och heuristisk utvärdering förhåller sig till varandra.

vid nästa tillfälle de använder applikationen. För att leta efter dessa användbarhetsproblem i Google Analytics kan man studera förstagångsanvändare. Problemet med att studera dessa är att de i större utsträckning tenderar till att ha ett mer oförutsägbart beteende än erfarna användare. Nya användare testar gärna funktioner och navigerar runt i applikationen för att bekanta sig med gränssnittet. Det är därför väldigt svårt att förutspå avsikterna med deras handlingar. Vilket i sin tur leder till att användbarhetsproblemen blir svåra att identifiera i Google Analytics. En annan anledning till att användbarhetsproblemen var svåra att upptäcka berodde på att de inte fanns någon funktionalitet för se om användarna behöver flera försök för att lyckas klicka på en knapp. Detta beror på att Google Analytics bara registrerar händelser då användarna interagerar med interaktiva element systemet, detta sker inte om de till exempel klickar bredvid en knapp.

Resultaten för mätning av time-on-task för de olika uppgifterna varierade en del

mellan användartesterna och Google Analytics. Tiderna för att genomföra ett köp var relativt lika och det skiljde endast ett fåtal sekunder mellan de två metoderna, medan det för lägga till telefonnummer och registrera användare skiljde minuter. En förklaring till detta kan vara att utföra köp baseras på en mycket större mängd sessioner än de andra två uppgifterna och det finns således en större säkerhet i resultatet.

Vid användartester samlas ofta en kombination av kvantitativ och kvalitativ data in. I denna fallstudie bestod kvantitativ data av tider för att genomföra uppgifter samt observationer av vilka fel som begicks. Kvalitativ data bestod av användarnas åsikter och samlades in med hjälp av intervjuer. Således placerar sig användartester i detta fall nästan i mitten av axlarna på figur 7.1. Typen av insamlad data från användartester och Google Analytics står på så sätt inte i motsats till varandra som insamlad data från heuristisk utvärdering och Google Analytics gör. Det bör dock nämnas att användartester kan anpassas efter vilken typ av data som ska samlas in och kan således röra sig kring båda axlarna beroende på vilken datainsamlingsmetod som används.

En av fördelarna med Google Analytics kontra användartester är att insamlad data kommer från riktiga slutanvändare som är representativa för hela målgruppen och som använder produkten i en verklig miljö. Vid användartester sker alltid testningen i en artificiell situation och det är väldigt sällan som testpersonerna är representativa för hela målgruppen. Insamlad data från Google Analytics är å sin sida kvantitativ och objektiv och missar således helt och hållet användarnas subjektiva åsikter.

7.3 Metodkritik och försvar

Det kan argumenteras för att en studie av det här slaget inte är tillräckligt generaliserbar för att resultatet ska vara av värde, vilket är ett problem som gäller för fallstudier i allmänhet [10]. Det finns personer som ifrågasätter den här typen av studier eftersom de endast berör enstaka händelser eller enheter, vilket enligt dem kan leda till snedvridna eller skeva resultat [10]. För att undvika detta har det här examensarbetet grundats på tidigare forskning som genomförts inom området. Detta gör det möjligt att jämföra resultat för att på så vis påvisa att resultatet från examensarbetet sträcker sig över fler applikationer än de som använts i arbetets undersökningar och analyser. Vad som kan hämtas från dessa resultat och hur de kan användas för vidare forskning adresseras i kapitel 7 respektive kapitel 9.

En stor del av den litteratur som teorin grundar sig på, främst angående användbarhetsprinciper och användbarhetsutvärdering, har funnits under en längre tid. Däremot har många av de moderna publikationer som studien utgått ifrån hänvisat till denna litteratur vilket styrker dess relevans. Se till exempel ”Heuristic Evaluation of Mobile Usability: A Mapping Study” [42] där författarna kartlagt litteratur kring heuristisk utvärdering. Den äldre litteraturen har i första hand använts för att ge en generell förklaring av användbarhetsprinciper och användbarhetsutvärdering, men då den i viss mån inte är anpassad för mobila enheter har den kompletterats med moderna publikationer.

