

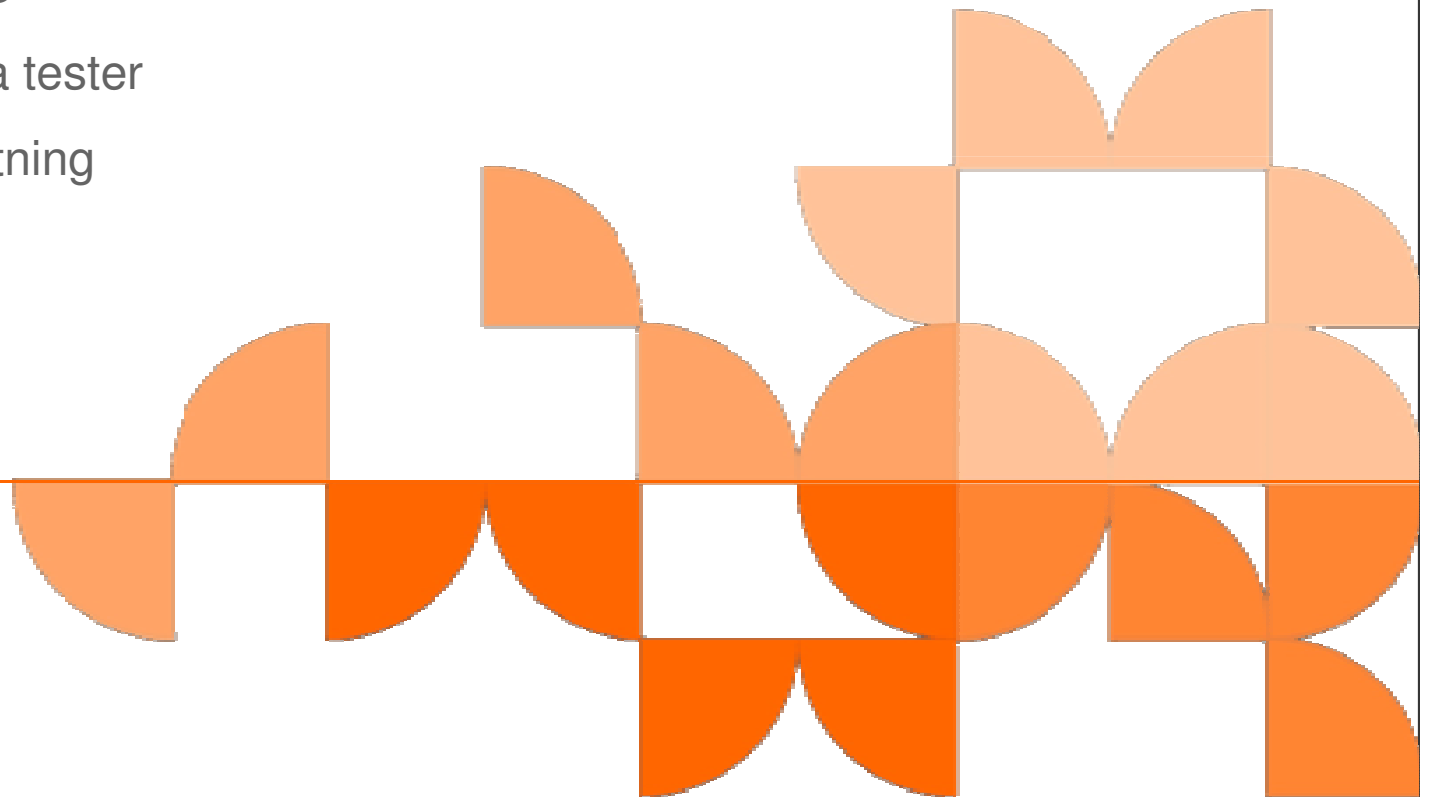
Testautomation

Användning av testautomation inom Extendas utvecklingsorganisation



Agenda

- **Presentation av Extenda**
- Vad är en POS?
- Test av POS
- Automatiska tester
- Sammanfattning



Kort historik

1982

- Extenda startar – 2 anställda

1984

- 10- årigt samarbete med Nokia/ICL/Fujitsu inleds

1994

- Extenda Retail ser dagens ljus
Windows 95

1995

- 100 butiker 200 kassor

2008



Idag


- 170 anställda
- 5 000 butiker - > 5 500
- 25 000 kassor - > 30 000
- Kunder i 25 länder (USA,Canada,Europa)
- 100 personer + tredjepart utvecklare
- Delphi, .NET, Java
- Windows (NT,2000,XP,WePOS)
- Unix / Linux

Kundreferenser - Butikssystem



Vår vardag

 finns idag i 24 länder med 1345 butiker.

 Hemköp, Willys och Willys hemma, Dagab samt Axfood Närlivs. Därutöver samverkar Axfood med över 500 handlarägda butiker.

 2 300 egna och handlarägda butiker i Sverige, Norge och Baltikum.

 25 000 aktiva rabatter.

