Reform av bedrägerifilter

LUNDS UNIVERSITET
Lunds Tekniska Högskola

LTH Ingenjörshögskolan vid Campus Helsingborg
Institutionen för datavetenskap

Examensarbete:
Niclas Fredriksson
Jonas Nilsson

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

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

Tryckt i Sverige
E-husets tryckeri
Lunds universitet
Lund 2013

Sammanfattning

Detta examensarbete berör Resurs Bank och deras bedrägerifilter inom e-handel. Ett försök till bedrägeri innebär att på något sätt utföra en form av identitetsstöld för att på så sätt undgå att behöva betala för produkten man beställt. Bedrägeriförsök sker dagligen hos Resurs Bank och kan innebära höga onödiga utgifter för banken.

Bankens nuvarande filter är säkert, men kräver mycket manuella kontroller som i sin tur innebär stora utgifter i form av lön till personalen som gör kontrollerna. Resurs Bank har hittills tjänat på att kontrollera så hög andel av beställningarna men nu vill banken effektivisera hanteringen. Det ska göras genom att minska antalet manuella kontroller men samtidigt behålla den höga säkerheten som finns.

I detta examensarbete har det genomförts en analys av bedrägerifiltret där två nämnare har hittats som är vanliga vid bedrägeriförsök. Två nya metoder har därför implementerats i Resurs Banks bedrägerisystem som håller kontrollerar respektive nämnare vid ett köp. Metoderna ger Resurs Bank möjlighet att hämta mer information från deras olika system än de tidigare kunde. Med hjälp av dessa två metoder har hela bedrägerifiltret kunnat bli effektivare då övriga metoder kunnat konfigureras till de bättre.

> *”/---/ Dessa två nya metoder kommer att minska våra manuella kontroller väsentligt vilket betyder att kostnaden för att förhindra bedrägerier kommer minska”* - Johan Ljungström 2013-05-22

Nyckelord: bedrägeri, filter, testning, Resurs Bank, e-handel, identitetsstöld

Abstract

This bachelor thesis concern Resurs Bank and their current fraud filter within ecommerce. An attempt to commit fraud means that in someway perform a form of identity theft and therefore avoid paying for the product that has been commissioned. Attempted fraud take place daily at Resurs Bank and can involve large unnecessary expenditures.

Resurs Bank's current fraud filter is safe but demands a lot of manual controls, which means large expenditures in the form of salary for the personnel that perform the manual controls. Resurs Bank have so far benefited from controlling a high amount of orders but wishes now to improve the eyfficiency of the filter. The solution is to reduce the amount of manual controls but at the same time keep the high security that exists.

It has conducted an analysis of the fraud filter where two denominators have been found that are common in attempt of frauds. Two new methods have been implemented in Resurs Bank's fraud system that checks each denominator when an order is made. These methods give Resurs Bank the opportunity to fetch more information from their different systems than before. These two methods have made the fraud filter more efficient because of the other methods in the filter have been configured in a better way.

> *"/---/ These two methods will reduce our manual controls significantly which means that the cost to prevent frauds will decrease"* - Johan Ljungström 2013-05-22

Keywords: fraud, filter, Resurs Bank, ecommerce, identity theft

Förord

Vi vill tacka Resurs Bank för möjligheten att göra vårt examensarbete hos dem. Examensarbetet har varit otroligt spännande och lärorikt. Även våra handledare Markus Kruse och Pär Nilsson ska tackas, när vi har stött på problem har vi alltid haft er nära till hands för rådfrågning och hjälp, utan er hade detta examensarbetet inte blivit avslutat. Ytterligare personer vi vill tacka är Eric Cedergren, Damir Kafedzic, Johan Ljungström och Markus Torstensson som alla bedragit med expertis inom sina områden.

På Campus vill vi tacka vår examinator Christin Lindholm för hjälpen kring projektrapporten.

Innehållsförteckning

1 Inledning

Resurs Bank AB samt Jonas Nilsson och Niclas Fredriksson från Lunds Tekniska Högskola har i detta examensarbete samarbetat för att förbättra Resurs Banks bedrägerifilter, då detta börjar bli föråldrat och underpresterande. Banken har från 2008 ökat antal butiker från 15 000 till 25 000 butiker. Eftersom Resurs Bank ökat sin omsättning vill de även förbättra och förbereda sig för fler bedrägeriförsök t.ex. genom att en person utger sig vara en annan, det vill säga identitetsstöld. Banken vill både se till att minska antalet bedrägerier samtidigt som de vill ha ner antalet onödiga frysningar. Onödiga frysningar innebär att Resurs Bank valt att följa upp ärendet och att det har uppenbarat sig att det inte var ett bedrägeriförsök. Resurs Banks målsättning är även att effektivisera arbetet kring sin bedrägerihantering. Filtret består av två olika delar, en del som kontrollerar om personen är kreditvärdig och den andra delen kontrollerar om det är ett bedrägeriförsök. Det är det sistnämnda som ska uppdateras.

När filtret skapades i slutet av 2011 var målet att inte vara en bank där det skulle vara enkelt att lyckas med ett bedrägeri. För att inte få detta rykte lade banken extra stora resurser på att kontrollera många köp. Vid examensarbetets start användes fortfarande denna metod.

Eftersom arbetet handlar om mycket känslig information så som kundinformation, bedrägerifiltrets struktur m.m. så kommer vissa delar inte dokumenteras i minsta detalj, bl.a. implementation och resultat.

1.1 Bakgrund

Resurs Bank har sin grund i Resurs Finans som i sin tur har sitt ursprung i Resurs Radio & TV. Resurs Finans bildades på mitten av 1980-talet av Thomas Paulsson med ett koncept de kallade räntefritt. Bankens omfattning har fram till 2013 ökat till ungefär 200 anställda i sitt huvudkontor i Helsingborg. Banken har inga bankkontor utan all support för kunderna sker genom telefon till huvudkontoret. 2 miljoner kunder i form av större företag inom detaljhandeln och privatpersoner väljer att låna eller spara i banken. Resurs Bank har 25 000 butiker som erbjuder sina kunder Resurs Banks delbetalning. Resurs Bank har några värderingar som är viktiga för banken, ”kunden är kung”, ”snabbhet för precision”, ”be hellre om förlåtelse än tillåtelse”, ”leverera kvalitet”, ”värdera våra resurser” och ”allt börjar med oss själva”. Dessa är alltid närvarande för de anställda i banken. [1.1]

1.2 Problemformulering

Syftet med examensarbetet är att undersöka och analysera Resurs Banks bedrägerifilter och undersöka om det finns några kryphål. Dessa kryphål ska försöka elimineras för att göra filtret säkrare genom att testa filtret på de bedrägerier som gled förbi filtret första gången. Den historik som innefattar försök till bedrägerier och faktiska bedrägerier gör det möjligt att hitta samband och därigenom konstruera nya metoder för filtret.

Som nämndes i inledningen kan ett exempel på bedrägeri vara att bedragaren lägger en beställning i ett falskt namn för att sedan kunna bevaka personens brevlåda och hämta ut paketet med hjälp av sitt falska id. Om inte filtret triggas av den inputen personen har angivit har personen som lagt beställningen kommit igenom filtret och då har personen hittat ett så kallat kryphål, personen i fråga har lyckats genomföra ett bedrägeri. Detta skulle exempelvis kunna lösas genom att kontrollera att namn och telefonnummer/e-mail stämmer överens med varandra. Här kan man också föra statistik kring telefonnummer och e-mail som är kända sedan innan på grund av bedrägeri eller bedrägeriförsök, en svartlista, och därför redan vid beställning hissa en varningsflagga.

Detta examensarbete kommer mer specifikt svara på följande huvudfråga:

a) Hur kan man förbättra filtret?