Skulle projektet sträckt sig över en längre tidsperiod hade mer tid kunnat läggas på checklistan till den heuristiska utvärderingen. Framförallt saknas en egen djupare undersökning inom området heuristiker för små skärmar med pekskärmsfunktion. Den tidigare forskning som finns inom tidigare berörda område och som utvärderingen bygger på upplevdes som aningen föråldrad, mycket pågrund av den snabba utveckling som skett inom området de senaste åren.

De tre personerna som utförde den heuristiska utvärderingen är inga användbarhetsexperter men har viss kunskap inom området. Då heuristisk utvärdering som metod är beroende av utvärderarnas kompetens inom området [23] kan resultaten komma att bli annorlunda om personer med annan kompetensnivå skulle utföra utvärderingen.

Under användartesterna användes 5 stycken försökspersoner. Alla tillhörde målgruppen för applikationerna men det var också en stor del av målgruppen som inte täcktes in. Man kan därför argumentera för att ett bredare urval skulle gjorts då försökspersoner valdes ut. För att på så vis kunna få en större blandning av till exempel kön, ålder och utbildning.

8 | Slutsats

Syftet med detta examensarbete var att besvara följande frågeställningar:

- Vilka slutsatser går att dra kring användbarhet och användarupplevelse med hjälp av användardata från Google Analytics?
- Är insamlad data tillräcklig för att kunna dra värdefulla slutsatser eller behöver den komplementeras med andra metoder och i så fall vilka?

De framkomna resultaten från studien över Google Analytics tyder på att det framför allt är händelser (se avsnitt 2.1.1) som kan utvinnas från mobila applikationer. Det ska dock nämnas att det kan finnas annan funktionalitet som kan vara värdefull i mer specifika situationer än de som undersökts i detta arbete.

Som helhet betraktat ger dessa resultat inte tillräckligt stöd för att rekommendera Google Analytics som ett verktyg för en omfattande användbarhetsutvärdering. Däremot indikerar resultaten av studien att det finns stor potential i verktyget. Speciellt vid situationer då man vill göra en jämförelse av en avgränsad del eller funktion, antingen mellan två likvärdiga applikationer eller mellan olika versioner av samma applikation. En närmare undersökning av händelsefunktionen skulle därför kunna bidraga till viktiga slutsatser om Google Analytics som ett verktyg för användbarhetsutvärdering. Detta kan dessutom göras med i sammanhanget väldigt lite resurser i form av arbetstid och pengar.

Resultaten från undersökningen av vilka metoder som kan användas som komplement till Google Analytics gav inga tydliga indikationer på att den ena metoden skulle vara bättre än den andra. Både heuristisk utvärdering och användartester har sina för- och nackdelar. Båda metoderna hittade dessutom användbarhetsbrister i applikationerna som inte Google Analytics kunde påvisa. Att ett bättre resultat fås genom att komplettera med någon av de två metoderna råder det inga tvivel om. Vilken metod som bör väljas är inte lika självklart utan beror mer på vilka resurser och förutsättningar som finns. Heuristisk utvärdering och användartester fungerar dessutom bättre i ett tidigare

skede av utvecklingsprocessen. Medan Google Analytics styrka ligger i att kunna göra mätningar då applikationen lanserats och används av slutanvändarna. Det ena behöver därför inte nödvändigtvis ses som ett komplement till de andra. En omfattande användbarhetsutvärdering kan göras utan Google Analytics under utvecklingen av applikationen. Google Analytics kan sedan användas vid förändringar i applikationen för att utvärdera enstaka funktioner eller avgränsade delar av systemet där förändringarna har skett.

9 | Fortsatt arbete

Ambitionerna med examensarbetet var att undersöka och ta fram metoder som var så pass allmänna att de skulle kunna appliceras på alla olika typer av mobila applikationer. Examensarbetet har dock fokuserat på tre mobila applikationer som kan placeras inom samma kategori. Således skulle det behövas ytterligare forskning och studier för att undersöka om det vore möjligt att applicera metodiken på andra typer av applikationer, exempelvis spel eller andra underhållningsapplikationer.

