

Datavetenskap

Åsa Magnusson

Bo Sundin

**Granskning och klassificering av
programvaruproblem ur ett semantiskt
perspektiv**

Examensarbete, C-nivå

2000:29

**Granskning och klassificering av
programvaruproblem ur ett semantiskt
perspektiv**

Åsa Magnusson

Bo Sundin

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Åsa Magnusson

Bo Sundin

Godkänd, 2000-06-31

Handledare: Eivind J. Nordby

Examinator: Stefan Lindskog

Sammanfattning

Denna rapport bygger på en empirisk undersökning av felrapporter för två helt skilda programvaruprodukter. Den fokuserar på semantiska problem vid utveckling av programvaror. Ordet semantik beskrivs, både allmänt och ur undersökningens perspektiv. Arbetsgången är beskriven i två olika kapitel som är upplagda på samma sätt. En viktig del av denna rapport är en diskussion om hur klassificeringen av felrapporterna gick till. Rapporten mynnar ut i ett statistiskt resultat som sedan analyseras och diskuteras. Resultatet blev att många av felen kan bestämmas till att vara semantiska men de kan inte med säkerhet härledas till brister i specifikationen vid programvaruutveckling. Orsaken till osäkerheten i klassificeringen är att undersökningen gjordes på ett stort antal felrapporter. Tiden räckte inte till att undersöka varje felrapport grundligt. Rapporten innehåller även ett antal förslag på åtgärder för att minimera risken för att vissa problem uppkommer. Det sista kapitlet i denna rapport behandlar de problem vi stött på och ett förslag på hur undersökningen skulle ha utförts för att få ett säkrare resultat.

Review and Classification Of Software Problems from a Semantic Perspective

Abstract

This report is based on an empirical investigation and analysis of software Trouble Reports (TR). The report focuses on semantic problems that arise during software development. The word “semantics” is described both from a general perspective as well as from the perspective of this specific investigation. The investigation is based on data from two distinctly separate products. The workflow is described in two different chapters with similar structure. An important part of this report is the elaboration regarding the process of classification. The general conclusion of this report is presented as statistical data, which is then analysed and discussed. One basic conclusion is that a lot of SW faults can be classified as semantic faults, however they cannot conclusively be linked to gaps of semantic nature in the documentation used within software development. One cause for this uncertainty in classification is that this investigation is based on a large number of TRs, and time limitations prevented a thorough analysis of each TR. There is a number of proposals for actions to minimize the probability for certain problems. The last chapter in this report deals with the problems we have encountered and contains suggestions for how this investigation could have been improved with respect to more conclusive results.

Tack

Vi vill framföra vårt tack till Eivind Norby, vår handledare som har hjälp och stöttat oss i vårt arbete med denna rapport. Utan hans stöd hade detta examensarbete aldrig blivit klart. Ett stort tack vill vi också framföra till Per Strömgren vår kontakt på Ericsson Infotech/N. Han har lagt ner mycket tid och arbete för att hjälpa oss. Vi vill även tacka Mikael Lindh på Ericsson Infotech/N som gav oss en djupare inblick i produktutvecklingen. Helena Lindskog och Niklas Sandberg på Ericsson Infotech/L är också förtjänta av ett stort tack, de har bidragit med tid och svarat på alla våra frågor. Till slut vill vi tacka nämnda avdelningar på Ericsson Infotech, för att vi fått ta del av allt material som behövts för att kunna genomföra vår undersökning.

1	Inledning.....	1
1.1	Rapportens uppbyggnad.....	1
2	Terminologi.....	2
2.1	Semantik.....	2
2.1.1	Microsoft Encarta	2
2.1.2	Nationalencyklopedin.....	3
2.1.3	Chalmers tekniska högskola.....	3
2.1.4	Semantik ur vårt perspektiv.....	3
2.2	Förkortningar	4
3	Bakgrund för examensarbetet.....	5
3.1	Introduktion.....	5
3.2	Problemformulering	5
3.3	Mål	5
4	Allmänna förberedelser	6
5	Ericsson SS7.....	6
5.1	Presentation av produkten	6
5.2	Resurspersoner	7
5.3	Material	7
5.4	Tillvägagångssätt	8
5.4.1	Förberedelser	8
5.4.2	Bearbetning	8
5.4.3	Statistisk bearbetning.....	10
5.4.4	Möten	11
5.4.5	Diskussion om klassificering.....	11
5.5	Klassificering	13
5.5.1	Kriterier	13
5.5.2	Tillämpning	14
5.6	Resultat.....	15
5.7	Tolkning av resultat och förslag till åtgärder.....	15
6	WBS.....	17
6.1	Presentation av produkten	17
6.2	Resurspersoner	18
6.3	Material	18
6.4	Tillvägagångssätt	18
6.4.1	Förberedelser.....	18
6.4.2	Bearbetning	19
6.4.3	Statistisk bearbetning.....	19
6.4.4	Möten	19
6.4.5	Diskussion om klassificering.....	19
6.5	Klassificering	20

6.5.1	Kriterier	20
6.5.2	Tillämpning	20
6.6	Resultat	21
6.7	Tolkning av resultat och förslag till åtgärder	21
7	Jämförelser av resultat.....	22
8	Erfarenheter.....	23
8.1	Problem	23
8.2	Förslag till annorlunda arbetssätt	23
	Referenser.....	25
A	Bilaga	27
A.1	Mall för Trouble Report för SS7	27
A.2	Mall för History file för SS7	29
A.3	Schema för arbetsgången för SS7	31
A.4	Mall för TRI för WBS.....	33

1 Inledning

Denna rapport är skriven som ett examensarbete i datavetenskap vid Karlstads universitet och redovisar en undersökning om de bakomliggande orsakerna till varför fel uppstår under utveckling av programvara. Arbetet har gjorts inom ramen för SKUTT, ett forskningsprojekt om semantisk kvalitet vid datavetenskap. Undersökningen fokuserar på semantik och semantiska fel och bygger på en fallstudie av två olika programvaruprodukter. Den ena, Ericsson Signal System No.7, förkortat till Ericsson SS7, implementerar signalhantering för informationsutbyte i digitala telekommunikationsnätverk. Kommunikation sker över SS7 som är ett protokoll och en världstäckande standard för telekommunikation. Produkten är skapad ur ett långsiktigt perspektiv och har existerat i cirka 10 år. Den andra produkten, WBS, utvecklades för att användare av WAP-telefoner ska kunna utföra olika WEB-baserade tjänster. Produkten utvecklades under några få intensiva månader och den beräknade livslängden var cirka 2 år.

Vår uppgift har varit att gå igenom ett antal felrapporter och bedöma och klassindela dem med avseende på om felkällan kan anses vara semantisk otydlighet eller inte. Resultatet av arbetet sammanställs för en statistisk analys, som skall kunna användas i SKUTT:s fortsatta arbete.

1.1 Rapportens uppbyggnad

Kapitel 2 beskriver ordet semantik både med hjälp av olika uppslagsverk och med våra egna ord. Dessutom innehåller kapitlet en förklaring av förkortningar som används i rapporten. Kapitel 3 beskriver bakgrunden till rapportens uppkomst. Här finns även en problemformulering och en beskrivning av de mål som skall uppnås. Kapitel 4 ger en bild av de inledande förberedelserna.

Kapitel 5 och 6 är strukturerade på samma sätt och beskriver var sin produkt som undersökts. De börjar med en kort presentation av produkten och de resurspersoner vi har haft tillgång till. Sedan beskrivs det material som vi har haft tillgång till och använt oss av. Dessa kapitel beskriver också arbetsgången och visar hur klassificeringen växt fram. Sedan resultaten

presenterats och analyserats innehåller kapitlen även en kommentar till resultatet och ger förslag på lösningar för problemen.