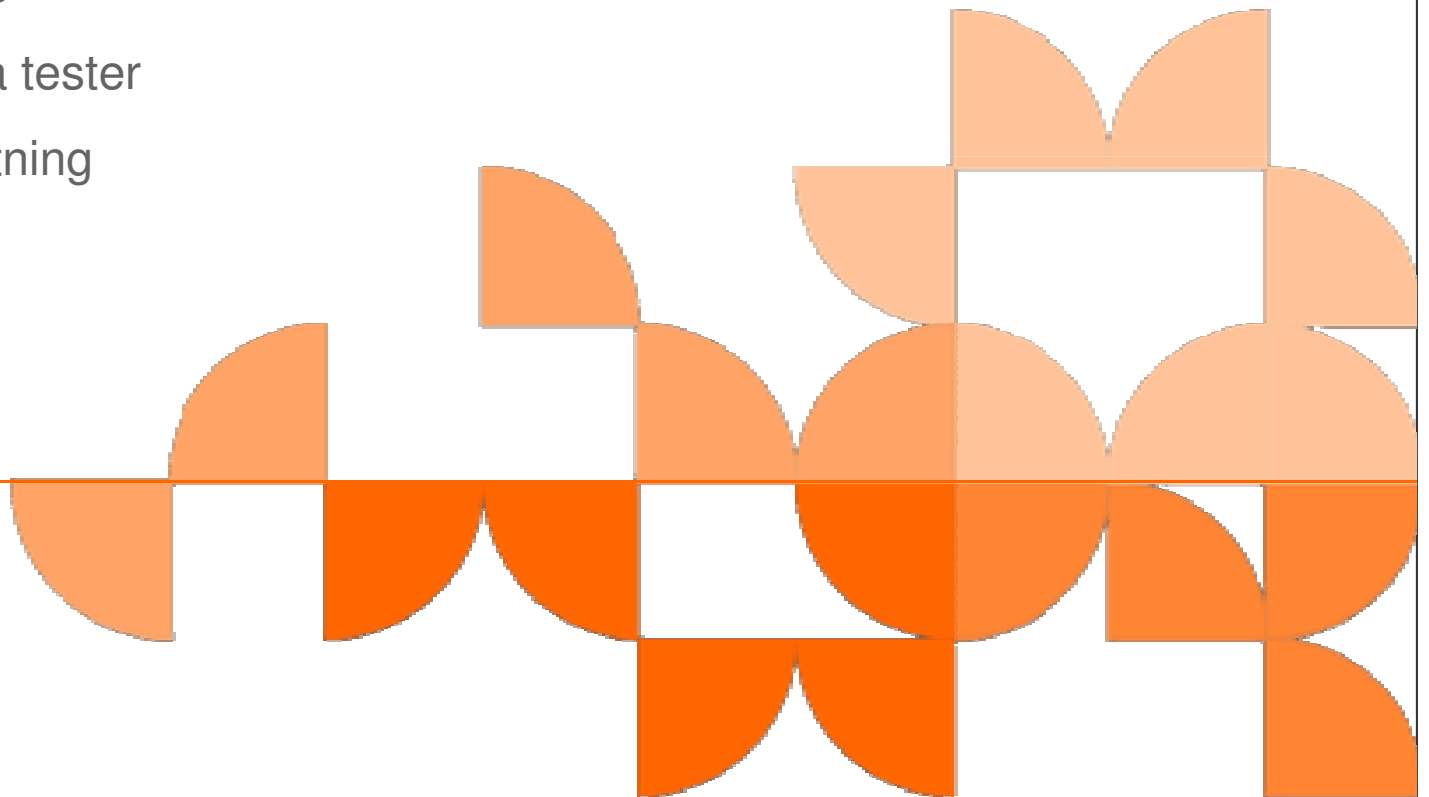
Vår ekvation

format × länder × integrationer × konfigurationer

= MYCKET TEST

Agenda

- Presentation av Extenda
- **Vad är en POS?**
- Test av POS
- Automatiska tester
- Sammanfattning



Vad är en POS?

- Point Of Sale
- Ett datorbaserat kassaregister där användaren kan skapa försäljningstransaktioner, hantera betalprocesser m.m.
- Vanlig periferiutrustning:
 - Touch-skärm
 - Specialanpassat tangentbord
 - Streckkodsläsare
 - Kvittoskrivare
 - Kunddisplay
 - Kassalåda
 - Våg
 - Betalterminal
- Mjukvara
 - POS
 - POS server
 - Back Office
 - Central Office