I arbetet så undersöktes det huruvida heuristisk utvärdering och användarstudier kunde på ett bra sätt komplettera resultatet från Google Analytics. Att testa fler än två metoder rymdes inte inom ramen för detta examensarbete, därför valdes dessa då de är de två mest använda metoderna. En tänkbar fortsättning på detta examensarbete hade varit att undersöka hur andra metoder för användbarhetsutvärdering kan komplettera resultatet från Google Analytics.

Under examensarbetets gång stod det klart att spårning av händelser var mest intressanta att titta på för att identifiera användbarhetsbrister med hjälp av Google Analytics. Tyvärr fanns det ingen möjlighet att implementera fler händelser i applikationerna än de som fanns att tillgå vid arbetets start. Under avsnitt 7.1.6 togs det fram förslag på hur fler händelser skulle kunna användas för att få ett mer omfattande resultat. Att genomföra en implementation av dessa hade varit önskvärt och är lämpligt för en framtida studie.

Litteraturförteckning

[1] Ariel. App Stores Growth Accelerates in 2014. http://blog.appfigures.com/app-stores-growth-accelerates-in-2014/, 2015-01-13. [Accessed: 19 mars 2015].

[2] AppBrain, Android statistics, Google Analytics. http://www.appbrain.com/stats/libraries/details/analytics/google-analytics, 2015-03-18. [Accessed: 19 mars 2015].

[3] Platform Overview. https://developers.google.com/analytics/devguides/platform, 2015-8-1. [Accessed: 11 februari 2015].

[4] J. Cutroni. Hits, Sessions & Users: Understanding Digital Analytics Data. http://cutroni.com/blog/2014/02/05/understanding-digital-analytics-data/, 2014-02-05. [Accessed: 2 mars 2015].

[5] Om händelser. https://support.google.com/analytics/answer/1033068?hl=sv. [Accessed: 15 april 2015].

[6] B. Clifton. *Advanced Web Metrics with Google Analytics*. John Wiley & Sons, Inc., 2012.

[7] J. Cutroni. *Google Analytics*. O'Reilly Media, Inc, 2010.

[8] Google Analytics for mobile website and apps. https://www.youtube.com/watch?v=nReGvqKidKE, 2012-04-05. [Accessed: 11 februari 2015].

[9] Om realtidsrapporter. https://support.google.com/analytics/answer/1638635?hl=sv. [Accessed: 09 april 2015].

[10] J. Bell. *Introduktion till forskningsmetodik*. Studentlitteratur, 1993.

[11] ISO 9241-11, Ergonomic requirements for office work with visual display terminals (VDTs).

[12] J. Nielsen. *Usability Engineering*. Morgan Kaufmann, 1993.

[13] J. Nielsen and D. A. Norman. The Definition of User Experience. `http://www.nngroup.com/articles/definition-user-experience/`. [Accessed: 6 februari 2015].

[14] E. Law, V. Roto, M. Hassenzahl, A. Vermeeren, and J. Kort. Understanding, scoping and defining user experience: A survey approach. *Proceedings of Human Factors in Computing Systems conference". CHI'09. Boston, MA, USA.*, pages 719–728, 2009.

[15] ISO 9241-210, Ergonomics of human-system interaction - part 210: Human-centred design for interactive systems.

[16] T. Tullis and W. Albert. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufmann, 2008.

[17] D. Lavery, G. Cockton, and M. P. Atkinson. Comparison of evaluation methods using structured usability problem reports. *Behaviour & Information Technology*, 16(4-5):246–266, 1997.

[18] D. A. Norman. *The design of everyday things*. Basic Book, 2002.

[19] B. Shneiderman and C. Plaisant. *Designing the User Interface*. Pearson, 2010.

[20] J. Nielsen and R. Mack. *Usability inspection methods*. John Wiley & Sons, Inc., 1994.

[21] R. G. Bias and D. J. Mayhew. *Cost-Justifying Usability, Second Edition: An Update for the Internet Age, Second Edition (Interactive Technologies)*. Morgan Kaufmann, 2005.