Det finns även följdfrågor som ska besvaras:

b) Vad för kryphål finns det?
c) Vad kan göras för att eliminera kryphålen?
d) Går det att öka säkerheten utan att öka frysningarna?
e) Går det att minska antalet frysningar utan att minska säkerheten?

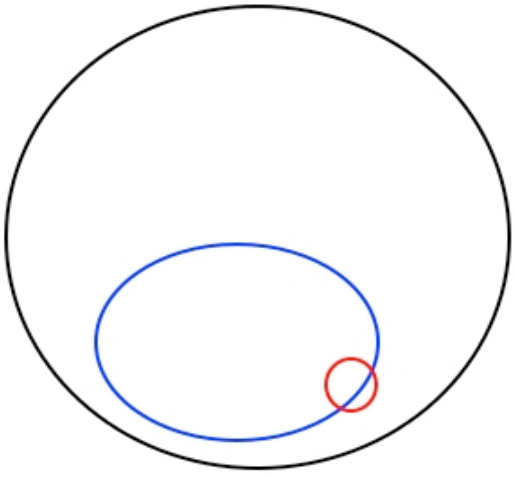


Fig.1.2.1 - Illustration av förhållandet mellan antal köp, antal frysningar och antal bedrägerier. OBS, endast approximativ

Efter några veckors arbete som mestadels berörde nya metoder visade det sig att effektiviseringen av filtret hade blivit underprioriterad och skulle därför få högre prioritet. Med detta menades att det gör alldeles för många frysningar på oskyldiga personer och det är väldigt tidskrävande att

kontrollera så många personer. Frågeställning e) blev därför den viktigaste frågeställningen. Här tog arbetet en liten men ny vändning.

I fig.1.2.1 innehåller den yttersta ringen alla köp som görs och den mellersta består av alla frysningar som görs. Den minsta cirkeln är alla bedrägerier/bedrägeriförsök. Figuren är en approximativ bild och ska inte begrundas allt för djupt. Tanken är att efter de nya metoderna lagts till och de gamla delarna optimerats så ska den mellersta ringen bli mindre och den minsta ringen inte ska bli större utanför den mellersta ringen.

Ett problem som uppstår vid implementation av nya metoder är att man inte kan hämta den nödvändiga informationen direkt från ett annat systems databas utan endast till den databas som är dedikerad till just det systemet. Fig.1.2.2 visar detta förhållandet. Detta av den orsaken att Resurs Bank vill ha sina applikationer så självständiga som möjligt annars leder det till instabila och opålitliga applikationer.

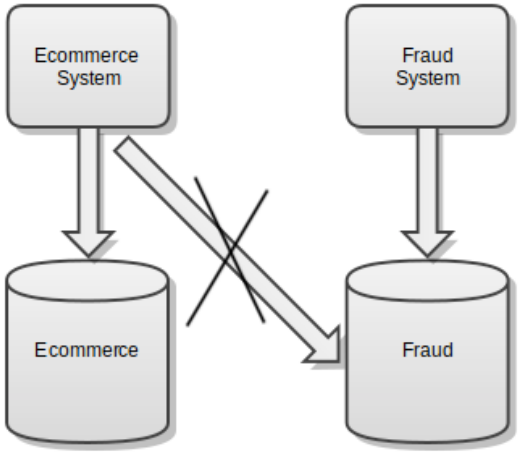


Fig.1.2.2 – Illustration av de olika beroendena mellan systemen och databaserna.

Fig.1.2.3 visar hur man löser instabilitetsproblemet med hjälp av en web service. På förfrågan från tillexempel fraud systemet hämtar web servicen den data som efterfrågas från ecommerce systemet och skickar tillbaka till fraud.

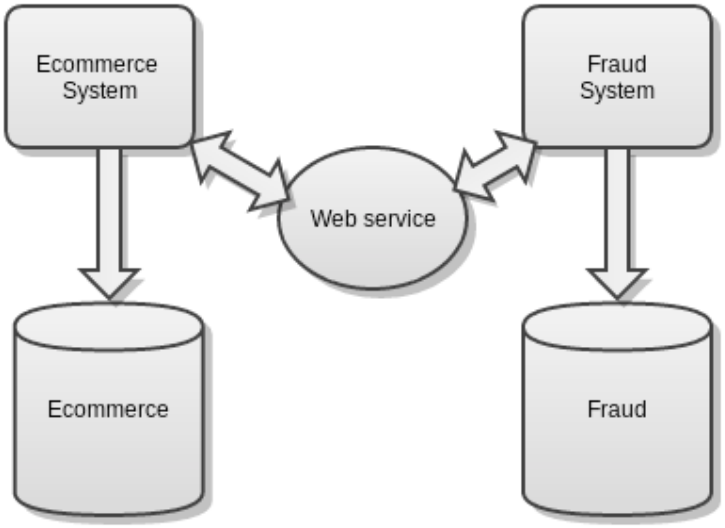


Fig.1.2.3 – Illustration av lösning på instabilitetsproblemet.

1.3 Avgränsningar

Eftersom examensarbetet innefattar 22,5 högskolepoäng så ligger arbetet inom ramen av 15 veckor. Analysen kommer därför begränsas till möten och föra statistik. Filtret kommer att behöva uppdateras även efter examensarbetets slut, detta på grund av att det ständigt kommer ny teknik och ändringar som gör det möjligt att lura filtret på nya sätt, och därför komma runt även de nya metoderna. Sett ur denna synvinkel kan man säga att det inte går att lösa problemet helt, utan bara delvis.

1.4 Förväntat resultat

Analysen ska ge underlag för vilka metoder som ska implementeras. Metoderna ska till stora delar lösa problemet på Resurs Bank och som resultat av det bör de lyckade bedrägerierna minska. Detta är svaret på hur huvudfrågeställningen ska lösas. Det är de nya metoderna som kommer att se till att kryphålen i de flesta fall elimineras.

Skulle det visa sig att det inte går att implementera några metoder som uppkommit av analysen, ska lösningsförslag och information om kryphålen ges till Resurs Bank i ett slutmöte.

Önskat resultat av filtret kommer innebära att fler bedrägeriförsök fastnar och att bedrägerigruppen inte behöver motringa lika många personer som de gör idag, vilket betyder ringa upp personer som fastnat i filtret för att avgöra om det är ett bedrägeriförsök eller ej. Resurs Bank motringer ungefär 250 personer i veckan och utöver dessa 250 gör dem andra manuella kontroller av personerna i köpen.

Efter vändningen som nämns i problemformulering, där effektiviseringen sattes till högsta prioritet istället för säkerheten, blev det förväntade resultatet lite annorlunda, men fortfarande likt det ovanstående. Skillnaden är att fokusen har hamnat på att minska antalet frysningar utan att minska säkerheten.

1.5 Befintliga filtret

Om en person som lagt en beställning på kredit hos NetOnNet kommer ansökningen gå igenom filtrets två delar, först kreditupplysningen och sedan bedrägerikontrollen. Om ansökningen passerar kreditupplysningen går den till bedrägerikontrollen. Den innehåller olika metoder, dessa kontrollerar grålistan, olika begränsningar och uppgifter som anges i ansökan m.m. Om ansökan triggar för många av metoderna kommer den att frysas och bedrägerigruppen tar vid. Metoderna har olika parametrar för att triggas igång, de har även olika allvarlighetsgrad, low, medium och high. Det behövs således fler medium än high för att metoden ska flagga medan low anses vara godkänt. Se kap.7.1 för ett fiktivt exempel.

1.6 Intressenter

För att göra examensarbetet så bra som möjligt för alla inkluderande parter behöver intressenterna i projektet tillfredsställas. Det är därför viktigt att kartlägga intressenterna.