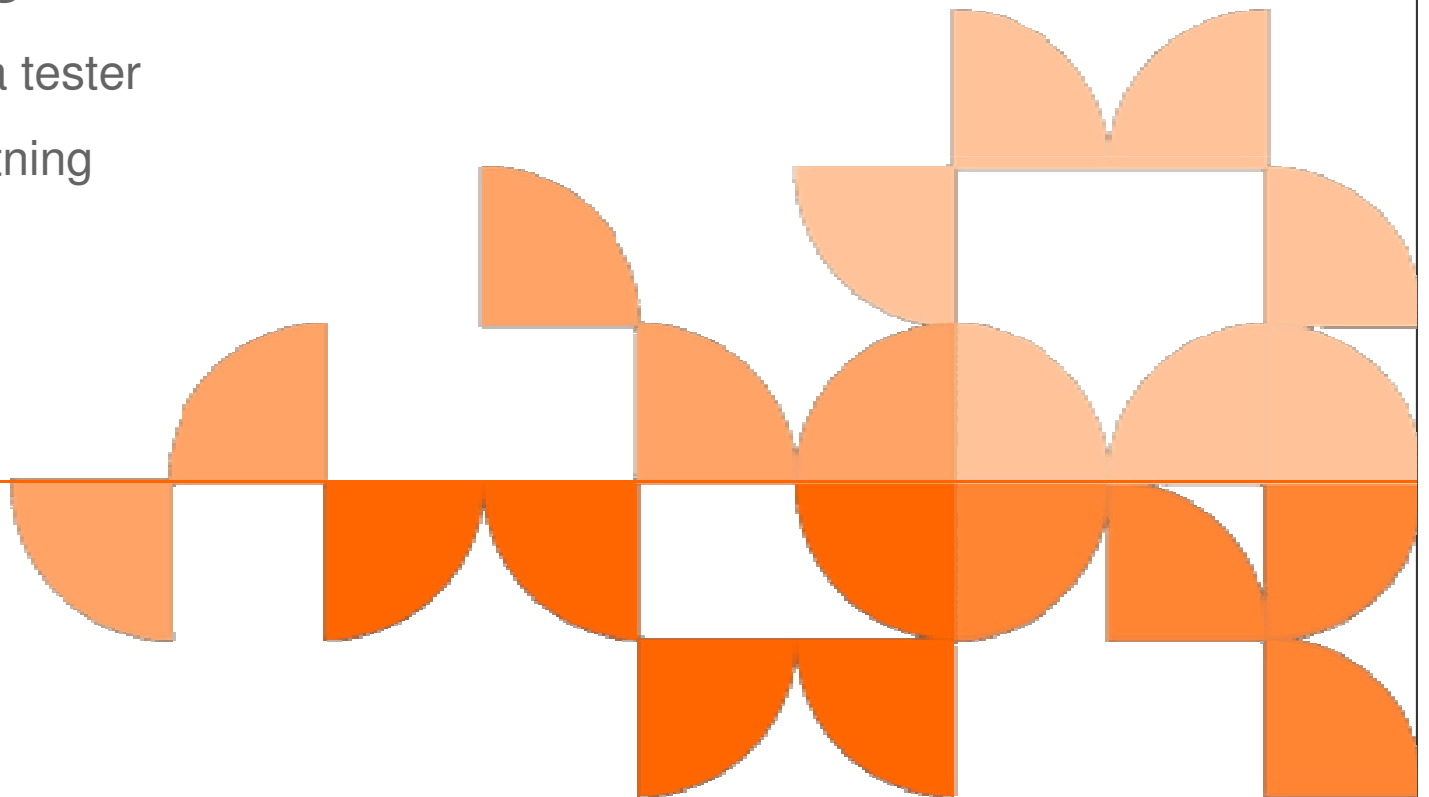


Extendas POS

- Applikation och ramverk
- Drygt 80 definierade användningsfall
- Koden uppdelad i olika delsystem för att kunna återanvändas i andra applikationer
- All klientkod är inte tillgänglig för test
- Flera aktiva versioner
- Höga krav på bakåtkompabilitet
- Java
- 250000 rader kod i ramverket

Agenda

- Presentation av Extenda
- Vad är en POS?
- **Test av POS**
- Automatiska tester
- Sammanfattning



Extenda - Test organisation

Distribuerad testverksamhet inom våra Quality Delivery units och Product Development

Testarnätverk

Testledarforum

Ca 25 testare och testledare

Test strategi baserad på ISTQB Foundation Syllabus

- SCRUM / RUP och W-modellen, kund-/produkt-/företagsanpassad release-strategi

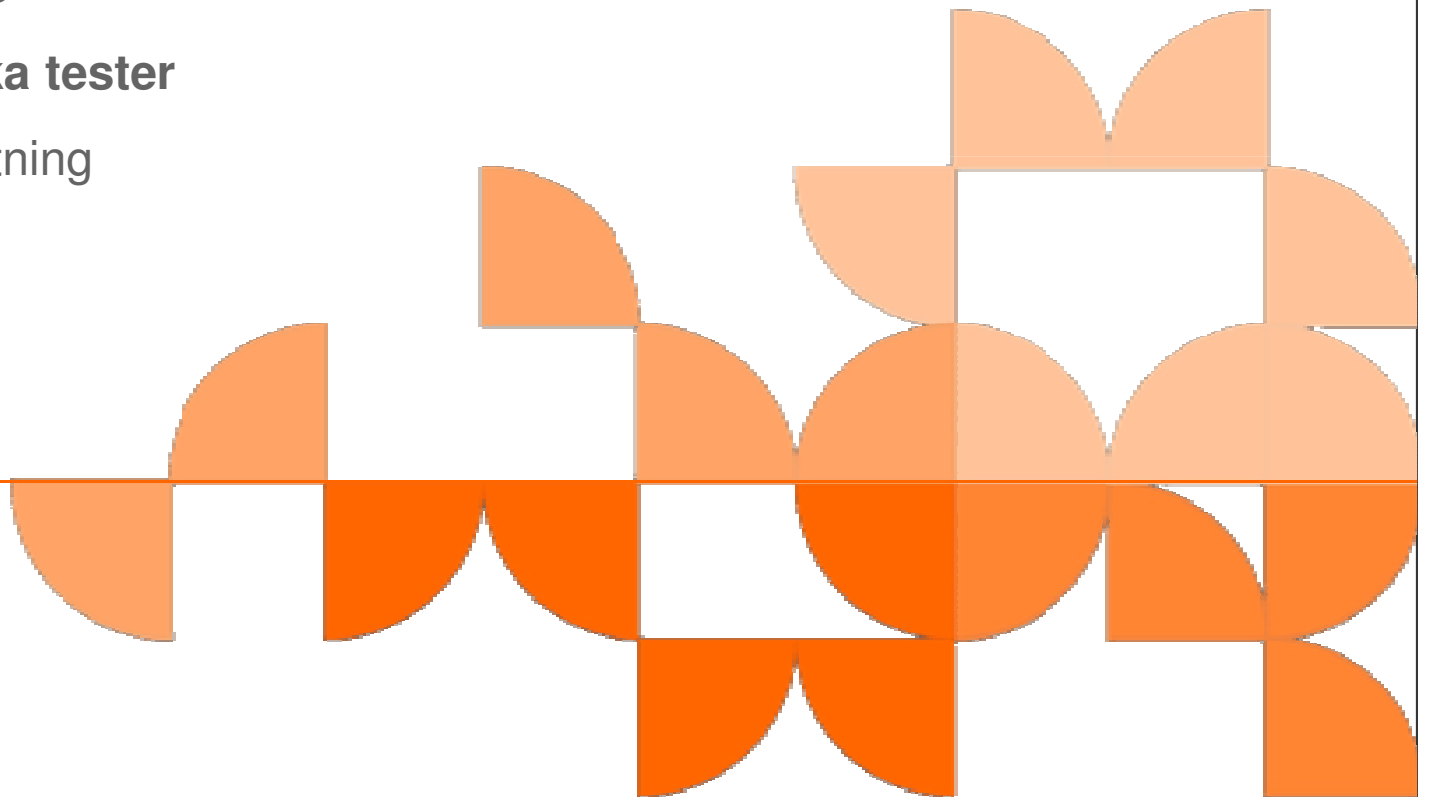
Testmål definierade för varje kund och produkt