[22] A. Holzinger. Usability engineering methods for software developers. *Communications of the ACM*, 48(1):71–74, 2005.

[23] J. Nielsen. Finding usability problems through heuristic evaluation. *CHI '92 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 373–380, 1992.

[24] J. Archer. Web Page Design: Heuristic Evaluation vs. User Testing. `http://web.ics.purdue.edu/~archerj/Course%20Deliverables/COM%`

20221%20-%20Web%20Page%20Design%20%28Heuristic%20Evaluation%20vs%20User%20Testing%29.pdf, 2010. [Accessed: 09 april 2015].

[25] S. Lauesen. *User Interface Design: A Software Engineering Perspective.* Addison Wesley, 2004.

[26] C. Lewis and C. Wharton. Cognitive walkthroughs. In *Handbook of Human-Computer Interaction (Second Edition)*, pages 717–732. North Holland, 1997.

[27] J. Rubin and D. Chisnell. *Handbook of Usability Testing, Second Edition: How to Plan, Design, and Conduct Effective Tests.* Wiley Publishing, Inc., 2008.

[28] J. Scholtz. Usability evaluation. http://www.itl.nist.gov/iad/IADpapers/2004/Usability%20Evaluation_rev1.pdf. [Accessed: 08 april 2015].

[29] J. Nielsen. Thinking Aloud: The #1 Usability Tool. http://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/, 2012-01-16. [Accessed: 23 mars 2015].

[30] J. Nielsen. Usability inspection methods. *Conference Companion on Human Factors in Computing Systems (CHI '94)*, pages 413–414, 1994.

[31] C. Karat, R. Campbell, and T. Fiegel. Comparison of empirical testing and walkthrough methods in user interface evaluation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*, pages 397–404, 1992.

[32] R. Jeffries, J. R. Miller, C. Wharton, and K. M. Uyeda. User interface evaluation in the real world: A comparison of four techniques. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*, pages 119–124, 1991.

[33] W. Tan, D. Liu, and R. Bishu. Web evaluation: Heuristic evaluation vs. user testing. *International Journal of Industrial Ergonomics*, 39(4):621–627, 2009.

[34] R. Jeffries and H. Desurvire. Usability testing vs. heuristic evaluation: Was there a contest? *SIGCHI Bull.*, 24(4):39–41, 1992.

[35] R. Inostroza, C. Rusu, S. Roncagliolo, S. Jiménez, and V. Rusu. Usability heuristics for touchscreen-based mobile devices. *Information Technology: New Generations (ITNG), 2012 Ninth International Conference on*, pages 662 – 667, 2012.

[36] C. Rohrer. When to Use Which User-Experience Research Methods. `http://www.nngroup.com/articles/which-ux-research-methods/`, 2012-10-12. [Accessed: 9 februari 2015].

[37] P. A. Smith. Towards a practical measure of hypertext usability. *Interacting with Computers*, 8(4):365–381, 1996.

[38] M.Y. Ivory and M.A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, 33(04):470–516, 2001.

[39] P. Burzacca and F. Paternò. Remote usability evaluation of mobile web applications. *Human-Computer Interaction. Human-Centred Design Approaches, Methods, Tools, and Environments*, 8004:241–248, 2013.

[40] D. M. Hilbert and D. F. Redmiles. Extracting usability information from user interface events. *ACM Computer Surveys*, 32(4):384 – 421, 2000.

[41] Jakob Nielsen. How to Conduct a Heuristic Evaluation. `http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/`, 1995-01-01. [Accessed: 24 mars 2015].

[42] A. de Lima Salgado and A. P. Freire. Heuristic evaluation of mobile usability: A mapping study. *Human-Computer Interaction. Applications and Services*, 8512:178–188, 2014.

[43] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 249–256, 1990.

[44] R. Y. Gómez, D. C. Caballero, and J. Sevillano. Heuristic evaluation on mobile interfaces: A new checklist,. *The Scientific World Journal*, 2014.