Damir Kafedzic är gruppchef för bedrägerigruppen. Hans grupp är de som jobbar närmast bedrägerilådan, därför har Damir störst kunskap gällande bedrägerier. Det innebär att han och hans grupp blir den viktigaste intressenten, då de jobbar mot bedrägerilådan hela tiden. Johan Ljungström är kredit- och riskchef, han har övergripande ansvar bl.a. för bedrägerilådan. Han arbetar inte lika intimt med bedrägerilådan som Damir, men har alltjämt en viktig roll eftersom han bestämmer.

Markus Kruse och Pär Nilsson är handledarna för projektet och jobbar i javagruppen. Där finns även Markus Torstensson som är systemutvecklare. Han, tillsammans med Erik Cedergren, som är javakonsult, har utvecklat det befintliga filtret med Anders Hovén, även han systemutvecklare**.** Det är först och främst handledarna som hjälper till med förståelsen av kod medan Torstensson och Cedergren hjälper till med implementation av nya metoder. Själva analysen och frågor angående bedrägerier som utförts går till Damir.

Henrik Eklund är CIO på Resurs Bank, han ansvarar för allt som sker på IT-avdelningen och blir därmed den högst uppsatta intressenten.

De hittills nämnda intressenterna är alla delar av Resurs Bank. Det finns en viktig intressent utanför Resurs Bank och det är NetOnNet. NetOnNet erbjuder tillsammans med Resurs Bank kunden möjlighet till delbetalning, en kredit upp till en viss summa. Kunden själv väljer uppläggningstid och delbetalar varje månad tillbaka till Resurs Bank.

1.7 Teknisk bakgrund

För att inhämta den önskade informationen som behövs vid analysen, så som testen, har olika program använts för att förenkla denna fas. Vid utveckling har en utvecklingsmiljö använts tillsammans med andra nödvändiga program. Därför kommer detta kapitel lista de program som använts.

Fig.1.7 visar hur de olika programmen är beroende av ett annat program på olika vis, de små pilarna beskriver detta. Beroendena kan vara att det behövs information från databasen i systemet MultiUpplys för att en metod som implementeras i IntelliJ ska kunna fungera som avsett. Följande underkapitel kommer förklara mer ingående hur programmen är beroende av varandra.

Samtliga program är program som Resurs Bank använder sig av, det är även dessa som lämpar sig bäst för uppgifterna. Vid val av IntelliJ fanns även Eclipse som alternativ till utvecklingsverktyg, men på stark rekommendation från handledarna och viljan att testa nytt föll valet på IntelliJ.

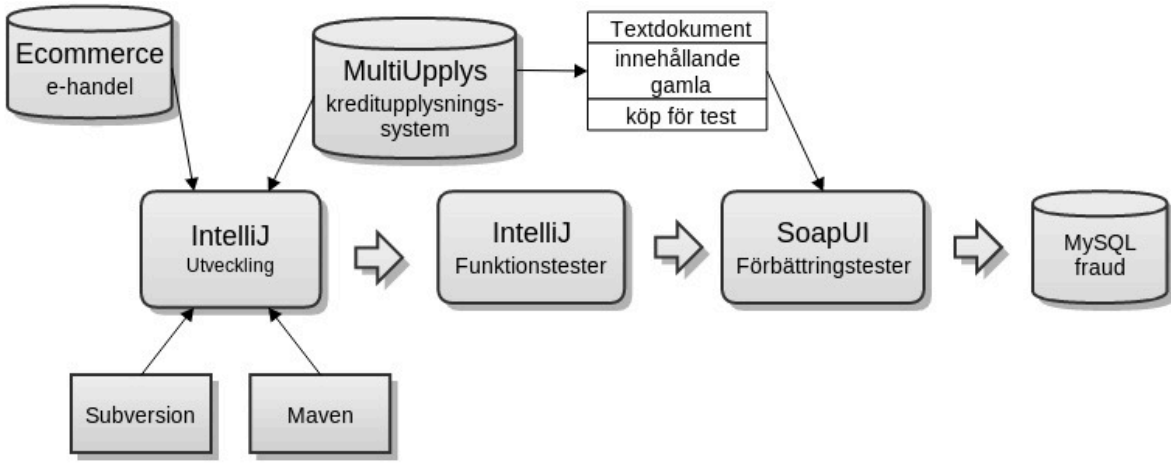


Fig.1.7 – Illustration av hur programmen nyttjas

1.7.1 IntelliJ IDEA
IntelliJ IDEA är en utvecklingsmiljö som används för programmering. IDE betyder att den har en textredigerare, debugger och kompilator. A står för att den finns som en Apache licens, vilket innebär lättare integration med Apache program, så som Apache Maven eller Apache Subversion. IntelliJ har framförallt använts vid javaprogrammering av de nya metoderna, men även för att skriva funktionalitetstester till metoderna och köra testen. Det behövs information till metoderna, den erhålls från Ecommerce och MultiUpplys, som vardera innehåller olika typer av information. Med hjälp av dessa har metoderna underlag att fullgöra sin funktion.
[1.7.1]

1.7.2 SoapUI
För att testa gamla köp på nya filtret (se kap 6.1) användes testverktyget SoapUI som är ett plattformsoberoende open-source program för automatisk testning. Istället för att köra ett test i taget manuellt, det vill säga att man skriver in ett referensnummer för ett köp och sedan kör testet, kan man i SoapUI köra t.ex. 5000 testfall samtidigt och automatiskt, och resultaten kan läggas in i en MySQL databas. Det görs genom att referensnummer hämtas från MultiUpplys och sparas i ett textdokument, dokumentet läses i sin tur av SoapUI.
[1.7.2]

1.7.3 MySQL Workbench
MySQL Workbench är ett program för att hantera databaser. Programmet använder sig av frågespråket SQL, som innebär att man ställer frågor i form av queries till databasen och får tillbaka informationen man sökt efter. Det var detta program som användes vid hämtning av information från SoapUI testerna. Informationen kan sedan granskas i MySQL Workbench. MySQL Workbench är open-source och används sedan tidigare av Resurs Bank.
[1.7.3]

1.7.4 MongoDB
MongoDB är ännu ett open-source program som Resurs Bank använder sig av. Det är en databas som nyttjar de fyra grundoperationerna inom databaser, CRUD. MongoDB använder sig inte utav SQL för att hämta ut data och sparar inte informationen i tabeller som exempelvis MySQL gör. MongoDB sparar istället undan relaterad data tillsammans i dokument. Detta gör denna databas snabbare och lättare att använda eftersom effektivare queries kan skrivas. Anledningen till att MongoDB också används är att Resurs Bank tidigare

använt sig av MongoDB för att spara undan just den information som behövs för att kunna använda gamla köp och för att lagra data från systemet MultiUpplys.
[1.7.4.1],[1.7.4.2]

1.7.5 Apache Maven

Apache Maven är ett program som underlättar kompilering, vilket används av IntelliJ IDEA. Apache Maven har använts till att underlätta förståelsen för hela kompileringsfasen på så kort tid som möjligt, bl.a. genom att man inte behöver förstå alla underliggande detaljer när man bygger ett projekt. En konfigurationsmodell som kallas Project Object Model (POM) används när man bygger ett projekt. En POM-fil är en XML-baserad fil som beskriver hur projektet ska byggas och eftersom alla Maven projekt byggs på samma sätt räcker det med att bekanta sig med en POM-fil för att kunna förstå hur andra projekt fungerar, vilket också sparar tid.
[1.7.5]

1.7.6 Apache Subversion

För att underlätta arbetet på Resurs Bank använder dem sig av versionshanteringsprogrammet Apache Subversion. Subversion har ändringsloggar som gör det möjligt för alla personer att se vad som ändrats i en fil sedan en tidigare version. Ett exempel på användning var när Markus Kruse gjorde en ändring i koden och gjorde en versionsuppdatering, då importerades enkelt den nya versionen med Subversions hjälp.
[1.7.6]