Spårbarhet Krav och Testfall hanteras i DOORs och Test Director per kund

Manuell och automatiserad test

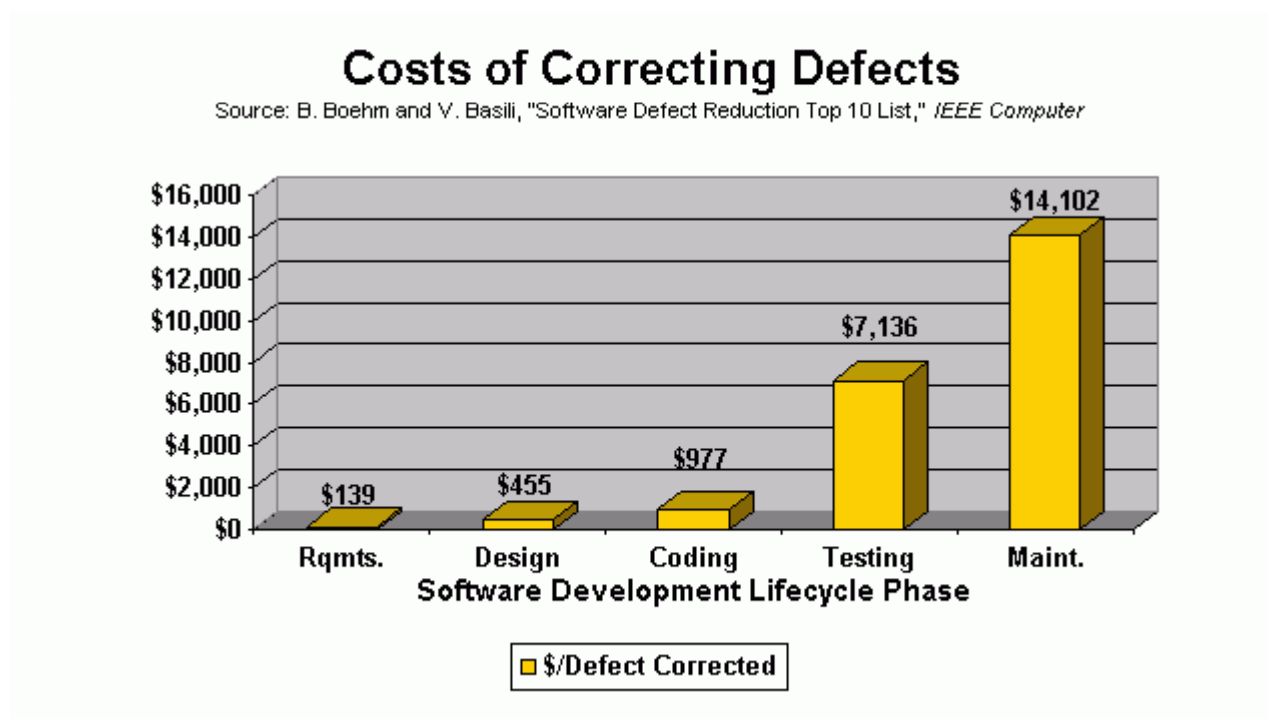
Agenda

- Presentation av Extenda
- Vad är en POS?
- Test av POS
- **Automatiska tester**
- Sammanfattning



Automatiska tester

- Kostnaden för att rätta defekter ökar dramatiskt när man går från utvecklingsfasen till test- och underhållsfaserna.
- Viktigt att hitta och rätta defekter medan det fortfarande är billigt



Syftet med automatiska tester

- Hitta fler defekter innan leverans till test
- Få bort defekter som dyker upp om och om igen
- Frigöra tid hos testgruppen för mer avancerade tester
- Göra utvecklarna modigare när det gäller att förändra existerande kod



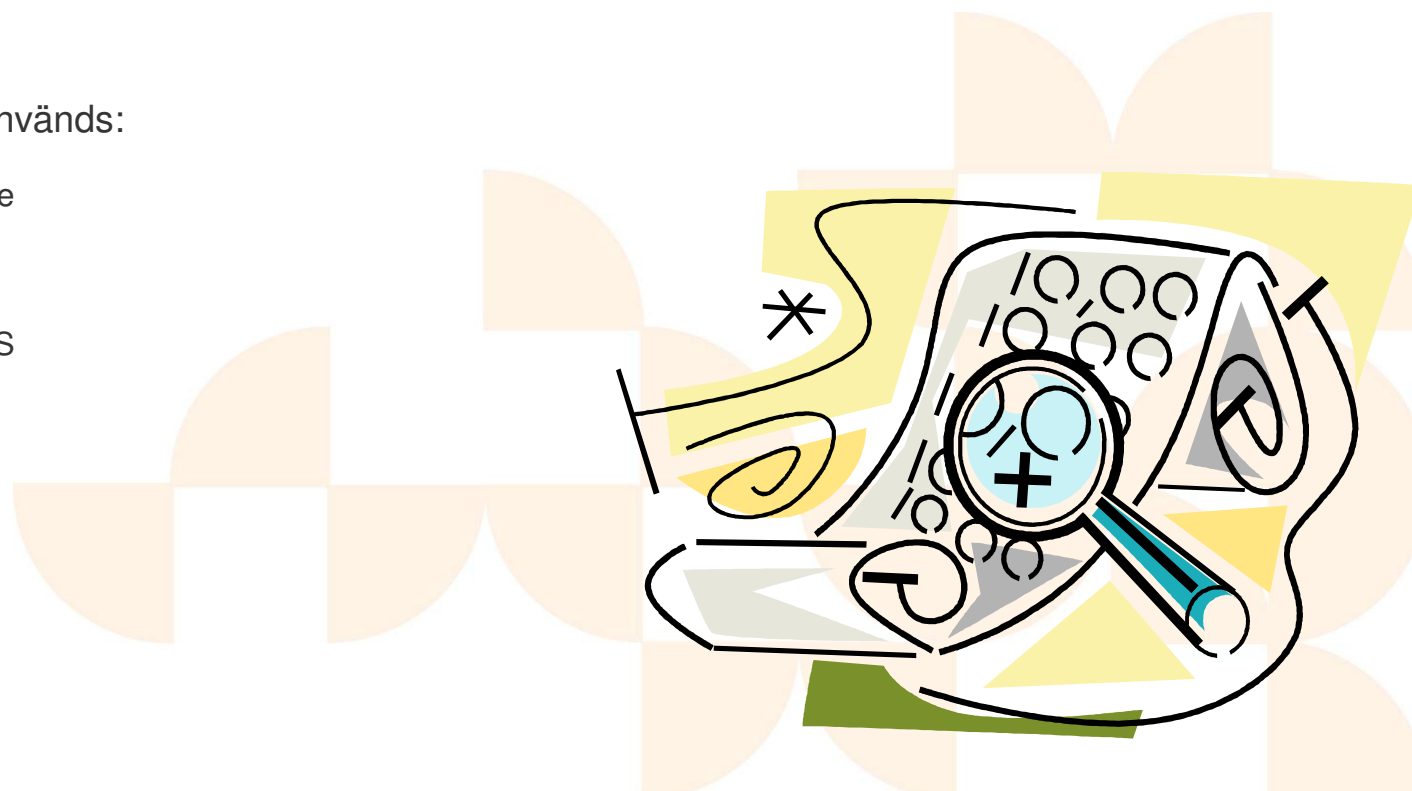
Typer av automatiska tester

- Statisk analys av källkod
- Test av funktionalitet
- Test av 3:e partskod/-applikationer
 - Databashanterare
 - Kodbibliotek/drivrutiner
- Komponenttest
 - Uppfyller komponenterna de krav som ställs av komponentramverket
- Meta-tester
 - Finns det unit-tester för alla komponenter
 - Täcker testerna tillräckligt stor del av koden



Statisk analys

- Analys av koden görs i samband med byggsteget
- Rapporter genereras med detaljerad information om analysen och vilka fel som hittats
- Verktyg som används:
 - Checkstyle
 - FindBugs
 - JavaNCSS
 - JDepend

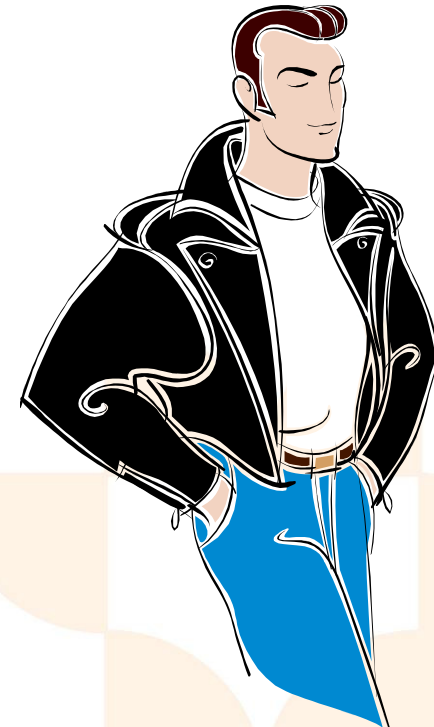


Checkstyle

Checkstyle kontrollerar att koden ser snygg ut och följer kodstandard. T.ex att:

- koden är dokumenterad
- namnkonventioner följs
- man håller sig inom definierade storleksgränser
 - Rader per fil
 - Rader per metod
 - Tecken per rad
 - etc
- det inte finns copy-paste kod
- komplexitetstalen för metoder är på en rimlig nivå

Checkstyle kan integreras med utvecklingsmiljön Eclipse. Detta gör att utvecklaren direkt ser när ett problem uppstår.