A | Checklista heuristisk utvärdering

A.1 Visibility of system status

System status feedback

1. Is there some form of system feedback for every operator action?
2. If pop-up windows are used to display error messages, do they allow the user to see the field in error?
3. In multipage data entry screens, is each page labeled to show its relation to others?
4. Are high informative contents placed in high hierarchy areas?
5. Are all the items in a list on the same page? Are they sorted in an order that matches the needs of the task?
6. If the list contains only one item, is the user taken directly to that item?

   Location information

7. Is the logo meaningful, identifiable and sufficiently visible?
8. Is there any link to detailed information about the enterprise, web site, webmaster...?
9. Are there ways of contacting with the enterprise?

   Response times

10. Are response times appropriate for the users cognitive processing?
11. Are response times appropriate for the task?
12. Is latency sufficiently reduced?

13. Are splash screens too long avoided?

Selection/input of data

14. Is there visual feedback in menus or dialog boxes about which choices are selectable?
15. Is the current status of an icon clearly indicated?
16. Is there visual feedback when objects are selected or moved?
17. Are links recognizable? Is there any characterization according to the state (visited, active,...)?
18. Are low discoverable areas as touch buttons well identifiable?
19. When swiping gesture is possible, is a visible clue offered to users? Is swiping used with a unique meaning in the same screen?
20. Are expandable menus used sparingly? Do menu labels clearly indicate that they expand to a set of options?

A.2 Match between system and real world

Metaphors/mental models

1. Are metaphors properly used as visual clues?
2. Are icons concrete and familiar?
3. If shape is used as a visual cue, does it match cultural conventions?
4. Do the selected colours correspond to common expectations about colour codes?

Navigational Structure

5. If the application uses hierarchical structure, are depth and height balanced?
6. Is a navigation map or table of contents included in the application?
7. Is too much navigation avoided?

Menus

8. Are menu choices ordered in the most logical way, given the user, the item names, and the task variables?

9. Do menu choices fit logically into categories that have readily understood meanings?
10. Are menu titles parallel grammatically?
11. In navigation menus, are the number of items and terms by item controlled to avoid memory overload?

Simplicity

12. Do related and interdependent fields appear on the same screen?
13. Is the language used the same target users speak?
14. Is the language clear and concise?

Ouput of numeric information

15. Does the system automatically format phone numbers according to country-specific formatting rules?
16. Does the system automatically format credt card numbers according to specific formatting rules?

A.3 User Control and Freedom

Explorable interfaces

1. Can users move forward and backward between fields or dialog box options?
2. If the system has multipage data entry screens, can users move backward and forward among all the pages in the set?
3. Are exits clearly marked?
4. Is there any way to inform user about where they are and how to undo their navigation?
5. Is accidental activation avoided or foreseen (a back button is offered)?

Some level of personalization

6. Can users set their own system, session, file, and screen defaults?

Process Confirmation

7. When a user's task is complete, does the system wait for a signal from the user before processing?

8. Are users prompted to confirm commands that have drastic, destructive consequences?

Undo/cancellation

9. Can users easily reverse their actions?
10. Can users cancel out of operations in progress?

Menus Control

11. If the system has multiple menu levels, is there a mechanism that allows users to go back to previous menus?
12. Are menus broad (many items on a menu) rather than deep (many menu levels)?
13. If users can go back to a previous menu, can they change their earlier menu choice?

A.4 Consistency and Standards

Orientation

1. Is constraining orientation avoided? (Users tend to switch orientation when an impasse occurs and, if the app doesn't support them, their flow is going to be disrupted and they are going to wonder why it's not working)
2. Is navigation (horizontal and vertical) consistent across orientations? (Some applications use a different navigation direction in the two orientations; for instance, they use horizontal navigation in landscape and use vertical navigation in portrait).
3. Is content consistent across orientations?

Designing Consistency

4. Are attention-getting techniques used with care?
5. Is intensity maintained in two levels only?
6. Is the number of colour used constrained up to four? Are additional colours saved for occasional use only?
7. Are the colour far apart along the visible spectrum?

