

# Testautomatisering

Intro

- FM:
  - Presentation
  - Genomgång av Kursplan / Kursupplägg
  - Varför testautomatisering?
  - Video + diskussion

**Idag**

- David Gullmarsvik
- david.g@jetas.se
- Software Developer
- Tidigare
  - Lärare KYH, TI
  - Datateknik
  - Systemtekniker Tele/Nätverkstekniker

**Presentation**

- Jetas Quality Systems
  - Systematiskt fastighetsunderhåll
  - Tid.: Testscript + Externa Testare
  - Nu: Automatiserade + Manuella, interna test

**Presentation**

- Agile?
- Selenium?
- Acceptanstest?
- Version Control (git / hg)

**Frågor**

- Genomgång Kursplan

**Intro**

- Genomgång Kursplanering

**Intro**

- Vad är problemen med Mjukvara/Mjukvaruutveckling?

**Intro**



- Vad är problemen med Mjukvara/Mjukvaruutveckling?
  1. Resultatet är för dåligt
  2. Det kostar för mycket

- Kuriosa: När uppstod det här problemet?

**Intro**

- Kuriosa: När uppstod det här problemet?
  - 1960-talet - The software Crisis
  - [http://en.wikipedia.org/wiki/Software\\_crisis](http://en.wikipedia.org/wiki/Software_crisis)

**Intro**

- The Software Crisis - Hur löser vi det?

**Intro**

- The Software Crisis - Hur löser vi det?
  - Med flera olika verktyg
  - Planering, Processer/Utvecklingsmetodik, Arkitektur, Design, Struktur, mätbara nyckeltal, etc...
  - En väldigt viktig komponent: Test

**Intro**

- Vad är problemet med Test av mjukvara?

**Intro**

- Vad är problemet med Test av mjukvara?
  1. Resultatet är för dåligt
  2. Det kostar för mycket

**Intro**

- Hur löser vi detta?
  - Planering, Processer/metodik, Design, Struktur, mätbara nyckeltal, etc...
  - En väldigt viktig komponent: Automatiserade Test

**Intro**



- Först och främst:
  - Testautomatisering är mjukvaruutveckling
  - I väldigt stor utsträckning

**Intro**

- Samma regler som för mjukvaruutv:
  - Vilket värde tillför detta?
  - Går detta att underhålla?
  - Automatiska tester kan innehålla buggar.

**Intro**

- Är vi inte tillbaka på ruta 1?
  - Nej, inte så länge vi hittar kritiska problem snabbare, effektivare och till lägre kostnad än tidigare

**Intro**

- Varför testautomatisering?

**Intro**

- Någon som hört talas om FitNesse?



**Intro**

- Någon som hört talas om FitNesse?
  - Ett verktyg för testautomatisering
  - Skapat av bl.a. Robert C. Martin (Författaren till Clean Code)

**Intro**

- Hur sköter FitNesse testautomatisering?
- Videoklipp från
  - <http://ndc2012.oktaset.com/t-4868>
- 29:44

**Intro**

- Vad är anmärkningsvärt här?
  - Testramverk (vi kan anta att det skapas av några som vet någonting om test) helt utan manuella tester

**Intro**



- Vad är anmärkningsvärt här?
  - Väldigt snabbt - potentiellt  $\sim 1.5$  min från beslut att släppa en ny version tills vi påbörjat släppet.

**Intro**

- Vad är anmärkningsvärt här?
  - "Stort" program - fler användare än det genomsnittliga projektet

**Intro**

- Vad är anmärkningsvärt här?
  - ...Jag vill dock inte att ni tar med er att testautomatisering ersätter all annan testning
    - => För det mesta är det ett komplement till manuella tester.

**Intro**

- Vinster med testautomatisering
  - Spara tid
    - När? Var lägger vi tiden istället?
  - Levande specifikation
  - Visa saker kan lämpa sig dåligt för manuellt test
  - "Keeps us sane" - repetativt, tråkigt, enkelt
  - (Vi glömmer ingenting)

**Intro**

- Exempel då testautomatisering är lämpligt
- Vi har 100 testfall för en webbapplikation
  - hur lång tid tar detta att testa?