JavaNCSS

Räknar ut måttal för:

- Antal kodrader
- Antal dokumentationsrader
- Komplexitetstal för metoder

Används i byggprocessen och genererar rapporter som ligger till underlag för beslut om refakturering av kod.



FindBugs



Findbugs letar efter defektmönster i kod

- Knepiga egenheter i programspråket
- Missförstådda API metoder
- Felaktiga booleska uttryck

FindBugs kan integreras i utvecklingsmiljön Eclipse och är ytterligare ett exempel på hur man kan hitta fel så tidigt som möjligt.

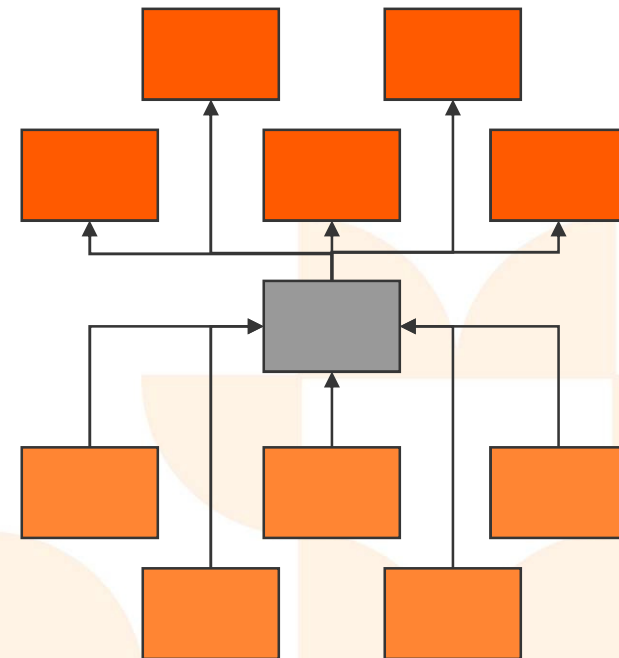
JDepend

JDepend mäter designkvalitet utifrån hur lätt koden är att

- utöka
- återanvända
- underhålla

Mätetal räknas fram som visar hur beroende/oberoende och hur abstrakta/konkreta olika delar av koden är.

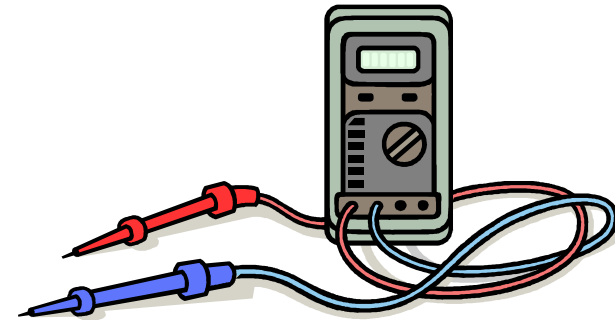
JDepend-rapporten som skapas i byggprocessen används som underlag för att förbättra ramverksegenskaperna i koden.



JUnit

JUnit är ett verktyg för att exekvera enhetstester i form av kod. Vi använder JUnit för att testa:

- Funktionalitet
 - API:er
 - affärsregler
- Egenskaper hos 3:e parts-kod
 - Transaktionsnivåer i databaser
 - Trådhantering i drivrutiner
- Komponenter
 - Uppfyller komponenterna de krav som ramverken ställer
 - Finns det dokumentation som beskriver komponenten
 - Är eventuella konfigurationspunkter dokumenterade
 - Har gränssnitt och klasser gjorts tillgängliga för användning i andra kodpaket på ett korrekt sätt
- Meta-tester
 - Finns det enhetstester för alla komponenter



Även Junit finns integrerat med utvecklingsmiljön



- Emma mäter täckningsgraden av enhetstesterna på olika nivåer.
 - Klass
 - Metod
 - Rad
 - Block
- Data från flera körningar kan ackumuleras
- Kan integreras med utvecklingsmiljön

```
148  *@see FileFilter#accepts
149  */
150  public boolean accept(File f) {
151      if(f != null) {
152          if(f.isDirectory()) {
153              return true;
154          }
155          String extension = getExtension(f);
156          if(extension != null && filters.get(getExtension(f)) != null) {
157              return true;
158          }
159      }
160      return false;
161  }
162  }
```

Automatiserade Regressionstester - Varför?

- Manuella regressionstester tar lång tid
 - Produkten är funktionsrik med många användningsfall, var och ett med ett antal flöden
 - Produkten är också mycket konfigurerbar
 - ~ 1 man-vecka för regressionstest av hela produkten
- Fel hittades sent på releasecykeln
- Förändringar kan påverka andra delar av systemet
- Utvecklare var "alltför" försiktiga att göra ändringar i koden
- För att lösa dessa problem, fanns ett behov av verktyg för automatiserade regressionstester.