8. Are soft tones used for regular positive feedback and harsh for rare critical conditions?
9. Have industry or company standards been established for menu design, and are they applied consistently on all menu screens in the system?
10. Are there no more than twelve to twenty icon types?
11. Has a heavy use of all uppercase letters on a screen been avoided?
12. Is there a consistent icon design scheme and stylistic treatment across the system?

Menus

13. Are menu choice lists presented vertically?
14. Are menu titles either centred or left-justified?

Input fields

15. Are field labels consistent from one data entry screen to another?
16. Do field labels appear to the left of single fields and above list fields?
17. Are field labels and fields distinguished typographically?

Naming Convention Consistency

18. Is the structure of a data entry value consistent from screen to screen?
19. Are system objects named consistently across all prompts in the system?
20. Are user actions named consistently across all prompts in the system?

Menu/task consistency

21. Are menu choice names consistent, both within each menu and across the system, in grammatical style and terminology?
22. Does the structure of menu choice names match their corresponding menu titles?
23. Does the menu structure match the task structure?
24. When prompts imply a necessary action, are the words in the message consistent with that action?
25. Does the look & feel correspond with goals, characteristics, contents and services of the application?

A.5 Error Prevention

1. Are menu choices logical, distinctive, and mutually exclusive?
2. Are data inputs case-blind whenever possible?
3. Does the system warn users if they are about to make a potentially serious error?
4. Do data entry screens and dialog boxes indicate the number of character spaces available in a field?
5. Do fields in data entry screens and dialog boxes contain default values when appropriate?

   Fat-finger Syndrome

6. Are touchable areas sufficiently big? (Research has shown that the best target size for widgets is 1cmx1cm for touch devices)
7. Is crowding targets avoided? (When targets are placed too close to each other, users can easily hit the wrong one)
8. Although the visible part of the target may be small, is there some invisible target space that if a user hits thatspace, their tap will still count?
9. When several items are listed in columns, one on top of another, can users hit anywhere in the row to select the target corresponding to that row?
10. Is there sufficient space between buttons that perform different actions? (For example a change button and an erase button)

A.6 Recognition Rather Than Recall

Memory Load Reduction

1. Are high levels of concentration not required and remembering information doesn't take more than two to fifteen seconds?
2. Are all data a user needs on display at each step in a transaction sequence?
3. If users have to navigate between multiple screens, does the system use context labels, menu maps, and placemarkers as navigational aids?

4. After the user completes an action (or group of actions), does the feedback indicate that the next group of actions can be started?
5. Are optional data entry fields clearly marked?
6. Do data entry screens and dialog boxes indicate when fields are optional?
7. Do the task flow should start with actions that are essential to the main task? And can the users start the taskas soon as possible?
8. Are the controls that are related to a task grouped together and reflect the sequence of actions in the task?

General Visual Clues

9. Does the data display start in the upper-left corner of the screen?
10. Have prompts been formatted using white space, justification, and visual cues for easy scanning?
11. Do text areas have breathing spaceäround them?
12. Are there åhiteäreas between informational objects for visual relaxation?
13. Does the system provide visibility: that is, by looking, can the user tell the state of the system and the alternatives for action?
14. Are size, boldface, underlining, colour, shading, or typography used to show relative quantity or importance of different screen items?
15. Is colour used in conjunction with some other redundant cue?
16. Is there good colour and brightness contrast between image and background colours?
17. Have light, bright, saturated colours been used to emphasize data and have darker, duller, and desaturatedcolours been used to de-emphasize data?
18. Is the visual page space well used?

Input/Output data

19. On data entry screens and dialog boxes, are dependent fields displayed only when necessary?
20. Are field labels close to fields, but separated by at least one space?

Menus

21. Is the first word of each menu choice the most important?

22. Are inactive menu items grayed out or omitted?
23. Are there menu selection defaults?
24. Is there an obvious visual distinction made between "choose onemenu and "choose manymenus?