I kapitel 7 jämförs resultaten av undersökningen för de båda produkterna. Både likheter och skillnader tas upp. En diskussion förs om orsaken till att resultaten ser så olika ut. I kapitel 8 finns en reflektion om själva undersökningen och upplägget på denna. Här finns även ett förslag till en annorlunda arbetsgång för att få en säkrare bedömning av felorsaken. Ett avsnitt i det kapitlet tar upp de olika problem vi stött på under arbetet.

Som bilagor finns några exempel på dokument som använts under arbetets gång, mallar för hur de olika felrapporterna ser ut. Det finns även en mall som visar vilka rubriker en historikfil skall innehålla. En av bilagorna är ett schema som visar arbetsgången hos Ericsson Infotech/N och vilka dokument som finns att tillgå.

2 Terminologi

Detta kapitel används för att definiera ordet semantik. Det innehåller även vår beskrivning av ordet och en lista över förkortningar som används i rapporten.

2.1 Semantik

Här kommer några citat ur olika uppslagsverk som förklarar ordet semantik. Till slut förklarar vi hur vi använder ordet i denna rapport.

2.1.1 Microsoft Encarta

Semantics (Greek *semantikos*, “significant”), the study of the meaning of linguistic signs—that is, words, expressions, and sentences. Scholars of semantics try to answer such questions as “What is the meaning of (the word) *X*?” They do this by studying what signs are, as well as how signs possess significance—that is, how they are intended by speakers, how they designate (make reference to things and ideas), and how they are interpreted by hearers. The goal of semantics is to match the meanings of signs—what they stand for—with the process of assigning those meanings. [5]

2.1.2 Nationalencyklopedin

Semantik (fr. *semantique*, av grek. *semantiko's* 'betecknande', ytterst av se'ma '*tecken', 'signal') studiet av språkliga uttrycks betydelse eller mening. Semantiken har varit föremål för intresse inom olika discipliner, framförallt språkvetenskap, filosofi och logik men även inom antropologi, psykologi och litteraturvetenskap och på senare tid också kognitionsforskning och artificiell intelligens.

Inom datalogi ägnas programspråkens semantik stor uppmärksamhet och området är ett livligt forskningsområde sedan 1970-talet. Man vill garantera att ett programspråk implementeras på samma sätt för alla datorer; ett program skall ge samma resultat oberoende av vilken dator som exekverar det. [6]

2.1.3 Chalmers tekniska högskola

I en kursinformation från deras hemsida hittade vi följande beskrivning.

En semantik för ett programspråk är ett sätt att definiera betydelsen hos program i språket, för att kunna resonera matematiskt-logiskt om programmet och dess uppförande.

2.1.4 Semantik ur vårt perspektiv

Semantik står för tolkning av innebörden (att tolka saker på rätt sätt). Omformulerat blir det, utför programvaran det som skall utföras, utför de olika modulerna vad de skall utföra? Följs de standarder och specifikationer som finns?

Rapporten behandlar semantiken vid utvecklandet av programvaror och har sin tyngdpunkt på var problem uppstår. Denna aspekt av semantik rör de olika specifikationer som finns och hur de är skrivna. Berörda dokument är de som skrivs före implementation. Finns t.ex. all funktionalitet klart beskriven i en specifikation så att personen som implementerar modulen inte kan missförstå vad modulen skall utföra? Semantiken för programvaror rör de olika gränssnitten i ett program. Att olika gränssnitt är beskrivna olika ligger i sakens natur, men de måste ändå vara entydiga. Kravspecifikationens utformning är av största vikt, för om ett fel uppstår i förberedelsefasen kan det bli mycket svårt och kostsamt att avhjälpa felet.

Rapporten tar alltså inte upp någonting som har med programspråkens semantik att göra. Denna stora fråga ligger utanför vår problemställning och behandlas i många andra forskningsprojekt.

2.2 Förkortningar

ANSI	American National Standards Institute
API	Application Programmer's Interface
CR	Custom Request
FD	Function Description
FS	Function Specification
IP	Implementation Specification
ITU	International Telecommunication Union
NUTEK	Närings- och teknikutvecklingsverket
RS	Requirement Specification
SDS	System Design Specification
SKUTT	Ökad programvarukvalite genom semantisk beskrivning
TPR	Trouble/Problem Report
TR	Trouble Report
TRI	Trouble Report Internal
TTC	The Telecommunication Technology Committee
USC	User Service Center
WAP	Wireless application protocol
WBS	WAP Basic Services

3 Bakgrund för examensarbetet

Med detta kapitel vill vi belysa orsaken till att denna undersökning utförts. Det innehåller även problemformulering och beskriver målet med undersökningen.

3.1 Introduktion

Bakgrunden till utredningen som gjorts, är att avdelningen för datavetenskap vid Karlstads universitet bedriver ett forskningsprojekt. Det utförs i samarbete med Ericsson Infotech AB och stöds av NUTEK-programmet ”Komplexa Tekniska System”. Forskningsprojektet heter ”Ökad programvarukvalitet genom semantisk beskrivning” och kallas SKUTT¹. SKUTT fokuserar på semantisk kvalitet och semantisk integritet. Forskningen går ut på att ta fram en metod för att höja den semantiska kvalitén och integriteten i programutvecklingen. SKUTT är intresserade av hur det ser ut i verkligheten på arbetsplatser vad det gäller semantik och har därför initierat vår undersökning. I SKUTT ingår Anna Brunström, Eivind Nordby som är vår uppdragsgivare och handledare samt Martin Blom.

3.2 Problemformulering

Vi skall kartlägga om det är semantiska eller andra orsaker som ligger bakom rapporterade problem vid utvecklingen av programvaror. Vi vill undersöka om semantiken hanteras på ett intuitivt eller klart beskrivet sätt. I uppgiften ingår även att ta fram ett statistiskt material om felfördelningen.

3.3 Mål

Målet är att göra en objektiv bedömning av hur stor andel av felen i de undersökta produkterna som kan spåras till semantiska brister i specifikationerna som används vid utvecklandet av programvaran. Vi ska klassindela de rapporterade felen, dra slutsatser om och jämföra orsakerna till uppkomsten av fel i de två produkterna. Vår rapport ska kunna användas av SKUTT som en del i forskningsprojektet.

¹ SKUTT kan med en smula fantasi tänkas härledas från ”Semantisk Kvalitetsutveckling”, som vore en lättare omskrivning av projektets namn.

4 Allmänna förberedelser

Vi träffade Eivind Nordby och Martin Blom på ett kickoffmöte där vi fick en del material att läsa in, Martin Bloms D-uppsats [1], ett kompendium som sammanfattade Bertrand Meyers teorier om lösningar på semantiska problem[4] och ett kompendium från Chalmers Tekniska Högskola[3]. Materialet gav oss en allmän inblick i semantiska frågor och förståelse för vad semantiska problem handlar om. Vi fick en mer ingående förklaring på vad ex-jobbet gick ut på samt namn och telefonnummer till kontaktpersoner på Ericsson Infotech.

För att få större inblick i vad SKUTTs forskning handlar om, har vi deltagit i ett antal möten där utvecklingen av deras metod Semla [2] har diskuterats och metodbeskrivningen redigerats. Vi deltog aktivt i redigeringen, och fick en inblick i hur en teknisk rapport kan se ut.