Extenda Cashier Player - ECP

- Produktens komplexitet gjorde att befintliga produkter på marknaden inte gick att använda
- Extenda beslöt att bygga sitt eget verktyg som kallas "Extenda Cashier Player".
- ECP simulerar kassören genom att spela in kassörens handlingar för att kunna spela upp dem vid ett sendare tillfälle.
- Handlingar spelas in i enheter som kallas "Script". De kan ses som olika flöden i ett användningsfall.
- Script kan antingen köras ett och ett eller i "batch-mode".
- Faktiska utdata från testerna, alltså data på kunddisplayen, kvittoskrivaren osv, jämförs med förväntat utdata.
- E-post skickas till supporten när fel uppstår vid tester

GUI

The screenshot shows the Extenda - ECP Client application window. The title bar reads "Extenda - ECP Client" and the menu bar includes "File View ECP Client Help". The main interface features a configuration section with "POS Client Address" set to "localhost" and "POS Client Port" set to "10990". Below this is a control panel with a play button, a red stop button, a grey pause button, and a double-line stop button. The status area displays "Stopped/Idle state".

The "Test Report" tab is active, showing the following details:

- Result for Test Script 'testResult1_2008-09-14 003910'
- ECP Test Results**
- POS Client: swchi-vi-x64
- Date: Sun Sep 14 00:38:49 CEST 2008

Device	Expected	Actual	Pass/Fail
LINE_DISPLAY	New customer	New customer	PASS
LINE_DISPLAY	Plastic bag 0.15To pay 0.20	Plastic bag 0.15To pay 0.20	PASS
LINE_DISPLAY	Paper bag 0.20To pay 0.40	Paper bag 0.20To pay 0.40	PASS
LINE_DISPLAY	Cash 0.40Cash back 0.00	Cash 0.40Cash back 0.00	PASS
PRINTER			PASS
PRINTER	1B	1B	PASS
PRINTER	N 1C	N 1C	PASS: - Ignored since header row
PRINTER	Adolf Fredriks Kyrk.. +46-(0)8-50537000	Adolf Fredriks Kyrk.. +46-(0)8-50537000	PASS: - Ignored since header row
PRINTER	SE-111 37 Stockholm www.extenda.se	SE-111 37 Stockholm www.extenda.se	PASS: - Ignored since header row
PRINTER	Salesm: 1 Lane: 75 No: 41	Salesm: 1 Lane: 75 No: 42	PASS: - Ignored since header row
PRINTER	Date: 2008-09-14 Time: 00:38	Date: 2008-09-14 Time: 00:39	PASS: - Ignored since header row

At the bottom left of the window, a status bar indicates "Application started".

Nuläge

- Alla våra regressionstester är inspelade som ECP scripts
- Varje natt kör ECP alla script mot det senaste bygget av produkten.
- Produktteamet informeras via e-post om utfallet av testerna.
- Vi genomför alltså dagliga regressionstester av den senaste kodbasen för vår produkt.

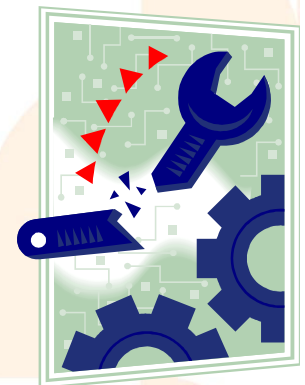


Vinster

- Defekter hittas tidigt.
- Utvecklarna kan göra refaktoreringar och andra kodändringar utan någon större oro.
- Utbyte av produktplattform eller middleware, som exempelvis databas, kräver inte en stor testinsats.
- Väsentligt reducerad testtid gör att testarna kan koncentrera sig på tester av mer komplex art.
- Kunder med specifikt anpassad kodbas kan skapa och köra script för denna.
- ECP är, tillsammans med andra former för automatiserade tester, en avgörande faktor för att Extenda ska leverera högkvalitativa produkter.
- Vinsterna överväger med god marginal kostnaderna för att utveckla ECP.

Problem med automatiserade tester

- Uppföljning av testresultat
 - Finns ofta fler fel än vad utvecklarna kan fixa och som det är ekonomiskt försvarbart att fixa
- Falska negativ
 - Verktøygen är inte 100%-iga. Ibland rapporterar de fel för saker som inte är fel
- Testerna ökar kodbasen
 - Ökat utvecklingskostnad
 - Ökat underhåll
 - Fler defekter
 - Tar längre tid att exekvera

