



Department of Computer Science

---

**Per Hurtig**

**Stefan Lindberg & Fredrik Strandberg**

**Unit testing methodology**

# 1 Övergripande utvärdering

Helhetsintrycket av uppsatsen är mycket bra. Den är i det stora hela lättläst, väl strukturerad, och inte minst informativ och väldigt intressant. Man får verkligen intrycket av att författarna satt sig in i ämnet ordentligt.

Projektet som uppsatsen bygger på verkar, även det, vara bra genomfört eftersom uppdragsgivaren, Manufacturing-IT, tydligen kommer att använda sig av slutprodukten.

## 2 Detaljerad utvärdering

### 2.1 Titel

Generellt sett så är titeln på rapporten bra. Den antyder att rapporten kommer att beskriva en metodik för enhetstestning, vilket är ungefär vad den gör. En alternativ titel som även skulle antyda att rapporten beskriver utveckling och utvärdering av en metodik, vilket är precis vad som görs, kanske skulle vara lämplig. Några förslag på en sådan titel är:

- “Development and evaluation of a unit testing methodology”
- “Unit testing methods - an evaluation”

### 2.2 Disposition & Röd tråd

Upplägget på uppsatsen är bra. Den som i förväg inte är insatt i “Software engineering” och mjukvarutestning får i början av uppsatsen en snabb resumé över de delar som han/hon kan tänkas behöva för att tillgodogöra sig resten. Efter det är uppsatsen indelad på ett logiskt och bra sätt där författarna först beskriver vilka önskemål och krav som fanns gällande metodiken, och hur den utformades utifrån dessa. Efter detta så följer kapitel där metodiken testas (genom en fallstudie), och slutligen utvärderas. I den sista delen av uppsatsen sammanfattas projektet i sin helhet. Här beskriver författarna vilka problem de stött på under arbetets gång och vad i metodiken som kan förbättras i ett senare skede.

Det finns dock vissa saker i uppsatsen som kan förbättras, främst saker som rör själva strukturen på uppsatsen:

**Fotnoter** Fotnotsnumreringen är inte konsekvent. Varje kapitel skall ha sina fotnoter numrerade, i stigande ordning, från 1. Istället så är numreringen av fotnoter aningen godtycklig genom uppsatsen. I vissa kapitel så är alla fotnoter numrerade med en etta, och i andra så är det blandat (t.ex. två ettor och en tvåa).

**Sidhuvud** En annan sak som skulle kunna underlätta läsningen av uppsatsen är sidhuvuden som visar i vilket kapitel man befinner sig.

**Kapiteldjup** Det som kanske är mest angeläget att åtgärda är dock “kapiteldjupet” i uppsatsen. I vissa delar av uppsatsen (främst i Kapitel 3 och 5) så är det för många nivåer av underkapitel för att det skall vara lättläst. Dessa underkapitel innehåller dessutom väldigt mycket text, vilket gör att man som läsare kan tappa tråden.

Sammanfattningsvis så kan man säga att grundstrukturen är bra, och att det finns röd tråd som löper genom hela uppsatsen. Den röda tråden förstärks ytterligare genom att alla kapitel är utrustade med bra introduktioner och sammanfattningar. Tråden skulle dock kunna bli ännu litet "rödare" om det s.k. "kapiteldjupet" minskade.

## 2.3 Vetenskaplig metod

Den vetenskapliga metod som presenterades i uppsatsen följer nedanstående bild.



Efter att författarna samlat in de önskemål och krav som uppdragsgivaren hade gällande metodiken, så konstruerades en prototyp baserat på dessa samt på vedertagna testmetoder. Dessa, vedertagna, testmetoder identifierades genom en omfattande litteraturstudie, och anpassades efter de krav som uppdragsgivaren ställde på metodiken. Resultatet av detta resulterade i en prototyp som utvärderades genom en fallstudie.

Det finns ingenting, vad gäller den vetenskapliga metoden, att anmärka på.

## 2.4 Argumentation och slutsatser

Författarna argumenterar inte för några tvivelaktiga metoder. Inte heller drar de några överdrivna, eller osannolika, slutsatser baserat på det material de presenterar i uppsatsen. Därför finns det inget att anmärka på vad gäller argumentationen och slutsatserna i rapporten.