1.7.7 LibreOffice Calc

LibreOffice Calc är ett program som fungerar nästan som Microsoft Excel, fast är gratis att ladda ner. Från MySQL Workbench kan man exportera data som kan öppnas med LibreOffice Calc. Programmet erbjuder behändiga funktioner, bl.a. formler för sammanslagning av olika kolumner och för att räkna antalet rader med ett specifikt kriterium. Dessa två formler var de som användes flitigast. En annan förträfflig funktion är att kunna skapa diagram genom att markera området man vill beskriva med ett diagram.
[1.7.7]

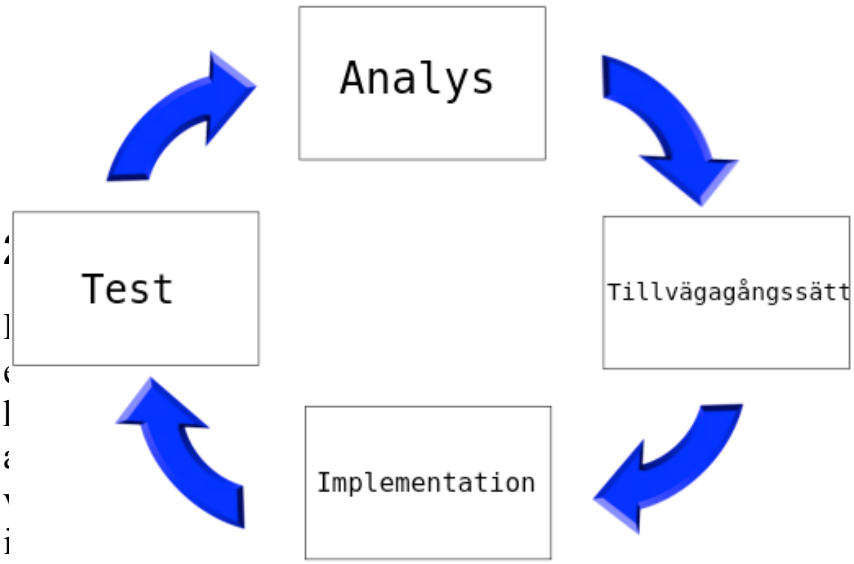


Fig2.1 - Illustration av arbetsflödet

idplan att arbeta
ap.5 och kap.6)
att behöva
etod som
odellen inte sker
å att arbeta iterativt,
att efter varje test återga till analysen. Någon känd processmodell valdes inte heller, detta för att kunna vara lättrörliga och anpassningsbara efter Resurs Banks önskemål. Det har inträffat dagliga morgonmöten där dagens agenda har gåtts igenom vilket gör att arbetsmodellen efterliknar Scrum[2] till en liten del. Utöver det finns inga större likheter med Scrum eller andra arbetssätt.

2.1 Arbetsflöde

Arbetet var, som syns i fig.2.1, uppdelat i 4 olika delar. Första delen bestod av att analysera bedrägerierna som skett och bedrägerierna som fastnat i filtret för att hitta bristerna. Efter det undersöktes det om bristerna går att reducera, och i så fall hur, för att senare implementeras och sist testas.

Detta inträffade iterativt eftersom efter testningen var det möjligt att se vad som gått igenom filtret. På grund av dessa arbetsformer valdes arbetsplatsen

till huvudkontoret i Helsingborg, endast där fanns tillgång till databaserna med historiken.

Under analysen var det tänkt att få en förståelse för problemen som finns med filtret medan åtgärder för dessa problem skulle uppkomma under tillvägagångssättsdelen.

Implementation gjordes i ett program som kallas IntelliJ IDEA, som är en utvecklingsmiljö för Java.

I testdelen finns det två olika test, funktionalitetstest och verifieringstest. Funktionalitetstesten utfördes på koden som skrivits och kontrollerade att metoden var rätt implementerad och fungerar som avsett. Verifieringstesten gjordes efter funktionalitetstesten och kontrollerade bara så att den nya metoden gjort någon positiv skillnad för filtret.

Innan den iterativa delen påbörjades första gången ägde en genomgång med upphovsmannen från Resurs Bank rum angående det nuvarande filtret. Även möten med personer som arbetar med och utanför filtret inträffade för att få en grund att stå på. De som använder sig av filtret är bedrägerigruppen. Tanken var att uppdatera det befintliga filtret istället för att skapa ett helt nytt, detta skapade mer tid åt de andra delarna.

Under den iterativa delen har kontakten med bedrägerigruppen fortlöpt då de besitter den djupare kunskapen om bedrägerierna som skett. Projektrapporten har frekvent uppdaterats för att minska mängden pappersarbete i slutet. En logg har förts och uppdaterats dagligen, den innehåller vad som gjorts på vilka dagar.

För att alltid kunna nå arbetet har filerna lagts upp på Google Drive, där det även fungerar som en kopia ifall arbetet försvinner på datorerna.

g.2.2 – Illustration av tidplanen

2.2 Tidplan

I fig.2.2 visas tidplanen som skrevs innan examensarbetet på Resurs Bank påbörjades. Eftersom den skrevs redan innan arbetet hade börjat så var det svårt att uppskatta hur lång tid varje del skulle ta och vad som krävdes för att varje del skulle avklaras. Tidplanen visar inte heller på någon iterativ arbetsprocess, detta för att ge en form av riktlinje som passar in på avgränsningarna i arbetet (kap.1.3), men som visas i fig.2.1 har arbetet skett iterativt, exempelvis har analysen hela tiden kommit tillbaka fram tills slutet.

Vecka:	Datum:	Att utföra:
1	2013-01-28	Intervjua/Få en lektion av nuvarande filter, läsa in sig och förstå systemet runt omkring och filter
2	2013-02-04	Fortsätta läsa in sig, även börja läsa in sig på programmeringen, annorlunda standard och så vidare..
3	2013-02-11	Börja analysera nuvarande filter
4	2013-02-18	Analysera
5	2013-02-25	Analysera och påbörja arbetet av ett bättre filter
6	2013-03-04	Fortsätta förbättra
7	2013-03-11	Tentamensvecka - Säkerhet 2013-03-15*
8	2013-03-18	Fortsätta förbättra
9	2013-03-25	Fortsätta förbättra
10	2013-04-01	Tentamensvecka - Realtidssystem 2013-04-08*
11	2013-04-08	Fortsätta förbättra
12	2013-04-15	Närmar sig slutet på arbetet med filtret
13	2013-04-22	Arbetet med filter klart efter godkännande av Resurs Bank
14	2013-04-29	Färdigställa dokumentatioin och förbereda redovisnig
15	2013-05-06	Färdigställa dokumentatioin och förbereda redovisnig

Tidplanen sträcker sig endast fram till 6:e maj och det visade sig inte vara tillräckligt, detta berodde på tre olika faktorer. Den främsta faktorn var att en web service behövde implementeras vilket inte var planerat från början. Andra faktorn var att verifieringstesterna (se kap.6) tog längre tid än tänkt från början. Sista anledningen var tentamensveckorna som även de tog längre tid än planerat.

3 Analys

I detta kapitel kommer det förklaras hur analysen som gett upphov till nya metoder gått till. Med hjälp av möten har en bild av hur filtret ser ut byggts upp och även de referensnummer som behövts har tilldelats.