5 Ericsson SS7

Detta kapitel presenterar produkten Ericsson SS7 och de resurspersoner vi hade till vårt förfogande under examensarbetet. Här beskriver vi materialet som använts och hur bearbetningen av det gått till. Kapitlet tar upp en diskussion om utvecklingen av klassificeringen, och hur den till slut kom att se ut. Den statistik som blev resultatet av undersökningen redovisas och en kort analys görs. Kapitlet innehåller även en tolkning av resultatet och förslag på vad som skulle kunna göras för att undvika en del av felen.

5.1 Presentation av produkten

Ericsson Signal System No.7, förkortat till Ericsson SS7, är ett internationellt kommunikationsprotokoll och en världstäckande standard för telekommunikation. Ericsson SS7 implementerar signalhantering för informationsutbyte i digitala telekommunikationsnätverk. Produkten säljs över hela världen och är därför anpassad till olika länders telefonstandarder t. ex. ANSI, ITU och TTC.

Ericsson SS7 är en produkt med lång livslängd som varit ute på marknaden i cirka 10 år och engagerar cirka. 100 personer. Den består av ett antal moduler som modifieras antingen tillsammans eller var och en för sig. Modifieringar görs på en kunds begäran eller när man

inom företaget anser att så behövs, t.ex. när en ny standard har definierats. Produkten säljs till företag som bygger applikationsprogram ”ovanpå” Ericsson SS7 för en speciell teleoperatörs tillämpning. Vidarekoppling, telefonsvarare i nätet (t.ex. Telias Mobilsvar) och gratisnummer (t.ex. 020) är exempel på typiska tillämpningar.

5.2 Resurspersoner

Den kontaktperson vi hade på Ericsson Infotech/N heter Per Strömgren och arbetar som Quality Controller för bland annat SS7. Han har hand om kvalitetssäkringen för produkten. Som kontaktperson har han försett oss med allt material som vi haft behov av. Han har även varit en diskussionspartner för olika problemställningar.

Mikael Lindh är en systemarkitekt som tidigare arbetat på supportavdelningen för Ericsson SS7. Det gör att han är mycket insatt i de felkorrigeringar som gjorts på produkten. Han kunde ge oss en bild av hur produktutvecklingen går till i ”verkstan”. Kontakten med Mikael Lindh förmedlades av Per Strömgren.

5.3 Material

Från SS7 har vi fått 234 felrapporter. Varje felrapport har en titel som i korthet beskriver felet, en avsändare, en feltyp och en lite utförligare beskrivning av felet (bilaga A.1). Vissa felrapporter innehåller även förslag på lösningar till problemet. Dessa rapporter gäller fel som är inkomna under 1999 och är klassade som A- eller B-fel. A-fel innebär att produkten är helt obrukbar och att felet måste åtgärdas omedelbart, B-fel innebär att vissa funktioner inte fungerar men att produkten går att använda. Här har man alltså lite tid på sig att rätta till felet.

I vårt val av material ingick även historik-filer (bilaga A.2), för de moduler som felrapporterna gäller. I dessa filer kan man spåra ett TR-nummer, som är numret för den aktuella felrapporten. Historiken innehåller en kort beskrivning av felet och vilka åtgärder som vidtagits. Det kan även utläsas i vilken fas av utvecklingen felet borde ha upptäckts, vilka andra moduler som har påverkats av rättningen samt vem som åtgärdat felet. Historikfilen ger en möjlighet att följa modulens livscykel.

Vi har haft fri tillgång till alla dokument vi haft behov av. Det fanns även möjlighet att ställa muntliga frågor, både till Per Strömgren och Mikael Lindh.

5.4 Tillvägagångssätt

Under denna rubrik presenterar vi de olika faser som undersökningen genomgån. Här beskrivs även det grovarbete som utförts. För att underlätta läsningen av rapporten har vi valt att dela upp detta avsnitt i flera underavsnitt baserade på arbetsområde. Detta trots att arbetet med de olika delarna naturligtvis inte var separerade utan gjordes växelvis.

5.4.1 Förberedelser

Vid vårt första möte med Per Strömgren hade han en liten genomgång av Ericsson SS7. Därefter blev det en lång diskussion om vad vi var intresserade av och vilka dokument som skulle vara av intresse. Vi fick lite olika sorters dokument för att se vilket material vi behövde och även att ha som diskussionsunderlag med vår uppdragsgivare. Med Per Strömgren diskuterades även vilka fel vi var intresserade av. Eftersom vårt intresseområde berörde semantik kom diskussionen in på gränssnitt och en sökning på API gjordes. Sökningen fungerade inte då kopplingen mellan gränssnitt och API inte fanns i felrapporternas rubriker. Vi kom fram till att alla A- och B-fel under 1999 var mest intressanta ur vår synvinkel. C-felen ansåg vi inte vara relevanta för undersökningen, då dessa var av estetisk typ och inte påverkade funktionaliteten hos produkten.

Utifrån detta material beslutade vi oss för att slumpa en provserie på 10 fel för att få en uppfattning om det går att dra några relevanta slutsatser utifrån materialet. Till de 10 felen fick vi felrapporter, historikfiler och en FS, funktionsspecifikation. För ett fel fick vi även kod då det saknades historikfil.

Efter att ha bearbetat provserien bedömdes att vi var på rätt väg och beslutade att fortsätta på samma sätt. Vi bestämde oss även för att i huvudsak använda felrapporterna och historikfilerna under vårt fortsatta arbete. Orsaken till att inte generellt använda funktionsspecifikationerna var att de var svåra att leta i och mycket omfattande. Vi skulle istället införskaffa de funktionsspecifikationer vi ansåg oss behöva senare.

5.4.2 Bearbetning

Steg ett i bearbetningen var att koppla ihop TR-numret med rätt historikfil för att underlätta fortsatt arbete. Vi försökte ett antal olika angreppssätt, då namnen på modulerna i TR-databasen i vissa fall inte stämde överens med namnet på historikfilen, som hade modulens namn. Det var ganska besvärligt och tog en hel del tid i anspråk. Till slut kom vi fram till att

sökning skulle ske på det specifika TR-numret i alla historikfiler. Då historikfilerna i de flesta fall var stora och komplexa så klippte vi ut de relevanta delarna och sammanställde egna filer. Dessa filer kunde underlätta arbetet om det skulle finnas anledning att granska felet igen.

Vi startade med att dela upp felrapporterna och försöka klassificera felen. Detta fungerade dåligt, då vi ganska snart upptäckte att det blev diskussion om nästan varenda rapport. Vi hade olika tolkningar av orsaken till felet. Slutsatsen vi drog var att det effektivaste sättet att arbeta på var att gemensamt arbeta med samma felrapport. Detta skulle underlätta och effektivisera klassificeringen.

Vi läste varje felrapport och historikfil tillsammans och diskuterade vad som kunde ha orsakat felet. Utifrån dessa två dokument försökte vi klassificera felen. I tveksamma fall bestämde vi oss för att söka efter mer information, antingen muntligt eller skriftligt. Detta arbete gjordes för de första 100 felrapporterna, för att få en uppfattning av tidsåtgången. Vår bedömning blev att tiden skulle räcka till för att bearbeta 50 stycken till. När vi var klara med dessa så kunde vi konstatera att material saknades för 20 fel. Vi beslutade oss därför för att behandla ytterligare 20 felrapporter.

Vår uppfattning till att börja med, var att all programmering gjordes utifrån en funktionsspecifikation. Senare visade det sig att funktionsspecifikationen och FD, funktionsbeskrivningen, för det mesta skrevs av den person som implementerade modulen. När detta kom till vår kännedom, förlorade funktionsspecifikationen en stor del av sin betydelse för vår undersökning. Det är inte troligt att programmeraren missförstår sig själv. Funktionsspecifikationen är det dokument som följer med produkten, den kan användas för att utveckla applikationer som ligger ”ovanpå” protokollstacken. Funktionsbeskrivningen är ett internt dokument som används för implementering.