## 2.5 Abstract

Abstract'en innehåller de delar som man kan önska. En introduktion till ämnet, en kort sammanfattning av arbetet som utförts, samt en snabb översikt av resultatet. Finns inget speciellt att anmärka på.

## 2.6 Språkbruk

Språkbruket som används i uppsatsen påvisar inga brister. Det är varierat och inte särskilt svårt, vilket det lätt kan bli i tekniska dokument.

# 3 Utvärdering av de enskilda kapitlen

Detta avsnitt av oppositionsrapporten kommer att behandla de olika kapitlen i uppsatsen. Detta görs på två olika sätt; om det inte finns något speciellt att anmärka på så kommer jag presentera ett sammanfattande helhetsintryck av kapitlet, om det däremot finns anmärkningar eller saker som är oklara kommer jag att ge dem i listform.

## 3.1 Kapitel 1 - Introduction

Detta kapitel ger en bra introduktion till arbetet, och förser samtidigt läsaren med en bra sammanfattning, både vad gäller arbetet i stort och vad de olika delarna i uppsatsens består av.

## 3.2 Kapitel 2 - Background

I kapitlet där bakgrunden ges gör författarna ett bra jobb genom att snabbt leda läsaren från “Software Engineering” till enhetstestning, vilket uppsatsen huvudsakligen handlar om.

**Kapiteldjup** I början av kapitel 2, sidan 7, finns det fyra stycken underkapitel (2.2.2.1 → 2.2.2.4). Med tanke på hur lite text dessa underkapitel innehåller, samt hur stort “kapiteldjupet” de resulterar i så kanske det vore bättre ifall de var ordnade som en lista.

**Mjukvarutestning** På sidan 10 i uppsatsen finns ett delkapitel som heter “Unstructured testing process”. Detta kapitel borde kanske slås samman med Kapitel 2.4 - “Software Testing” eftersom det mesta som nämns i delkapitlet snarare hör till mjukvarutestning än till uppdragsgivarens specifika testprocess.

## 3.3 Kapitel 3 - Unit testing methodology development

Detta kapitel presenterar själva skapandet av prototypen.

**Kapiteldjup** I Kapitel 3 är problemen med kapiteldjupet som värst. I början av kapitlet är allt väldigt bra strukturerat, men under delkapitel 3.4.4 finns det för många underkapitel. Man kanske borde lyfta dem en nivå.

**Punktlista** I punktlistan, på sidan 22, skulle man kunna slå ihop punkt 1 & 3 eftersom de är så starkt knutna till varandra.

**Metoder medtagna i UTM** Avsnitten som beskriver och jämför olika testmetoder är aningen inkonsekventa. För vissa metoder, som “Equivalence partitioning and boundary value analysis”, så står det klart och tydligt att de skall upptas i metodiken. För andra metoder, såsom “Multiple condition testing”, nämns det inget om eventuellt inkluderande i metodiken.

**Dataflödestestning** På sidan 35 beskrivs “Data flow testing” som en path testing metod. Finns det några skäl till varför denna metod inte beskrivs i avsnittet om just path testing?

**Objekt instanser** När objekt-orienterad testning behandlas så används begreppen lite felaktigt. Det står t.ex. följande (sidan 37): “A state of a class...”. En klass kan dock inte ha ett tillstånd, det kan endast ett objekt ha. Detta fel förekommer även längre fram i kapitlet.

**Stubbar** I delkapitel 3.4.8 diskuteras “stubbar”. Det framgår dock inte när de skall användas. Skall de användas i de fall som punktlistan, på sidan 48, anger eller i andra fall också?

**Bild** Bilden är lite “märklig”.

## 3.4 Kapitel 4 - Case Study

**Stubbar** På sidan 68 står det att stubben `SerialIO` var omfattande, samt att stubben `LoggFasad`, och andra de andra stubbarna(?) bara fanns för att kunna kompilera koden. Vilka var de andra stubbarna? Det finns nämligen inga andra stubbar beskrivna (i t.ex. Figur 4.3).