3.1 Möten

För att få en förståelse om bedrägerifiltret har en del möten ägt rum. Johan Ljungström presenterade sin syn på det gamla filtret och då även förslag på förbättringar inför det nya filtret. Dessa förslag kunde senare bekräftas genom statistik. För att få en bild av hur arbetet runt omkring utförs träffade vi bedrägerigruppen och Damir Kafedzic. Detta gav en grund inför analysering av gamla bedrägerier, bl.a. så fryses väldigt många köp till ingen nytta då bedrägerigruppen efter manuell kontroll kan slå fast att det inte är något bedrägeriförsök.

3.2 Statistik

Bedrägerigruppen samlar ihop alla bedrägerifall som sker och sparar undan personnumren som används vid dessa fall, dessa används bl.a. för att kunna göra en grålista. De sparar även referensnummer i ett textdokument och det är dessa som används som underlag för analysen. För att kunna implementera nya och användbara metoder fanns det en grundtanke, vad har alla bedrägerier gemensamt, finns det några samband?

För att undersöka om det fanns gemensamma nämnare fanns det tillgång till MultiUpplys, som är ett system byggt av Resurs Bank. Systemet har tillgång till den information som kunden fyller i vid ett köp genom att ange ett referensnummer från till exempel listan som bedrägerigruppen gjorde. MultiUpplys gör även kreditupplysningen, bedrägerikontrollen samt presenterar resultatet av det. Det var med hjälp av MultiUpplys som informationen kring alla gamla köp kunde hämtas.

På de inledande mötena med Johan Ljungström och Damir Kafedzic tillhandahölls textdokumentet med referensnummer och personnummer samtidigt som de tipsade om olika nämnare att titta närmare på gällande de olika bedrägerier och bedrägeriförsöken som skett. Med hjälp av dessa tips var det enklare att hitta rätt information i MultiUpplys och börja föra statistik samt upptäcka svagheter kring filtret. Statistiken fördes genom att skriva upp faktorer för varje bedrägeri och bedrägeriförsök. De faktorer som förekom mest frekvent togs sedan upp med Johan och Damir där det kom fram om det skulle ske någon åtgärd.

3.3 Källkritik

Informationen med gamla köp (se kap.3.2) som hämtades, med hjälp av de referensnummer som tillhandahölls från kunddatabasen är trovärdig. Detta är information som kunden själv fyller i vid köp och information om köpet som till exempel vilken produkt det är och när köpet gjordes. Det får därför anses vara trovärdigt och inget som kan styras över. Vad man kan vara misstänksam mot är referensnumren, då Resurs Bank inte har någon standard vid hantering av referensnummer som blivit bekräftade bedrägeri eller bedrägeriförsök. På de olika mötena har olika personer presenterat deras syn på vad som anses vara bedrägeri, och här har det varit en delad syn. Beroende på vem som hanterade numren sparades antingen bekräftade bedrägerier undan, eller också försöken till bedrägeri. I detta examensarbete anses det vara samma sak eftersom båda ska hanteras på samma sätt i bedrägerifiltret, men det har gett anledning till att vara extra misstänksam mot annan information som tillhandahållits av Resurs Bank efter denna händelse.

Gällande den tekniska bakgrunden har information i första hand hämtats från respektive programs hemsida. På detta sätt kan man försäkra sig om att informationen kring programmens olika funktioner är trovärdig. Vad som är svårare är att bedöma vilket program som är bäst lämpad för just den uppgiften det använts till. Programmens hemsidor är skrivna på ett säljande sätt vilket gör det svårt att veta vad programmet passar bäst till, men då har handledarna funnits nära till hands för att ge sina åsikter angående programmen. Att Resurs Bank använder en del av programmen sedan tidigare har lett till att så många val av program inte behövt göras.

När metoderna testades för verifiering gjordes ett test med originalfiltret för att få en referens. Resultatet av detta verifieringstest blev som förväntat då det blev samma resultat som den verkliga händelsen, dvs. en beställning som frysts när den lades blev även frusen i verifieringstestet. Slutsatsen av detta blir att man kan lita på att resultaten från övriga verifieringstester då dessa har testats på samma sätt.

4 Implementation

Efter analysdelen påbörjades implementationen av metoderna. Som tidigare nämnts användes programmet IntelliJ IDEA och programspråket var java.

Det skedde en implementation av två nya metoder för att kontrollera två nya faktorer, det gjorde att filtret kunde triggas på ännu fler faktorer. När implementation av metoderna var klara lades de till i det befintliga filtret. Det befintliga filtrets metoder konfigurerades genom att ändra graden för när de skulle triggas genom att antingen sänka eller höja kravet för att systemet ska varna. (Se vidare kap.7.1 för hantering av konfigurationer)

För att metoden skulle fungera på ett korrekt sätt behövdes information från e-handelssystemet. Informationen innefattar beskrivning av produkterna som beställts och en typisk beskrivning kan vara ”Sony Playstation 3 250 gb”. Att låta metoden hämta den informationen själv skapar beroenden som gör systemet instabilt.

Nyckeln till att lösa problemet med instabilitet var att skapa en web service som hanterar inhämtningen av informationen istället för att låta metoden göra det själv. Implementationen är gjord så att metoden anropar en web service som i sin tur anropar en metod i e-handelssystemet (Ecommerce), denna metod återfår informationen från databasen. Informationen kan sedan användas i vårt system där våra metoder är implementerade.

En web service beskrivs i XML-format, och en standard i XML som Resurs Bank använder sig av är WSDL, detta för att deras testprogram SoapUI har en omfattande support för WSDL.
[4]

5 Testning

För att veta om de nya metoderna fullgör sin funktion krävs det att de avverkar en följd av test, kapitel 5 omfattar detta. Dessa testar stabiliteten och funktionaliteten i koden som skrivits.

5.1 Funktionalitetstester

Funktionalitetstester är tester som skrivs efter att en metod har implementerats. Dessa kontrollerar funktionaliteten och hanterar olika typer av input. Testen validerar metodens funktion och därför måste en metod passera alla tester utan fel. Om ett fel uppstår i något av testerna analyseras felet och en lämplig åtgärd implementeras. Antalet funktionalitetstester som metoden måste passera berodde på hur metoden implementerats och vad dennes avsikt är. Testen eliciterades och kördes mot metoden en i taget. Slutligen kördes alla testerna mot metoden för ett slutgiltigt funktionalitetstest. Passerades inte detta gjordes det ändringar tills det klarade hela testet.

Metoden ska även kunna hantera felaktig input. Denna input kommer från den person (en person från bedrägerigruppen) som vill ändra i filtret som är i bruk. Om vi exempelvis vill vi blockera alla som anger mobilnummer som börjar med 070 ska ett nummer med 10 siffror kontrolleras i bedrägerilådan medan ett nummer med 5 siffror eller bokstäver ska bli fel, då kastas ett exception. Likaså ska det bli fel om filtret förväntar sig siffror som input men får bokstäver.

6 Verifiering

Detta kapitel behandlar området kring verifiering av nya metoder, med andra ord om det nya filtret med nya konfigurationer fungerar som tänkt och om det ger någon slags förbättring.

6.1 Bidrar det nya filtret med en förbättring

För att avgöra om de nya metoderna samt de nya konfigurationerna av de redan existerande metoderna har bidragit till en förbättring (problemformulering d och e) går filtret igenom ett test. Detta test är inte att blanda ihop med kap.5.1 Funktionalitetstester som bara testar funktionaliteten hos den enskilda metoden. Här körs tester med gamla köp fast på det nya filtret. För att få ut en pålitlig statistik att utgå ifrån testades så många fall som möjligt mot det nya filtret. På grund av att gammal information måste raderas efter 90 dagar begränsades antalet testfall till 7177. Den 7177 långa listan med referensnummer består av köp som inte är bedrägerier/bedrägeriförsök. Förutom denna lista skapades ännu en lista inom samma tidsspann, även detta på grund av ovanstående problem. Den listan består av alla bekräftade bedrägerier samt alla bedrägeriförsök.