När vi gått igenom 170 felrapporter var nästa steg att börja om med de 100 första rapporterna. Detta var nödvändigt då vi insåg att vår syn på klassificeringen under arbetets gång hade förändrats. Ett fåtal av dessa felrapporter omvärderades.

Utifrån felrapporter och historikfiler indelade vi felen enligt en preliminär klassindelning. Till klass 1 hör de fel som inte var semantiska. Här hittar man rent slarv, syntaktiska fel och logiska fel. I grupp 3 som blev stor, hamnade alla de fel som vi tror är semantiska men där vi

anser att vi har för lite att gå på för att säga det med säkerhet och alltså måste söka mer information. Grupp 5 innehöll bara ett fåtal fel som klart kunde härledas till brister i den semantiska specifikationen. Ytterligare en klass skapades för odefinierbara fel.

5.4.3 Statistisk bearbetning

Vi slumpade ordningen för de 234 felrapporterna då vi inte hade en aning om hur lång tid det skulle ta att klassificera felen. Då skulle undersökningen anses statistiskt korrekt, oberoende av hur många felrapporter vi hann med, och kunna användas inom SKUTT som en del av deras forskning.

För att kunna ta fram statistik runt materialet använde vi oss av en formel för intervallskattning[5].

$$\hat{P} \pm 1,96 * \sqrt{\frac{\hat{P} * (1 - \hat{P})}{n}} * \sqrt{\frac{N - n}{N}}$$

$N =$ Hela populationen

$n =$ Den undersökta populationen

$\hat{P} =$ Andelen i procent ur n som en viss grupp har

Från en tabell tas siffran 1,96 fram [5]. Den ger, insatt i formeln ovan, att man med 95 % sannolikhet kan bestämma avvikelserna i det slumpade materialet. Beräkningen går till så att procentsatsen först räknas ut för de olika grupperna. Denna siffra läggs in i formeln.

Ex

Klass 0 var 16,5% av den undersökta populationen, när detta sätts in i formeln får man:

$$0,165 \pm 1,96 * \sqrt{\frac{0,165 * (1 - 0,165)}{170}} * \sqrt{\frac{234 - 170}{234}} = 0,165 \pm 0,0289$$

Med denna uträkning visas att med 95% säkerhet återfinns mellan 13 och 20% av denna feltyp i hela populationen. Denna uträkning har gjorts för de andra klasserna och redovisas i avsnitt 5.6.

5.4.4 Möten

I inledningsskedet hade vi ett antal möten med Per Strömgren. Syftet med dessa möten var dels att få en inblick i produkten som sådan men också att få det material som behövdes. Under perioden som vi arbetade med klassificeringen så hade vi inget behov av möten.

När arbetet med klassificering närmade sig slutet hade vi samlat på oss ett antal frågor. Vi kontaktade Per strömgren för att bestämma en tid. Till mötet hade han på önskemål från oss konsulterat ingenjör Mikael Lindh. En del av våra frågor rörde rent konkreta felrapporter, andra handlade om arbetsgången för produkten (se bilaga A.3). Detta möte var väldigt givande, då Mikael Lindh och Per Strömgren är mycket kunniga och kunde ge oss en övergripande inblick i arbetet och arbetsgången. Mikael Lindh hade även stor kunskap om felrapporterna då han hade skrivit vissa av dem och rättat andra. Detta gjorde att han kunde svara på de specifika frågor vi hade om dessa. För att kunna se hur dokumenten är utformade och om det finns några semantiska brister i dem, fick vi fler dokument, en kravspecifikation (RS) och en funktionsbeskrivning (FD).

5.4.5 Diskussion om klassificering

Under arbetets gång har vår syn på klassificeringen genomgått flertalet förändringar och vi har haft ett antal diskussioner med vår handledare om hur dessa skall definieras.

Vid första kontakten med vår handledare var diskussionen fokuserad på att undersöka vilken sorts fel som hade högst frekvens. Vår syn på klassificeringen var mycket vag. Hur många klasser skulle vi ha och hur skulle de definieras?

Till att börja med sökte vi information om semantisk klassificering både på Internet och i det material vi fått av Martin Blom och vår handledare. Inget av detta material gav oss någon hjälp med att lösa klassificeringsproblemet.

Under tiden vi bearbetade materialet växte det fram en viss känsla för vilka typer av fel som fanns och hur en klassificering skulle kunna se ut. Känslan var fortfarande ganska vag men gav en grund att stå på i vår fortsatta diskussion med uppdragsgivaren.

En preliminär klassificering diskuterades fram:

1. Fel som inte kan spåras till brister i den semantiska beskrivningen.
3. Fel som kanske (osäkert) kan spåras till den semantiska beskrivningen.
5. Fel som kan spåras till brister i eller avsaknad av den semantiska beskrivningen.

1, 3 och 5 valdes för att ge utrymme för en finare gradering om det skulle behövas senare.

Vi fick också en klass som vi inte kunde dra några slutsatser om. Detta beroende på att det antingen helt saknades material eller att man utifrån felrapporten inte alls kunde se vad som orsakade felet och att en historikfil saknades för vidare bearbetning.

Dokumentationen för Ericsson SS7 är omfattande och verkar väl genomarbetad, vilket vi trodde skulle underlätta vårt arbete med klassificeringen. Under det fortsatta arbetet visade sig det att så inte var fallet. Att dokumentationen var omfattande ledde till svårigheter att härleda felorsaken till brister i något specifikt dokument.

Vid ett möte med vår uppdragsgivare framförde han ett önskemål om att vi skulle titta på om det fanns några typer av fel som var rikt förekommande i de olika grupperna, exempelvis slarv eller hårdvaruproblem. Detta för att ge en klarare bild av vilka fel som ingår i de olika klasserna, men även för att det kunde finnas ett intresse rent statistiskt att se om vissa typer av fel hade en högre frekvens än andra och om man på ett enkelt sätt skulle kunna avhjälpa dem.

När felrapporterna var bearbetade bestämdes att den preliminära klassindelningen skulle bibehållas, då våra slutsatser av felorsakerna inte kan garanteras utan är tolkningar utifrån vår kunskap och materialets beskaffenhet. En ytterligare uppdelning skulle inte tillföra slutresultatet någonting positivt. Vi benämnde klasserna i fortsättningen 0, 1, 2 och 3 för att underlätta den statistiska sammanställningen.

5.5 Klassificering

Kapitlet tar upp de urvalskriterier som använts vid klassificeringen samt tillämpningen av dem. Kriterierna beskrivs först kortfattat i 5.5.1. Detta är en principbeskrivning som gäller för både SS7 och WBS. I kapitel 5.5.2 respektive kapitel 6.6.2 utvecklas dessa beskrivningar med de konkreta feltyper som hittades i respektive projekt.

5.5.1 Kriterier

Klass 0:

Till klass 0 hör fel med bristande beskrivning så att de inte går att klassificera. Detta rör följande feltyper:

- Fel med ett inregistrerat TR-nummer, men där både felrapport och historikfil saknas.
- Fel med bristfällig beskrivning om felorsaker och åtgärder, så att det inte går att utläsa från felrapporten vad som orsakat felet.

Klass 1:

Till klass 1 hör fel som inte kan härledas till brister i specifikationen. Detta rör följande feltyper:

- Fel som rör egenskaper som ligger på en så detaljerad nivå att de inte finns specificerade.
- Fel som bryter mot tydligt uppsatta generella regler och riktlinjer för programmeringsarbetet.
- Fel som bryter mot tydligt uppsatta kravspecifikationer.
- Rena syntaxfel.
- Felaktiga felanmälningar som rör önskemål om ändringar och inte fel i programvaran.