A.7 Aesthetic and Minimalist design

1. Is only (and all) information essential to decision making displayed on the screen?
2. Are field labels brief, familiar, and descriptive?
3. Are prompts expressed in the affirmative, and do they use the active voice?
4. Is layout clearly designed avoiding visual noise?
5. Are application icons recognizable enough to be found in the crowded list of applications?

   Multimedia Content

6. Does the use of images and multimedia content add value?
7. Are images well sized? Are they understandable? Is the resolution appropriate?
8. Are cyclical animations avoided?
9. Is flash content avoided?
10. Is the use of animated carousels avoided? And if they exists, can users control them?
11. Are image sizes smaller than the screen? (The entire image should be viewable with no scrolling)
12. Are moving animation avoided?
13. Is the whole screen surface used to place information efficiently (specially for popovers and modals)?

    Icons

14. Has excessive detail in icon design been avoided?
15. Is each individual icon a harmonious member of a family of icons?

16. Does each icon stand out from its background?
17. Are all icons in a set visually and conceptually distinct?

Menus

18. Is each lower-level menu choice associated with only one higher level menu?
19. Are menu titles brief, yet long enough to communicate?

Orientation

20. Desktop websites have a strong guideline to avoid horizontal scrolling. But for touch screens, horizontal swipes are often fine. Is this option taken into account?

Navigation

21. Is the application designed to avoid a large number of persistent navigation options across all screens?

A.8 Help users recognize, diagnose and recover from errors

1. When signalling an input error in a form, is the text box that needs to be changed specifically marked?
2. When an error message is presented, does it describe the issue in a precise way? Is the error message easy to understand?
3. Does the system suggest a constructive solution to the issue?

A.9 Help and Documentation

1. Do the instructions follow the sequence of user actions?
2. If menu choices are ambiguous, does the system provide additional explanatory information when an item is selected?
3. If menu items are ambiguous, does the system provide additional explanatory information when an item is selected?

4. Is the help function visible; for example, a key labeled HELP or a special menu?
5. Navigation: Is information easy to find?
6. Presentation: Is the visual layout well designed?
7. Conversation: Is the information accurate, complete, and understandable?
8. Is the information relevant? It should be relevant in the following aspects: Goal-oriented (What can I do with this program?), Descriptive (What is this thing for?), Procedural (How do I do this task?), Interpretive (Why did that happen?) and Navigational (Where am I?).
9. Is there context-sensitive help?
10. Can the user change the level of detail available?
11. Is it easy to access and return from the help system?
12. Can users resume work where they left off after accessing help?
13. Is the design focused on one single feature at a time? (Only those instructions that are necessary for the user to get started should be presented at a time).

A.10 Skills

1. If the system supports both novice and expert users, are multiple levels of error message detail available?
2. If the system supports both novice and expert users, are multiple levels of detail available?
3. Are users the initiators of actions rather than the responders?

A.11 Pleasurable and Respectful Interaction

1. Is the users' work protected? For example, for data entry screens with many fields or in which source documents may be incomplete, can users save a partially filled screen?

   Input Data

2. Users dislike typing. Is information computed for the users? For instance, ask only for the zip code andcalculate state and town; possibly offer a list of towns if there are more under the same zip code.
3. Can users save history and select previously typed info?
4. Does default information make sense to the user?
5. If the app does not store any information that is sensitive (e.g. credit card), is the user kept logged in (with log out clearly presented)?
6. Is the number of submissions (and clicks) minimized for the user going through in order to input informationon the site?
7. When logging in must be done, are graphical passwords used at least some of the time, to get around typing?
8. Is registration not mandatory? Is skipping registration the default option?
9. When logging in must be done, is there an option that allows the user to see the password clearly?

   Banking and Transactions

10. Whenever users conduct transactions on the phone, can they save confirmation numbers for that transaction by emailing themselves?

A.12 Privacy

1. Are protected areas completely inaccessible?
2. Are protected or confidential areas only accessible with certain passwords?
3. Is there information about how personal data is protected and about contents copyright?
4. For multiuser devices: Is permanently signing in on an application avoided?
5. If the app does store credit card info, can users decide if they want to remain logged in? If the user opts to be kept logged in, he/she should get a message informing of the possible risks.