För att testet i SoapUI skulle kunna hantera referensnumren skrevs ett program. Programmet skrevs i Groovy som är ett programmeringsspråk vilket stöds av SoapUI. Programmet är enkelt och effektivt, alla referensnummer var sparade i ett textdokument med ett referensnummer på varje rad. När programmet läst en rad utförde det ett test med hjälp av det referensnummer det just läst och resultatet sparades i vald databas. Med hjälp av MySQL Workbench gick det att skriva queries som väljer ut specifik information ur testen.

För att det skulle räknas som en förbättring skulle antalet frysningar av personer på listan med de 7177 oskyldiga köpen minskas, men antalet frysningar av personer som begått bedrägerier skulle minst vara samma. Om så inte är fallet skulle filtret vara sämre än tidigare. En annan möjlighet var att filtret frös lika många personer och fångade in fler personer från bedrägerilistan. Resultaten av dessa tester finns att beskåda i kap.7.

6.2 Krav

För att det nya filtret ska vara användbart för Resurs Bank måste den vara pålitlig. Därför finns det ett generellt krav som alla metoder måste uppnå och det är att koden ska gå igenom alla funktionalitetstester för att se till att metoden reagerar på det den ska reagera på och hanterar olika input på rätt sätt(se kap.5.1).

7 Resultat

Kapitlet kommer presentera resultaten av testerna beskrivna i kapitel 6.1. Det finns sektioner som inte får nämnas i detalj och som resultat av det kommer det mestadels finnas beskrivande bilder på hur resultaten ändrar sig vid olika tester. Av de nedanstående faserna beskriver varje fas en speciell konfiguration av filtret där fas 3 beskriver det viktigaste resultatet.

7.1 Fiktivt exempel

För att underlätta förståelsen för hur konfigurationen fungerar kommer ett fiktivt exempel att beskrivas. Fig.7.1.1. illustrerar hur resultaten presenteras efter ett test, filtret som har använts vid testet har tio olika metoder. I detta fiktiva exempel får metod 1 hantera vilket land man gör beställningen från och metod 7 hanterar antal varor i beställningen.

I konfigurationen sätts metod 1 till att ge low (L) om man gör beställningen från Sverige och medium (M) annars. Metod 7 konfigureras att sätta low om beställningen är på 1-5 varor, medium om antalet varor är 6-10 och annars high (H) om beställningen innehåller fler än 10 varor. För att filtret ska göra en frysning på ett köp konfigureras filtret så att det behövs 5 medium eller 3 high, dessa konfigurationer syns inte i figuren.

Referensnummer hämtas från MultiUpplys via ett textdokument och är här 01234567. När testet har körts fås resultatet som visas i fig7.1.1.

Analys
MultiUpplys referensnummer: 01234567
Extern referens:
Tidpunkt:
Total exekveringstid: 1845ms

Bedrägerimatris

Analysnamn	?	L	M	H	Resultat	Tid
metod 1		X				2 ms
metod 2		X				2 ms
metod 3		X				2 ms
metod 4		X				0 ms
metod 5		X				136 ms
metod 6		X				235 ms
metod 7			X			114 ms
metod 8		X				47 ms
metod 9		X				23 ms
metod 10			X			1194 ms
Totalt:		8	2			

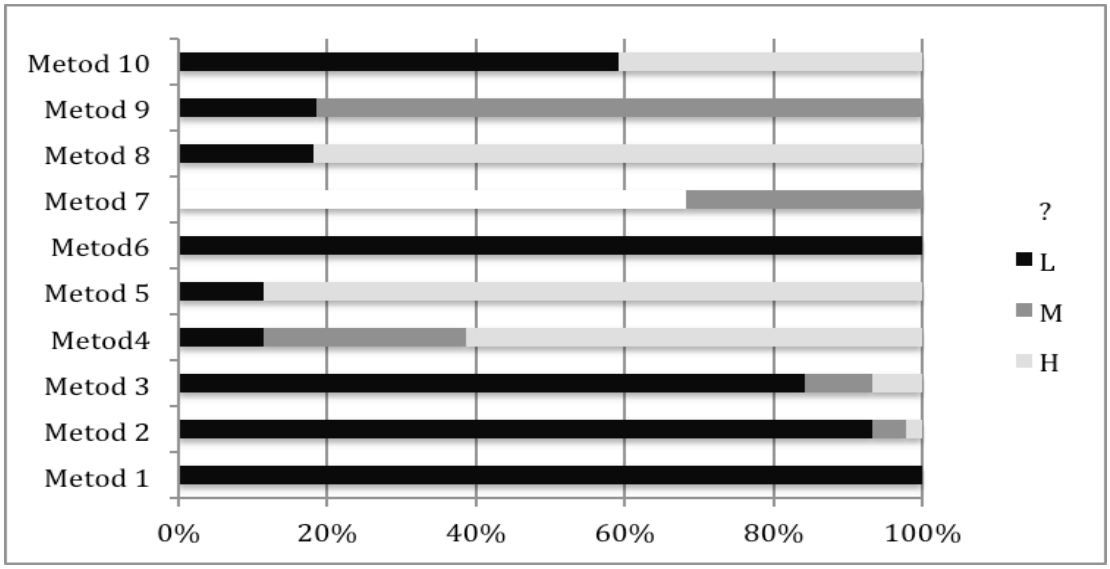
Rekommendation
Övergripande rekommendation enligt regelverk: Suspected fraud / Suspected not fraud

g.7.1.1 – Illustration av hur resultatet [illegible]

Fig.7.1.2 – Illustration av hur de olika metoderna har triggats.

Metod 1 som kontrollerar från vilket land beställningen gjorts blev low och det betyder att beställningen måste gjorts inom Sveriges gränser. Metod 7 blev däremot medium, det betyder att beställningen innehöll mellan 6 och 10 varor. Sammanlagt blev det två medium och inga high, vilket resulterade i ingen frysning.

I fig.7.1.2 kan man se ett fiktivt exempel på hur metoderna har triggats efter en fas. Detta underlättade vid konfiguration av filtret eftersom man fick en tydlig bild av hur testet hade gått.



7.2 Resultat av analys

Analysen visade att nya metoder i filtret behövdes. Efter att fört statistik av alla bedrägerier som begåtts mot Resurs Bank hittades det tillräckliga samband för att implementation av två nya metoder skulle bli aktuellt. Dessa metoder skulle förhindra att dessa samband går att nyttjas. Efter ett avstämningsmöte med bedrägerigruppen och handledarna bekräftades dessa påståenden ytterligare och implementation av de nya metoderna påbörjades.

7.3 Fas 1

För att få en utgångspunkt testades Resurs Banks befintliga filter med alla de oskyldiga köpen i den 7177 långa listan och resulterade i en fördelning enligt fig.7.3 och fig.7.4.2. Detta var utgångspunkten som även förklarades i kap.1.2. Önskvärt resultat hade varit att hela denna lista inte hade blivit frysta, med andra ord så skulle den mellersta ringen täckt samma yta som den innersta

ringen. När listan med bedrägerier kördes mot filtret frystes alla dessa av filtret och det var utgångspunkten inför kommande faser.