Klass 2:

Till klass 2 hör fel som inte kunnat härledas till någon av de andra klasserna. Det gäller klassificerbara fel som vi inte har hunnit placera i rätt klass än. Detta rör följande feltyper:

- Fel som kan beror på feltolkningar av de specifikationer som skall följas.
- Fel som kan beror på att specifikationerna inte lästs tillräckligt noggrant eller att inte lästs alls.
- Fel som kan beror på bristande specifikationer.
- Felen ligger på en detaljnivå som skall vara beskriven i en specifikation.

Klass 3:

Till klass 3 hör fel som tydligt kan härledas till brister i specifikationen. Detta rör följande feltyper:

- Specifikationerna saknar eller har otydliga direktiv för funktionens eller modulens funktionalitet.
- Standarder som ligger till grund för programvaran är tvetydiga.
- Avsaknad av specifikation för hårdvara.

5.5.2 Tillämpning

Klass 1 innehåller fel som kan spåras till andra felorsaker än brister i specifikationerna. Här finns till exempel slarvfel, logiska fel och syntaktiska fel. Till slarvfel hör fel som kanske beror på trötthet hos programmeraren, som att han/hon bryter mot grundläggande programmeringsregler. Dessa regler finns mycket väl dokumenterade, till exempel att prefix skall användas på alla funktionsnamn och konstanter. Hit hör även de fel där programmeraren implementerat fel trots tydligt angivna specifikationer. De detaljer som inte är specificerade är till exempel längden på strängar och adresseringen av pekare. Med logiska fel menar vi att programmeraren förstått problemet men löst det på ett felaktigt sätt. Slutligen har vi sett ett antal fel där koden enligt felrapporten inte ens går att kompilera, vilket beror på felaktig syntax. I denna grupp hamnar även rapporter som inte härrör sig från fel utan är önskemål om förändring av produkten, såkallade CR, men där kunden har skrivit en felrapport, TR.

I klass 2 finns de fel som är av semantisk karaktär, men där vi inte kan bedöma om felet beror på specifikationerna eller på människorna som har läst dem. I denna grupp finns bland annat fel som kan härledas till parameteröverföring och omhändertagande av returvärden till/från funktioner och moduler. Antalet parametrar eller deras värde kan vara felaktigt. Här finns även fel som beror på brister i eller avsaknad av felkontroll på inkommande värden. Dessa detaljer skall normalt finnas klart definierade i specifikationen. Vi har dock inte hunnit göra en närmare bestämning av felorsaken i varje enskilt fall beroende på att den befintliga dokumentationen är så omfattande och tidsramen för begränsad. Klass 2 skulle inte ha existerat då dessa fel antingen hör till klass 1 eller klass 3.

I klass 3 hamnar de fel som med stor säkerhet beror på brister i specifikationen, dvs. felaktig specifikation eller avsaknad av information i densamma. Till denna grupp hör även de fel som kan härledas till brister i kommunikationen och specifikationen mellan hårdvaruleverantören

och Ericsson Infotech när det gäller uppdatering av hårdvara och/eller drivrutiner. Vid uppgradering av hårdvara så meddelas inte alltid detta. Här hamnar också de fel som beror på tvetydiga standarder. Dessa specifikationer ligger utanför Ericsson Infotechs kontroll. I standarderna så kan samma sak beskrivas motstridigt på olika ställen i dokumentet. Andra saker har inte definierats mer än att det står ”kommer att definieras senare”.

5.6 Resultat

Resultaten blev:

Klass 0 = 16,5% ± 2,7%

Klass 1 = 39,5% ± 3,8%

Klass 2 = 32,4% ± 3,7%

Klass 3 = 11,8% ± 2,5%

Vi har även försökt specificera vissa feltyper i varje klass, och fått följande resultat. I klass 0 beror 57 % på att material saknas. I klass 1 finns det 34 % slarvfel, 15 % logiska fel och 7 % som beror på att en kund skrev en felrapport när de hade önskemål om ändringar i programvaran. Klass 2 innehåller 18 % fel som kan härledas till parameteröverföring. I klass 3 beror 75 % av felen på problem med hårdvara och 15 % är relaterade till olika standarder. I varje klass finns det ett antal fel som inte tillhör någon gruppering, de var för få för att skapa en egen grupp.

Dessa siffror visar att den största gruppen av felorsaker är klass 1 och i denna klass återfinns även den största enskilda felgruppen (slarv). Av dessa siffror utläses att klass 2 med sina 33 % är den näst största gruppen.

De tre största enskilda felorsakerna i hela populationen är slarv 14 %, hårdvaruproblem 9 % och parameteröverföring 6 %.

5.7 Tolkning av resultat och förslag till åtgärder

Vår tolkning av dessa siffror är att man för att få ned felfrekvensen måste titta på alla de 4 olika klasserna. Denna bedömning är gjord utifrån att de största enskilda felgrupperna kommer ur olika klasser

Den ur semantisk synvinkel största gruppen är grupp 2. Vår bedömning är att oberoende av hur felen har uppstått i denna grupp skulle huvuddelen av dem kanske kunna avhjälpas av en enklare och klarare dokumentation. Användandet av klart utskrivna för- och eftervillkor skulle kanske kunna reducera antalet fel. I vår bedömning av klass 2 finns en viss osäkerhet då felen är semantiska men vi inte har kunnat peka på brister i specifikationen i det enskilda fallet. Generellt sett är dokumentationen för Ericsson SS7 mycket välskriven, men den är omfattande och innehåller ibland hänvisningar till andra dokument. Detta kan vara en orsak till feluppkomsten. Vi har ingen bra lösning på problemet.

Andelen av fel relaterade till hårdvaran (ofta nätverkskort) i klass 3 är 75 %, denna grupp beror på brister i kommunikationen mellan programvaruföretaget och leverantören av hårdvaran. Ett förslag är att utarbeta nya och bättre rutiner för information om när nya/uppdaterade drivrutiner släpps. Det måste ligga i båda företagens intresse att samarbetet löper så friktionsfritt som möjligt.

Klass 1, den absolut största gruppen, är svår att bedöma. Inom denna grupp återfinns ett stort antal fel som vi bedömt som slarv. Vid mötet med Mikael Lindh så berättade han att en del uppdatering hade skett utanför Ericsson Infotech av konsulter. Delar av det arbete som dessa konsulter lämnade ifrån sig var av undermålig eller mycket undermålig kvalitet. Dessa fel har i många fall klassats som slarv av programmeraren. Andra orsaker till att slarvfel kan uppstå är att programmeraren jobbar under tidspress. Hur dessa felorsaker ska kunna avhjälpas är svårt att säga men en lösning är att inte pressa tidsschemat maximalt. Merkostnaden för detta kommer troligtvis att löna sig i slutändan då det kostar mer att ta hand om felen.

Klass 0 innehåller en stor grupp där material saknas, detta beror troligen på produktens storlek och antalet personer engagerade i det. För en del felrapporter finns TR-nummer och rubrik men ingen rapport. Uppfattningen vi fått är att alla inte är lika intresserade av att skriva i historiken för modulen. Det verkar som om denna del inte alltid tas på allvar.

Generellt gäller strikta regler för hur dokumentationen för Ericsson SS7 skall vara utformad. Reglerna verkar följas exemplariskt med en liten reservation. Historikfilerna är av en mycket varierande kvalitet. Här tycker vi att det finns en hel del att göra. Historikfilerna bör innehålla en utförligare beskrivning av felorsaken och en analys om dess uppkomst. Det här påverkar

kanske inte programkvaliten direkt, men vid spårning av fel och vid den här typen av undersökningar skulle arbetet underlättas betydligt. Tidsåtgången för att skriva utförligare i historikfilen borde vara marginell i förhållande till tiden det tar att rätta felet.