**Intro**

- Vissa saker kräver automatisering:
  - Hur mycket behöver vi testa eg?
  - 5 browsers \* 5 versioner \* 5 plattformar \* 100 testfall

- Vissa saker kräver automatisering:
  - $5 \text{ browsers} * 5 \text{ versioner} * 5 \text{ plattformar} * 100 \text{ testfall}$
- Manuellt:
  - $X * 5 \text{ minuter} \Rightarrow 43 \text{ dagar}$
- Automatiskt:
  - parallellisering "gratis"  $\Rightarrow 100 \text{ testfall parallellt, snabbare}$   
test/dokumentation  $\Rightarrow 10\text{s/testfall}$
  - $X/100 * 10\text{s} \Rightarrow 20 \text{ min}$
  - (10s/testfall är en extremt hög siffra)

- Hur är detta relaterat till livscykeln för ett släpp?
- 1 släpp/år
  - => kanske fine att en testcykel tar 2 månader

**Intro**



- Agile mantra: "Release early, release often!"
- Release var 14:e dag
  - => vi vill ha tester klara på 1 vecka
  - => 7 testare på heltid
    - finns det budget för detta?
- Release varannan dag
  - => 43 heltidstestare
  - Är det realistiskt att göra ett släpp varannan dag?

**Intro**

- Större applikation:
  - => 100 000 testfall
  - => 43 000 dagar för manuella test
  - => 13 dagar för automatiserade test
- Är 100 000 testfall realistiskt?

- Exempel från Jetas:
  - Services är hjärtat i vårt system.
    - Ett antal automatiska program som utför arbete i bakgrunden
  - De har inget GUI
    - Man kan se effekterna av services i GUI – men det finns inga garantier för hur lång tid det dröjjer (sekunder upp till 10 minuter)
  - Manuella tester har visat sig vara mycket krångligt, på gränsen till omöjligt
  - Utan dessa services har vi inget system

**Intro**

- Ytterligare ett exempel
- SQLite är en databasmotor
  - <http://www.sqlite.org/testing.html>
- ~130k testfall körs efter varje individuell förändring
  - (tiden att utföra dessa kan räknas i sekunder)
- Fullständigt test: de listar > 10M test i olika kategorier
  - (? finns ingen fullständig listning, men de uppger > 10M test i olika kategorier, i verkligheten förmodligen ännu fler)
- (SQLite Open Source - många av deras test är tillgängliga för vem som helst)

## Intro

- Gratis är gott
  - Vi får flera saker på köpet när vi börjar automatisera tester
- => Gratis Smoke Test
- => Gratis ~Regression Test
- => Gratis Performance Test
- => Utöver färre buggar finna även annat att tjäna:
  - Testability => Maintainability
    - Kodunderhåll är generellt största kostnaden för mjukvaruutveckling
    - (Ex. genom att tvingar fram modularitet och mindre kodsnuttar)
  - Testability => Flexibility
    - Här spelar modulariteten in igen
    - (Kan anv. ex. för 100% uptime)

# Intro

- Utmaningar

- Värde

- Vilket affärsnytt eller värde för användare tillför våra tester?

- Återanvändning

- för att få något värde generellt, så måste testerna gå att återanvända
      - Ex: Tidsbaserade värden? Miljöberoende värden?

- Isolation

- Testar vi rätt saker? har vi dolda beroenden?

- Prestanda

- Setup? Network connections? Database calls? HD writes?

- Underhåll

- Vad händer när mjukvaran ändras, kan vi köra samma test om 1 år?

**Intro**

## Business Facing

C  
r  
i  
t  
i  
q  
u  
e

Functional  
Examples  
Story Tests  
Prototypes  
Simulations

Exploratory Testing  
Scenarios  
Usability  
User Acceptance Testing  
A/B Testing

Test

Unit Test  
Component Component Test

Performance & Load Testing  
Security Testing  
"Ility"-Testing\*

## Technology Facing

S  
u  
p  
p  
o  
r  
t

# Intro

\*"ility"-testing: maintainability, interoperability, compatibility, reliability, installability, etc...