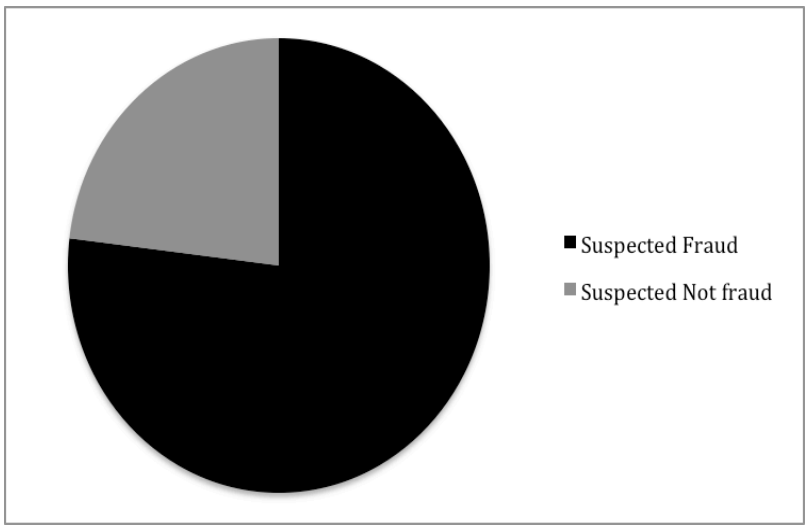
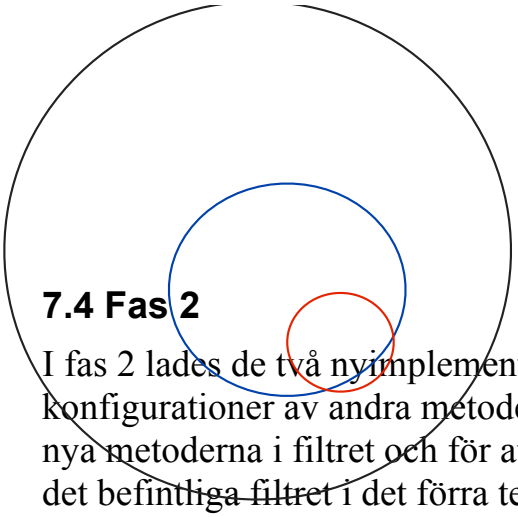


Fig.7.3 – Illustration av resultatet från fas 1.

Som kan ses i fig7.3 är en majoritet av listan med de oskyldiga köpen misstänkta för bedrägeri. Det ultimata resultatet hade varit om hela figuren var grön.



7.4 Fas 2

I fas 2 lades de två nyimplementerade metoderna till i filtret. Inga konfigurationer av andra metoder gjordes. Detta för att få en grund med de två nya metoderna i filtret och för att se om någon förändring skett jämfört med det befintliga filtret i det förra testet. I fas 1 och fas 2 är där ingen märkbar skillnad av den orsaken att de nya metoderna inte triggar så många nya samtidigt som de gamla metoderna triggar lika många. Detta test fryser 0,31 % fler än det föregående och eftersom de gamla metoderna har samma konfiguration som tidigare fångades hela listan med bedrägerier upp även här. Detta är inte ett önskvärt resultat. Resultatet av testet kan ses i fig.7.4.1. Fig.7.4.2. ger en mer övergripande bild av förhållandet mellan antal köp, antal frysningar och antal bedrägeri.

Fig.7.4.2 – Illustration av förhållandet mellan antal köp, antal frysningar och antal bedrägerier. OBS, endast approximativ.

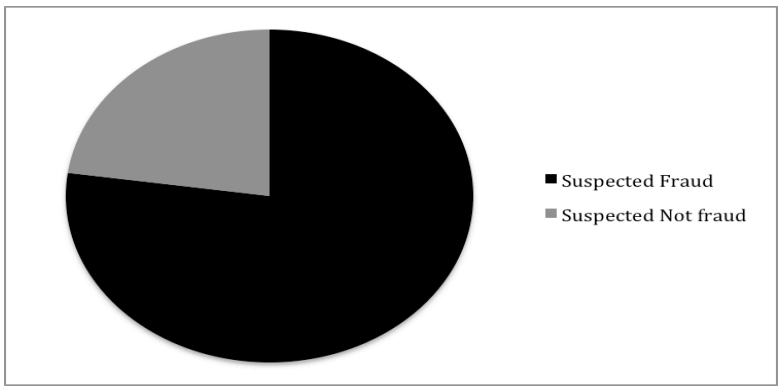
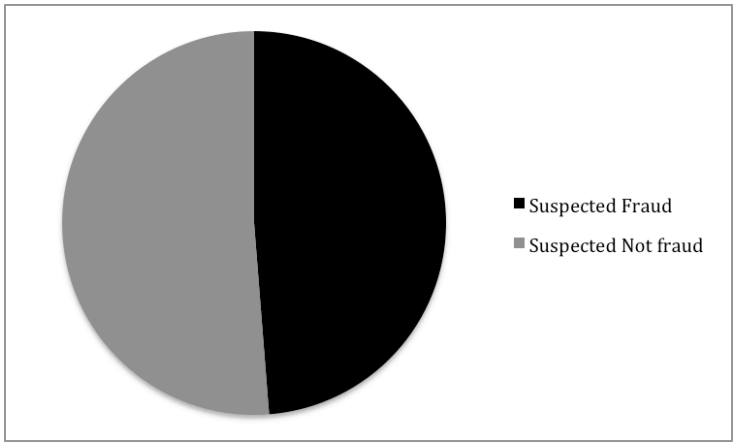


Fig.7.4.1 – Illustration av resultatet från fas 2.

g.7.5.1 – Illustration av resultat från fas 3.

7.5 Fas 3

Den stora förändringen skedde först i denna fas. Inför testningen i fas 3 ändrades konfigurationer för de metoder som skulle påverka utfallet mest. De föregående faserna gav en god inblick i vad som skulle behöva ändras. Eftersom filtret hade fler faktorer att trigga på kunde de gamla metoderna konfigureras att inte trigga lika tidigt som innan. I resultatet som syns i figuren nedan kan man se att mellersta ringen minskats betydligt medan den innersta ringen inte har förändrats. Detta betyder en effektivisering av filtret och antalet misstänkta bedrägerier minskades med 37 % jämfört med Resurs Banks nuvarande filter i fas 1. Det innebär avsevärt mindre jobb för bedrägerigruppen, i det avseendet att färre ärenden behöver hanteras manuellt.



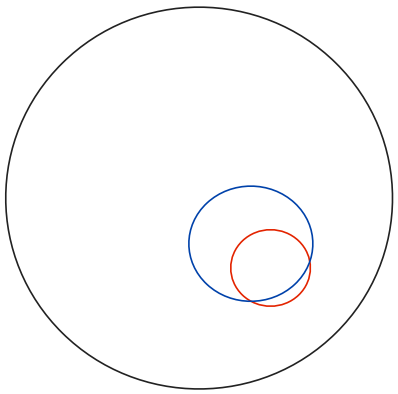


Fig.7.5.2 – Illustration av förhållanden mellan antal köp, antal frysningar och antal bedrägerier. OBS, endast approximativ.

7.6 Utlåtande

För att förstå hur dessa nya metoder påverkar banken har tre intressenter gjort ett utlåtande om deras syn, dessa citeras nedan.