Under mötet med Mikael Lind framkom det att ingen kod slängdes, utan uppdateringar och ändringar gjordes i den gamla koden. Detta arbetsätt måste ha till följd att koden för de äldsta modulerna måste vara enorm och komplex. Antagligen försvårar det arbetet med att skapa en ny funktionalitet hos produkten.

6 WBS

Detta kapitel presenterar WBS och innehåller en presentation av resurspersonerna. Materialet som använts är beskrivet och även arbetsgången. Kapitlet behandlar också en diskussion om klassificering och hur den slutgiltiga klassificeringen blev. Den statistik som blev resultatet av undersökningen redovisas och en kort analys görs.

Detta kapitel innehåller även en tolkning av resultatet och förslag på vad som skulle kunna göras för att undvika en del av felen.

6.1 Presentation av produkten

WBS togs fram och utarbetades på bara några få månader under våren 1999. Projektet var ganska litet och det var cirka 10 personer som utvecklade produkten. Produkten är framtagen på begäran av ett Ericsson Radio System i Linköping som direkt svar på en beställning av ett norskt företag, och den ingår i en större produkt USC. WBS är ett paket med bastjänster som utvecklades för USC-plattformen

När WBS togs fram så fanns inte några fungerande WAP-telefoner eller -terminaler utan allt är testat mot simulatorer i PC-miljö. WBS skulle anpassas så att den var kompatibel även med konkurrenternas telefoner. Produkten utvecklades parallellt för två olika språk, HTML och WML, för att fungera både på vanliga webb-läsare och på WAP-telefoner.

Livslängden för denna produkt beräknades till att vara cirka två år. Några uppgraderingar var inte planerade, istället skulle en helt ny produkt utvecklas.

6.2 Resurspersoner

Våra kontakter på Ericsson Infotech/L var Helena Lindskog och Niklas Sandberg. Helena Lindskog arbetade som projektledare för Wicket, som var arbetsnamnet på det projekt där WBS utvecklades. Hon deltog även själv i implementationen. Niklas Sandberg deltog i implementationen och var testledare. Han tog emot alla felrapporter och fördelade dem bland sina medarbetare. När felen var åtgärdade låg det på hans ansvar att korrigeringarna blev testade och avslutade.

6.3 Material

Från WBS har vi fått 102 stycken felrapporter (bilaga A.4). Rapporterna innehåller en kort summering av respektive fel med avsändare, prioritering och en lite längre beskrivning av felet. Vissa av rapporterna innehåller även ett lösningsförslag. Vi fick en beskrivning på hur användargränssnittet skulle vara utformat i dokumentet "The Wicket-Guideline". Det består av en beskrivning hur det grafiska gränssnittet skall se ut. Ett flödesschema beskriver hur de olika knapparna i skärmbilderna skall länkas till andra skärmbilder. Det beskriver också generellt hur felmeddelanden och knappar skall utformas. Ett annat dokument var en systemdesign, SDS, som beskriver hur systemet är uppbyggt.

6.4 Tillvägagångssätt

Under denna rubrik presenterar vi de olika faser som undersökningen genomgått. Här beskrivs även det grovarbete som utförts. Vi har valt att dela upp detta kapitel i flera olika underkapitel för att underlätta läsningen av rapporten. Detta trots att arbetet med de olika delarna naturligtvis inte var separerade utan skedde växelvis.

6.4.1 Förberedelser

Vi hade ett inledande informationsmöte på Ericsson Infotech med Helena Lindskog och Niklas Sandberg där vi fick grundläggande kunskap om hur projektet Wicket hade genomförts. Diskussioner fördes om vilket material vi kunde ha behov av och få tillgång till. Det bestämdes att vi skulle få tillgång till alla felrapporter för hela projektet.. Vi fick en delleverans på 21 felrapporter, som vi lästa igenom och konstaterade att det här inte skulle bli lätt. Vi bestämde att försöka ändå. Eftersom uppdragsgivaren hade varit delaktig i Wicket-projektet, hade vi en del diskussioner rörande problemet med att kunna klassificera dessa felrapporter, för det fanns brister i det material som fanns att tillgå.

6.4.2 Bearbetning

Efter att på var sitt håll suttit och granskat cirka 30 % av rapporterna uppskattade vi att en ganska stor andel av felen kunde härledas till användargränssnittet, från knappar som inte fungerade, till felaktiga meddelanden. Utifrån denna iakttagelse beslutade vi oss för att dela upp felrapporterna mellan oss och bearbeta dem individuellt, då cirka hälften av felen inte krävde någon diskussion. Denna iakttagelse visade sig senare inte stämma med verkligheten, då vi hade överskattat andelen av felen i användargränssnittet.

Nästa steg blev att gå igenom resterande felrapporter för gemensamma diskussioner och klassificering. När vi var färdiga med det, upptäckte vi att det fanns ett trettiotal fel där vi inte kunde dra någon slutsats. De vidarebefordrades till vår handledare som hade erbjudit sig att hjälpa oss med problemet, då han hade deltagit i utvecklingen av produkten. Han gjorde sin tolkning av felrapporten och beskrev för oss vad felen handlade om. Utifrån den tolkningen så gjorde vi vår klassificering.

Vi beslutade oss för att utföra en snabbgranskning av alla felrapporter och att under detta arbete notera vad felet berodde på, t.ex. slarv eller fellänkade knappar. Det fanns ett önskemål från vår handledare om den informationen och klassindelningen fick ytterligare en dimension. Man får ett mer konkret resultat och kan utläsa vilka felorsaker de olika klasserna innehåller.

6.4.3 Statistisk bearbetning

Vi slumpade ordningsföljden för felrapporterna på samma sätt som för Ericsson SS7, ifall vi inte skulle hinna med att bearbeta allihop. Vi hann dock med att gå igenom alla felrapporter, så någon beräkning av felmarginaler behövde ej genomföras. Det var bara att räkna ut den procentuella fördelningen mellan klasserna och andelen av de specifika felen som togs fram för en del av klasserna.

6.4.4 Möten

De möten som berörde WBS hade vi huvudsakligen med vår handledare. Han hade ju deltagit i projektet och kunde lösa de flesta problem vi stötte på. Frågorna vi hade togs upp vid våra ordinarie handledarmöten och var oftast av enklare art.

6.4.5 Diskussion om klassificering

Den första diskussionen om klassificering var grundläggande och gällde båda produkterna, se avsnitt 5.4.5. Under arbetets gång hittade vi vissa feltyper som var överrepresenterade.

Specifikt för WBS var att en grupp av fel blev avvisade på grund av att köparen och projektgruppen hade olika åsikter om vad som skulle utföras och i vissa fall hur detta skulle göras. Avsaknaden av specifikationer för produkten gjorde att det blev mycket svårt att säga hur felen uppstått.

6.5 Klassificering

Kapitlet tar upp de urvalskriterier som använts vid klassificeringen samt tillämpningen av dem.

6.5.1 Kriterier

Vi beslutade oss för att använda samma klassindelning, 0, 1, 2 och 3 med samma kriterier som vi använde för Ericsson SS7.