- Caveat:
- Jag vill inte att ni skall få intrycket av att testautomatisering är ngt som man bara slänger på
- Jag vill inte att ni skall få intrycket av att testautomatisering är lämpligt för allt/löser alla våra problem

**Intro**



- Testautomatisering
  - Kan vara tidskrävande
  - Kan kräva att vi tänker om från grunden i vissa fall
  - Kan vara kostsamt
  - Är inte lämpligt för allting

**Intro**

- Testautomatisering
  - 85% av alla företag som försöker automatisera sina test misslyckas (statistik från Borland - suspekt)
  - Misstaget:
    - De "köper" testautomatiseringsprogramvara och räknar med att deras problem med tid och resurser är lösta

- Testautomatisering
  - De som lyckas?
  - Framgångsfaktor:
    - De jobbar medvetet och långsiktigt med testautomatisering, och tar med det i deras strategi från start

- Testautomatisering

- De som lyckas?

- Vinster

- Kan fokusera resurserna på att höja värdet av annan QA
    - Kan göra flera releases om dagen - vad innebär detta?
    - Kan göra förändringar utan att vara allt för rädda.

**Intro**

- Tidigare kritik mot den här kursen (2011)
  - Kursen var väldigt intensiv/svår
    - Jag har kikat på era programmeringslabbar.
  - Min målsättning är att kursen skall vara utmanande men ej omöjlig
  - Ni får gärna komma med feedback kring detta

**Intro**

- Tidigare kritik mot den här kursen (2011)
  - Installationsproblem
    - Hoppas vi undviker lite av detta med Virtual Machine.
    - Detta är alltid ett problem för kurser som behandlar mjukvaruverktyg – det blir värre ju fler verktyg man använder
    - Installation/Installationsproblem är en del av er vardag, bli bra på det så tidigt som möjligt.

**Intro**

- Tidigare kritik mot den här kursen (2011)
  - Denna kursen är för utvecklare
    - Testautomatisering "är" i stor utsträckning mjukvaruutveckling
    - Kursinnehållet är något förändrat dock (Tidigare: Mer fokus på CI, samt versionshantering).

**Intro**

- **Förväntningar**

- Har ni några förväntningar/synpunkter/etc. på kursen?
- Om ni kommer på ngt under kursens gång – kontakta mig via mail eller på lektionen.
- Ifall ni har kritik så är även det välkommet.

**Intro**



- Video

- Jag planerar att visa en video i veckan.
- Innehållet har beröringspunkter med testautomatisering.
- Till för att ge er lite andra perspektiv än mitt.

**Intro**

- Video

- Jag planerar att visa en video i veckan.
- Innehållet har beröringspunkter med testautomatisering.
- Till för att ge er lite andra perspektiv än mitt.
- Vi diskuterar videon gemensamt i klassen sedan.

**Intro**

- Video

- Ytterligare ett syfte:

- Ni jobbar i en snabbt föränderlig branch – ni kommer behöva snappa upp ny information

- Jag vill att ni upptäcker några av de konferenser, etc. som tillgängliggör videos online (dvs. kika på adressen för videon).

**Intro**

- Video

- Future of Testing and Quality

- Goranka Bjedov - Facebook

- <http://oredev.org/2012/sessions/future-of-testing-and-quality>

**Intro**

- Watir + Övning + Lab

**Nästa gång**

- Alt 1: Installera Virtualbox
  - [www.virtualbox.org/wiki/downloads](http://www.virtualbox.org/wiki/downloads)
  - Virtual Machine + Instruktioner kommer på kurshemsidan ikväll.

**Nästa gång**

- Alt 2: Installera Ruby + Watir
  - Installationsanvisningar:
    - (Dessa är något mer detaljerade än watir-hemsidans)
    - [watir.github.com/watirbook](http://watir.github.com/watirbook)
  - Varning för Windows
  - Kommer Virtual Machine
  - Kommer lista över vad jag installerat på min Virtual Machine

**Nästa gång**

...

Fin