Damir Kafedzic (2013-05-22)

> *”De två nya kontrollerna som presenterats kommer att påverka våra bedrägerisiffror markant. Kostnaden per handlagt ärende kommer att minska då vi har mer data att utgå ifrån. Vi kommer att ha färre manuella påslag vilket betyder att kostnaden för att bedriva vår verksamhet kommer att sjunka.”*

Johan Ljungström (2013-05-22)

> *"Med hjälp av de nya metoderna kan vi nu fokusera oss på mer väsentliga faktorer än tidigare samtidigt som vi har en god grund att bygga vidare på inför framtiden. Dessa två nya metoder kommer att minska våra manuella kontroller väsentligt vilket betyder att kostnaden för att förhindra bedrägerier kommer minska.”*

Pär Nilsson (2013-05-24)

> *”Niclas och Jonas har tillsammans med sina kontaktpersoner i verksamheten kommit fram till hur vi ska kunna förbättra vår bedrägerikontroll. Detta har inneburit kodförändringar i bedrägerikontrollen, där man har byggt ut den befintliga funktionaliteten. De har med sina kodförändringar lyckats göra en djupare bedrägerikontroll genom att samla in information från fler av våra befintliga system detta för att kunna ge ett bättre omdöme av ett enskilt köp. Niclas och Jonas har bevisat förbättringarna genom att köra samma data genom bedrägerikontrollen med olika konfigurationer. Jag tror att banken kommer ta inspiration av deras sätt att testa bedrägeri-konfigurationer i framtiden.”*

8 Slutsats

Examensarbetets inriktning har tagit en del svängar under arbetets gång. Vid analysens start gavs tillgång till information som skulle hjälpa att hitta samband inför elicitering av nya metoder. I detta stadie var fokus på att endast skapa nya metoder för att filtret skulle kunna frysa på ännu fler faktorer och det gjorde att den mest väsentliga delen åsidosattes, att effektivisera filtret eller med andra ord minska andelen onödiga frysningar. Denna del kom ändå att bli en stor del av projektet fast under ett senare skede.

8.1 Processens gång

Arbetsprocessen som visas i fig.2.1 gav en bra struktur att arbeta efter. Analysen bestod av att hitta svar på frågeställningarna där det kom fram att det inte fanns några kryphål (problemformulering b), istället lades tid på att hitta gemensamma nämnare för de bedrägerier och bedrägeriförsök som skett (problemformulering c). Det har därför implementerats metoder som kontrollerar de gemensamma nämnare som hittades vid analysen. Problematiken med analysen visade sig vara att all information kring tidigare bekräftade bedrägeriförsök hanteras manuellt och av olika personer. Detta har gjort att information har varierat, statistiken har varit trovärdig dock ibland ofullständig, vilket gav en från början lite oklar bild kring hur bedrägerifiltret skulle förbättras. Det har därför även vid analysen tagits hjälp av bedrägerigruppen som har haft bättre insikt i gemensamma nämnare för bedrägerier. Tillvägagångssättet var en kort del i arbetsprocessen där det kom fram hur metoden ska implementeras.

Implementationen av metoderna var det som förväntades ta längst tid, vilket var en av anledningarna till att implementationen påbörjades i ett tidigt skede. Med kunskaper från skolan och genomgång av IntelliJ och av bedrägerisystemet gick implementation av de två nya metoderna snabbare än förväntat.

8.2 Resultat

För att se om de nya metoderna bidrog med förändring behövde de gå igenom en verifiering vilket förklaras i kap.6. Detta gjordes i SoapUI och denna del var den som tog längst tid av projektet, detta berodde på att förkunskaperna var obefintliga. Utifrån dessa tester kunde sedan frågeställningarna kring effektiviseringen av filtret besvaras. Efter granskning av fig.7.4.2 och fig.7.5.2 ser man en stor förändring på storleken av den mellersta ringen. Denna del har minskat kraftigt samtidigt som de köp som är bedrägeri fryses vilket betyder

att filtret går att effektivisera utan att för den delen minska säkerheten, vilket bekräftar frågeställning d). Tyvärr går det inte att avgöra om säkerheten har ökats då statistiken kring bedrägeri och bedrägeriförsök har varit för tunn, följaktligen kan frågeställning e) inte avgöras.

8.3 Framtid

Resultatet från fas 3 var till freds och förbättringen var mer påtaglig än vad som misstänktes från början. Bedrägerigruppens arbete kan minskas med mer än en tredjedel vilket innebär stora besparingar i både resurser och tid. Självklart finns det fler möjligheter att förbättra ännu mer för Resurs Bank. Bland andra har Damir Kafedzic och Johan Ljungström bättre förståelse för vilka konfigurationer (för de nya metoderna) som kan vara mest effektiva än vad vi har, metoderna är implementerade på så sätt att Johan och Damir kan utnyttja dem på det mest effektiva sätt. Skulle de bygga vidare på filtret i test nummer 3 (kap7.5) kan resultatet förmodligen bli ännu bättre.

En annan sak som kan förbättras i framtiden skulle vara att dela upp den befintliga bedrägerilådan ännu mer. Bedrägerilådan består idag av två delar (omnämns i kap 1.5) och dessa flyter dessvärre ihop lite. Om gränserna varit tydligare hade det blivit enklare att hantera båda delarna. Resurs Bank har påbörjat denna uppdelning under vår tid på Resurs Bank och kommer då att lyfta ut de delar som hör hemma i kreditupplysningen från bedrägerifiltret, vilket betyder en metod mindre som triggar på icke bedrägeri nämnare.

En rekommendation till Resurs Bank skulle vara att skapa en databas för att på ett enkelt sätt spara undan bedrägerier och bedrägeriförsök och även försöka spara undan bedrägeri som lyckats. Då skulle man ha en standard på vilken information som ska sparas undan och den skulle även vara lättillgänglig för de som behöver den. Detta skulle underlätta betydligt vid framtida tester och att hitta gemensamma nämnare. Skulle man lyckas spara undan mer information om lyckade bedrägerier kan man enklare testa om filtrets säkerhet ökar vilket har varit svårt att göra.

9 Terminologi

Bedrägerigruppen	En grupp som jobbar med att avgöra om personerna i ansökningarna är de som de utger sig vara och avgör om det är ett bedrägeriförsök.
Bedrägerilådan	Där bedrägerifiltret finns
CIO	Chief Information Officer, chef på direktörsnivå
CRUD	Skapa, läsa, uppdatera och ta bort
Exception	Ett undantag, kastas av programmet när något oväntat händer
Grålista	Lista med personer som försökt göra bedrägeri
Intressent	Personer eller roller som direkt eller indirekt påverkas av systemet
Javagruppen	Personerna som jobbar med java på Resurs Bank
Kryphål	Där bedrägerifiltret inte kontrollerar
Web service	Ett datorprogram som kan utbyta information med andra web servicar över internet
WSDL	Web Service Definition Language, ett sorts XML-format
XML	Programmeringsspråk, används för att beskriva t.ex. en web service.

10 Referenser

[1.1]
Kunder & Historik. (2012). Hämtad 15 februari 2013 från:
http://www.resursbank.se/om-resurs-bank/kunder-historik/

[1.7.1]
IntelliJ IDEA :: Features. (2013). Hämtad 19 mars 2013 från:
http://www.jetbrains.com/idea/features/index.html

[1.7.2]
What is soapUI?. (2013). Hämtad 19 mars 2013 från:
http://www.soapui.org/About-SoapUI/what-is-soapui.html

[1.7.3]
MySQL :: MySQL Workbench. (2013). Hämtad 19 mars 2013 från:
http://www.mysql.com/products/workbench/

[1.7.4.1]
MongoDB. (2013). Hämtad 20 mars 2013 från:
http://www.mongodb.org/

[1.7.4.2]
Introduction to MongoDB. (2013). Hämtad 15 maj 2013 från:
http://www.mongodb.org/about/introduction

[1.7.5]
Maven – What is Maven?. (2013). Hämtad 20 mars 2013 från:
http://maven.apache.org/what-is-maven.html

[1.7.6]
Apache Subversion Features. (2013). Hämtad 15 maj 2013 från:
http://subversion.apache.org/features.html

[1.7.7]
Calc Features >> LibreOffice. (2013). Hämtad 26 april 2013 från:
http://www.libreoffice.org/features/calc/

[2]
Henrik Kniberg, (2009-06-29), Version 1.1 *Kanban vs Scrum*, 5
http://www.crisp.se/file-uploads/Kanban-vs-Scrum.pdf

[4]
Web Service Definition Language. (2001). Hämtad 29 april 2013 från:
http://www.w3.org/TR/wsdl