6.5.2 Tillämpning

I klass 1 återfinns slarvfel, meddelanden som var felaktiga eller saknades, och knappar som inte fungerade. De felaktiga meddelanden var oftast felmeddelanden som var missvisande eller dåligt formulerade. I de flesta fall där knapparna inte fungerade så berodde det på tidsbrist, man hann helt enkelt inte att göra klart innan det var dags för test. Några av felen kunde härledas till inkonsekvens när det gällde utseendet i det grafiska gränssnittet t.ex. texten på OK-knappen skrevs inte på samma vis i de olika skärmfönstren. Det skulle skrivas OK men skrevs även som Ok och ok

Klass 2 innehåller de fel som inte kan härledas till brister i specifikationen. Programmeraren har alltså av någon orsak inte kunnat tolka de dokument som finns. I några få fall är knappar fellänkade. Vi har även en ganska stor grupp som har blivit avvisade på grund av olika uppfattningar mellan Ericsson Radio System i Linköping och Ericsson Infotech. Dessa fel har hamnat i denna klass då vi inte har sett någon kravspecifikation för produkten.

I klass 3 finns de fel som med säkerhet beror på brister i, felaktig eller avsaknad av information i specifikationen. Till denna grupp hör de fel som kan härledas till brister i kommunikationen mellan Ericsson Radio System i Linköping och Ericsson Infotech när det gäller uppdatering av programvara. Dessa problem låg helt utanför projektgruppens kontroll.

6.6 Resultat

Klass 0 = 10 %

Klass 1 = 65 %

Klass 2 = 20 %

Klass 3 = 5 %

Vi har även försökt specificera vissa feltyper i varje klass, och fått följande resultat.

I klass 1 finns det 20 % fel som berodde på felstavade eller felformulerade meddelanden, 14 % där felaktiga felmeddelanden skrevs ut och 11 % där felmeddelande saknades. Andelen slarvfel var 9 % och 6 % orsakades av att knapparna ej fungerade som de skulle. Det hände ingenting när man tryckte på dem eller så kom fel fönster upp.

Klass 2 innehåller 2 feltyper som vi kunde urskilja. 62 % berodde på olika uppfattning om vad programmet skulle utföra. Köparen hade en uppfattning och Ericsson Infotech hade en annan, och någon beskrivning fanns inte med i kravspecifikationen. Resultatet blev att de avvisades med hänvisning till kravspecifikation. Andelen som orsakades av knappar som reagerade på fel sätt var 19 %. Knapparna var rätt länkade men något gick fel i den funktionen som skulle utföra nästa tjänst.

6.7 Tolkning av resultat och förslag till åtgärder

Klass 1 är den klart största gruppen, den innehåller en stor andel fel som kan härledas till felaktiga meddelanden, och knappar som inte fungerande. Hälften av felen i klassen hör hit. Det antydde att en hel del av de här felen tillkommit på grund av mycket stor tidspress i slutskedet av projektet. Det nämndes 80-timmars arbetsveckor. Det verkar som en mycket rimlig förklaring. Det här problemet är svårlöst, vi har i alla fall ingen lösning.

Felen som avvisades, drygt hälften i klass 2 kan i vissa fall ha orsakats av brister i kravspecifikationen. En annan orsak till den feltypen, var att köparen förändrade sina krav i efterhand. Enligt Niklas Sandberg ledde detta i en del fall till långa diskussioner. De här felen tilldelades klass 2, eftersom vi i det enskilda fallet inte kunde bedöma, om det med stor säkerhet berodde på brister i kravspecifikationen. En lösning på problemet skulle kunna vara en tydligare kravspecifikation. Problemet med knapparna som var fellänkade är det svårt att

hitta orsaken till. Vi har inte kunnat läsa ut ur dokumentationen hur de skall länkas. Beror det på oss eller på specifikationen? Lösningen kan vara en tydligare specifikation.

Andelen för klass 3 var mycket liten. Vi kunde inte urskilja några feltyper i denna grupp. Vår bedömning är att oberoende av hur felet har uppstått så skulle huvuddelen av dem kunnat avhjälpas med en bättre dokumentation.

Dokumentationen för WBS är av ganska bristfällig natur. WBS hade mycket fria händer vid utvecklingsarbetet. Antalet personer som arbetade med produkten var liten vilket gjorde att man kunde lösa många problem muntligt under utvecklingsarbetet.

7 Jämförelser av resultat

I detta kapitel belyses de likheter och olikheter som kommit fram om de båda produkterna.

Dessa båda produkter är egentligen 2 ytterligheter av programvaror. Ericsson SS7 är en produkt som är framtagen för att användas under en lång tidsperiod medan WBS är dess raka motsats. Ericsson SS7 är utvecklad under lång tid vilket gör att de fel vi har hanterat är inte av typen barnsjukdomar utan kommer sig av ändringar och uppdateringar. WBS var en helt ny produkt som skapades från grunden och där det inte fanns någon liknade produkt på marknaden.

För klass 0 så beror antagligen den större procentandelen för Ericsson SS7 på att material saknades. Räknar man bort den typen så återstår 7 % oklassificerbara. Orsaken till att Ericsson SS7 har en lägre representation efter denna korrigering kan vara att vi utifrån två dokument bedömt felorsaken.

I klass 1 har WBS nästan dubbelt så stor representation som Ericsson SS7. Förklaringen ligger i att WBS utvecklades under mycket stor tidspress, minst hälften av felet känns stressrelaterade. Generellt för klass 1 är att en stor del av felet relateras till hög stressfaktor.

Ericsson SS7 har en klart större andel fel i klass 2 än WBS. Anledningen till detta borde bero på att Ericsson SS7 är ett stort och komplext system med många personer involverade. En

annan orsak kan vara att man anlitar konsulter som kan ha svårt att få en överblick över alla de dokument som krävs. Det krävs en stor grundkunskap i Ericsson SS7:s programmeringsteknik, detta bör läsas in innan arbetet påbörjas.

En slutsats varför Ericsson SS7 har dubbelt så stor representation i klass3 kan bero på att vi haft utförligare felrapporter för Ericsson SS7.

8 Erfarenheter

Detta kapitel innehåller en reflektion om undersökningen och upplägget på denna. Här finns även ett förslag till en annorlunda arbetsgång för att få en säkrare bedömning av felorsaken. Ett delkapitel tar upp de olika problem vi stött på under arbetet.

8.1 Problem

Ett genomgående problem för båda produkterna var att felrapporterna överhuvudtaget inte beskrev vad felet kunde bero på. Vårt problem utifrån beskrivningen i historikfilen var att hitta den funktion som var orsaken till felets uppkomst. Resultatet blir att det är svårt att dra slutsatser om specifika fel. Istället får man göra en allmän bedömning av kvaliteten på dokumenten som helhet. Vi har slumpmässigt läst igenom några olika specifikationer.

Ett annat problem var att vi ville undersöka så många felrapporter som möjligt för att få ett bra statistiskt underlag som möjligt. Eftersom tiden var begränsad så kunde vi inte göra en djupare undersökning av varje rapporterat fel.

Rapportskrivningen gav oss stora problem i slutet, själva undersökningen tog för mycket tid och vi avbröt inte arbetet i tid. En orsak till tidsbristen var att vi inte skrev tillräckligt mycket på rapporten parallellt med undersökningen.

8.2 Förslag till annorlunda arbetssätt

En av de erfarenheter vi fått är att upplägget av undersökningen skulle varit annorlunda. Vårt misstag var att undersökningen blev kvantitativ istället för kvalitativ. Vi trodde att undersökningen skulle få högre kvalitet än den fick. För att få kvalitet i undersökningen bör

produkten mer likna Ericsson SS7 än WBS, då det är svårt att få ett kvalitativt resultat med de brister i specifikationerna som vi upplevde att det fanns i Wicketprojektet. Här kommer några tankar om vad vi skulle göra annorlunda om vi skulle genomföra undersökningen för Ericsson SS7 idag.