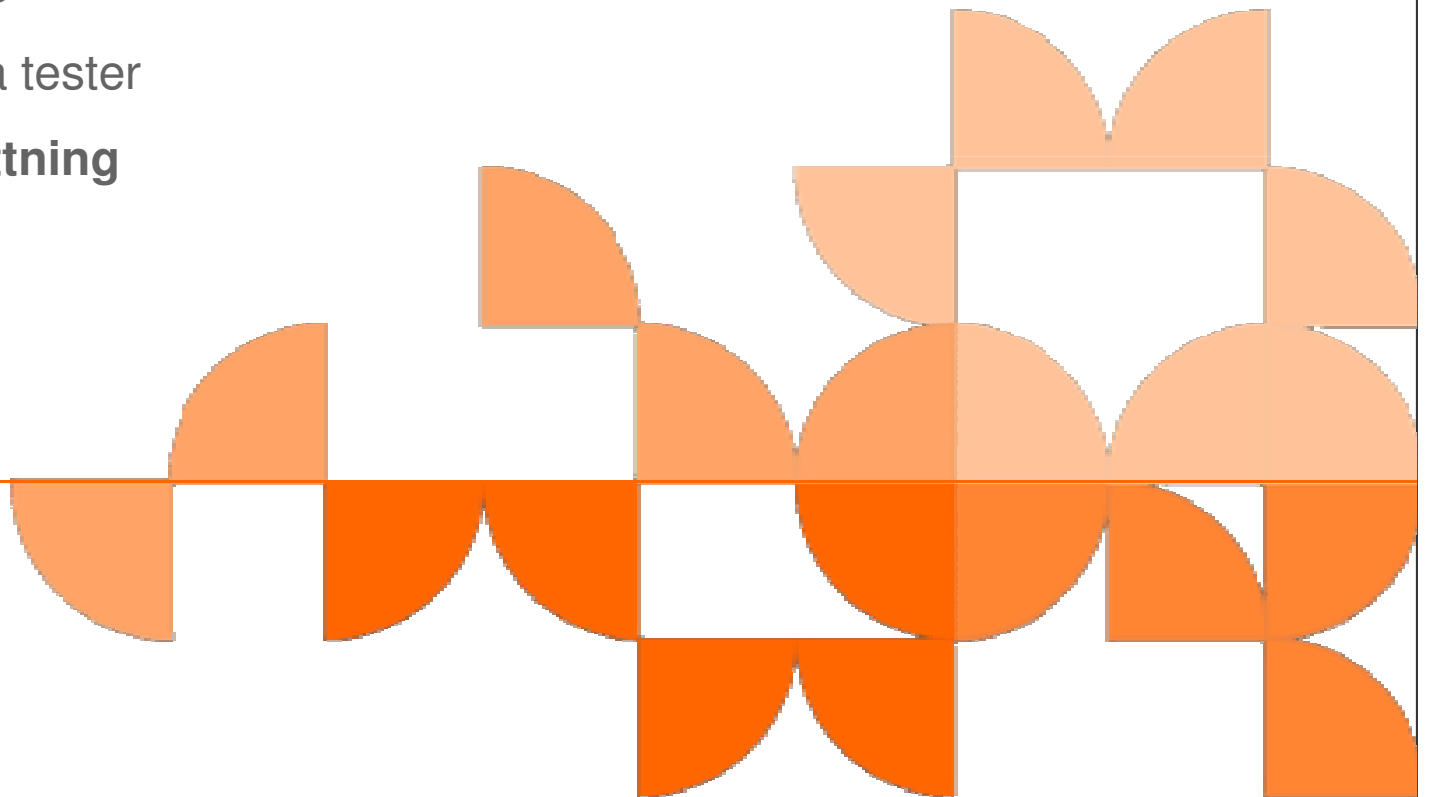


Framtida förbättringar

- Dela upp enhetstesterna i olika grupper som körs
 - Innan incheckning till versionshanteringsverktyg
 - Vid varje bygge
 - Varje natt
 - Inför release
- Samla täckningsdata från körningar av Cashier Player
- Reducera antalet falska negativ

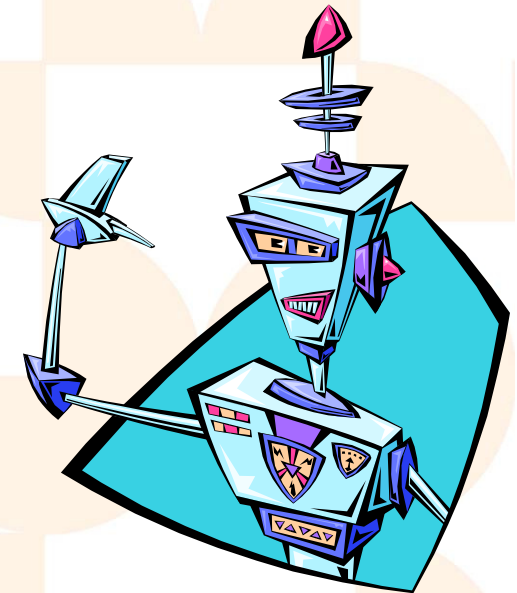
Agenda

- Presentation av Extenda
- Vad är en POS?
- Test av POS
- Automatiska tester
- **Sammanfattning**



Sammanfattning

- Automatiserade tester lönar sig
- Var inte rädda för att utveckla egna testverktyg
- Hitta felen så tidigt som möjligt, helst samtidigt som koden skrivs.
- Bygg in testbarhet i applikationerna



Länkar

- Checkstyle (<http://checkstyle.sourceforge.net/>)
- JavaNCSS (<http://www.kclee.de/clemens/java/javancss/>)
- FindBugs (<http://findbugs.sourceforge.net/>)
- JDepend (<http://clarkware.com/software/JDepend.html>)
- Junit (<http://www.junit.org/>)
- Emma (<http://emma.sourceforge.net/>)

- Doors (<http://www.telelogic.com/products/doors/>)
- TestDirector (<http://www.hp.com>)

- Extenda (<http://www.extenda.se>)



EXTENDA[®]

peter.mathsson@extenda.se

anders.rennermalm@extenda.se