**Förväntningar** En synpunkt på detta avsnitt är att den kanske borde komma direkt efter introduktionen i kapitlet. Detta för att ge en översiktlig bild över de förväntningar som ställdes på fallstudien innan själva detaljerna för den samma presenteras. Det skulle också eliminera ett litet problem; i avsnittet “Case study execution”, sidan 67, så nämns det att nyckelordet “friend” inte kan användas för att komma åt “gömd” information i ett objekt, i nästa delkapitel “Case study expectations” är sedan användandet av just nyckelordet “friend” ändå med som ett förväntat resultat.

### 3.5 Kapitel 5 - Results and evaluations

**Resultat** Likt kommentaren om fallstudien så kanske även detta kapitel börja med en översikt av resultaten, innan detaljerna går igenom. En rekommendation är därför att flytta delkapitel 5.4 så att det hamnar först i kapitlet. En annan rekommendation är att presentera testfalls resultaten och tidsåtgången tidigare i kapitlet. Som det är nu så kan en läsare av uppsatsen lätt missa resultaten eftersom de är “gömda” i delkapitlet som behandlar hur testresultaten skall presenteras. Om dessa omflyttningar görs så kan man lätt få en snabb överblick av resultaten och sedan bestämma sig ifall man vill tränga in djupare i detaljerna. Som det är nu så får man göra det omvända.

**Tidsåtgång** Enligt uppgifter ifrån detta kapitel, och delvis kapitel 6, så kan man se att det uppskattningsvis skulle ta 6v, för hela utvecklingsteamet, att testa hela det projekt som fallstudien testade en del av. Är det rimligt? Om inte:

- Går det, uppskattningsvis, fortare när man “lärt” sig metodiken?
- Går det fortare ifall man får bättre testverktyg (såsom automatiska flödesgrafsprogram)?
- Eller får man helt enkelt tumma på täckningen av testerna?

**Tabell 5.1** Tabell 5.1 behöver förklaras lite mer ingående. Kanske iallafall en referens till avsnittet som beskriver hur en “all-pair” matris ser ut.

**Inspektion** I uppsatsen framgår det att vanlig inspektion av koden uppdagade allra flest fel, även om inspektion inte var med i metodiken. Kommer inspektion att inkluderas i senare versioner av metodiken?

### 3.6 Kapitel 6 - Conclusions

Inga speciella anmärkningar.

### 3.7 Appendix

Inga speciella anmärkningar.

## 4 Små korrigeringar

Sida (nr)	Föreslagen ändring
v	...in Västerås, Sweden manufactures... ⇒ ...in Västerås, Sweden, manufactures...
1	...in Västerås, Sweden is a part of... ⇒ ...in Västerås, Sweden, is a part of...
7	This definition from the IEEE is similar... ⇒ This definition, from IEEE, is similar
10	...lower-level areas.” [21] Another... ⇒ ...lower-level areas” [21]. Another...
11	In Section 2.6.2 about existing... ⇒ In Section 2.6.2, about existing...
14	Referensen till EP-310 ([59]) borde vara i den första meningen, istället för den tredje.
14	Felaktig referens. Istället för referens till Appendix C, så refereras det till Appendix A.
21	...in a controlled manner.” [45] All methods... ⇒ ...in a controlled manner” [45]. All methods...
28 & 29	Liknande referering (första styckena).
43	(described in section 3.4.5.4) ⇒ (described in Section 3.4.5.4)
51	I bildtexten så är bilden refererad till [33], i innehållsförteckningen till [34]
57	and ahamad [3] ⇒ and Ahamad [3]
61	...case study was set up and how the case study... ⇒ ...case study was set up and how it was...
78	Är helt blank.
80	I sista stycket nämns Clover.NET med en referens ([14]). Eftersom ni redan beskrivit verktyget kanske referensen skulle gå till avsnittet i den egna rapporten istället.
85 & 86	Footnot 1 & 2 är inte korrekt formaterade.
88	I kapitel 4 stavas verktyget dotUnit med litet “d”, i kapitel 5 med stort. Vilket stämmer?
98	Första stycket innehåller fyra stycken “solution(s)”, där innebörden skiftar mellan några av dem. Vore bra ifall några “solution(s)” byttes ut.
120	Bilden kunde göras större, och stavfelskorrigeringen (i MS Word) borde slås av.