Man bör börja med att göra en grundligare undersökning om hur arbetet går till i verkligheten, innan man påbörjar bearbetningen av felrapporter. Skapa kontakter med personer som arbetar aktivt med produkten, för att kunna göra studiebesök i ”verkstan” med jämna mellanrum. Det ger en inblick i det praktiska arbetet och en möjlighet att diskutera arbetsgången och felorsaker vid olika stadier av undersökningen. Det ger även en möjlighet att få se olika moment i produktionen t.ex. skapandet av nya moduler, uppgraderingar och felrättning. Diskutera de flesta felen med t.ex. Mikael Lindh, även om man tror sig se felorsaken. Vi blev mycket förvånade vilken kunskap han hade när det gällde de olika felen. Han hade bra förklaringar till cirka nittio procent av de felen vi diskuterade med honom. Diskutera tveksamma felrapporter med olika personer som arbetar med produkten. För att kunna genomföra ovanstående måste mängden felrapporter minskas markant, då man gör en betydligt grundligare undersökning av varje rapport. Det här bör resultera i kvalitet istället för kvantitet.

Kvaliteten hade ökat i vår undersökning om historikfilerna hade innehållit en utförligare beskrivning av felorsaken och en analys om dess uppkomst. En utvärdering om modulen följer de specifikationer som finns, om så inte är fallet är den otydlig eller saknas information. Dessa åtgärder skulle underlätta liknande undersökningar.

Referenser

- [1] Blom Martin. *Semantic Integrity in Program Development*, Department of Computer Science at Karlstad University, 1999.
- [2] Blom Martin, Nordby J. Eivind, Brunström Anna. *Method Description for Semla. A Software Design Method With a Focus on Semantics*, Department of Computer Science at Karlstad University, Ericsson Infotech AB, 2000.
- [3] Christmansson Jörgen. *An Exploration of Models for Software Faults and Errors: a Journey through Field data and Injection Experiments*. Chalmers tekniska högskola, 1998.
- [4] Meyer Bertrand. *Design by Contract & the Component Revolution, The New Breakthrough in Software Tecnology*, a one –by one day course
- [5] Vännman Kerstin. *Matematisk statistik*. Studentlitteratur, 1990
- [6] Microsoft Encarta 96 Encyclopedia
- [7] NATIONALENCYKLOPEDIN band 16, Bra Böcker Höganäs 1995

A Bilaga

A.1 Mall för Truble Report för SS7

TR Number:
TR ref.number :

Date:

T R O U B L E R E P O R T

for

P O R T A B L E S S 7

Title:

REPORTED BY
=====

Company:
Customer Id:
Reference:
Address:

Date:

Telephone:
Fax:

Comments to CSC:

LICENCE/PRODUCT INFORMATION
=====

Delivered product ID:
Version/Revision:

Module/Document afflicted
with error:
Standard (ITU/ANSI/TTC):
Version/Revision:

Network Interface:
Host processor/Machine Type:
Operating system, version:
Additional hardware:
Firmware level:

Failure class (A, B or C):
A - System can't be used at all
B - Some functions can't be used
C - Minor effect on usability

TROUBLE DESCRIPTION
=====

Operation Impact:
Is error repeatable (Y/N):
Description:

A.2 Mall för History file för SS7

```
*****
Title:
Ref:
Name:
Date:
Based on version:

Desc.:

Action:

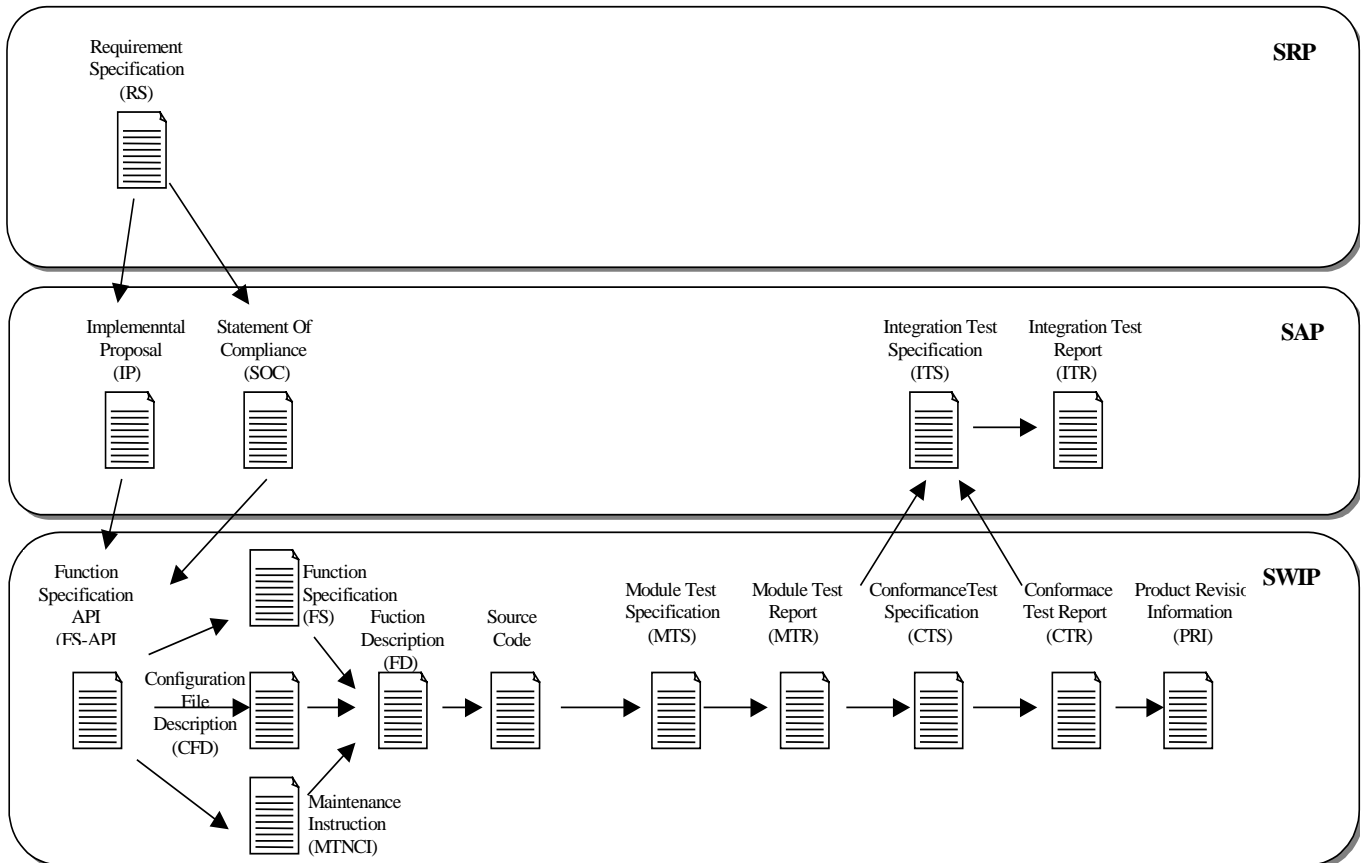
Source of error:

Modified files:

Other affected modules:
Tested and verified
-----
Name:
Date:
Environment:

*****
```


A.3 Schema för arbetsgången för SS7



A.4 Mall för TRI för WBS

ID:	Priority:
Project:	Status:
Subproject:	
Module:	Repeatability:
Keywords:	

Reported by:	Problem owner:
Created date:	

Summary:

Descriptions:

Explanation:

Suggestion:

Action:

Corrected:

Verified:

Closed: