

SAST Q1

”Som att börja arbeta på ett nytt jobb”

*Testautomatisera med Modell-baserad
testning*

Christina Nordström

Kristian Karl



prolore

Christina Nordström

- Test sedan 1996
- Aldrig testautomatiserat
- Enhetschef Testenheten It-avdelningen PPM
- Tf chef It-avdelningen PPM (fr o m januari 2009)



Kristian Karl

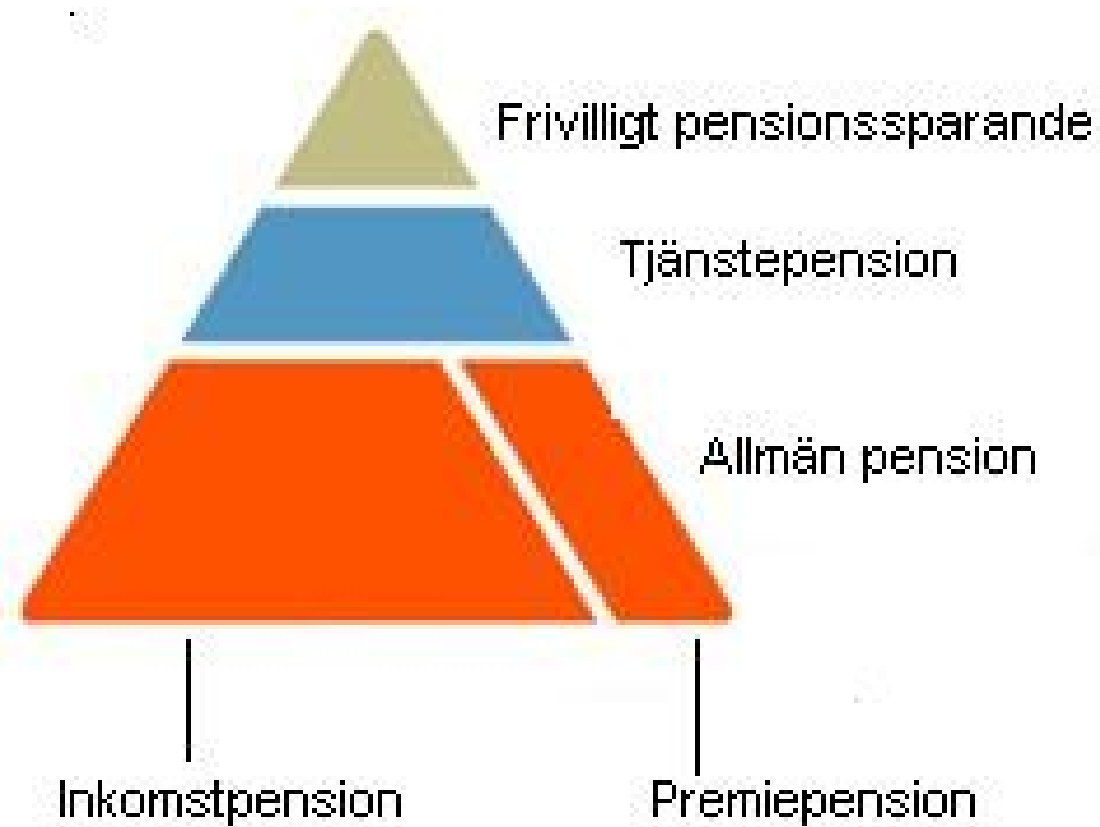
- Testautomatiserat sedan 1994
- Modell-baserad test sedan 2004
- Arbetar på Prolore
*[Specialiserat på Testautomatisering,
Prestanda samt Utbildningar inom dessa
områden]*



Agenda

- PPM
- Projektet
- Testautomatisering
- Modell-baserad test (MBT)
- MBT på PPM
- Reflektioner
- Framgångsfaktorer

PPM

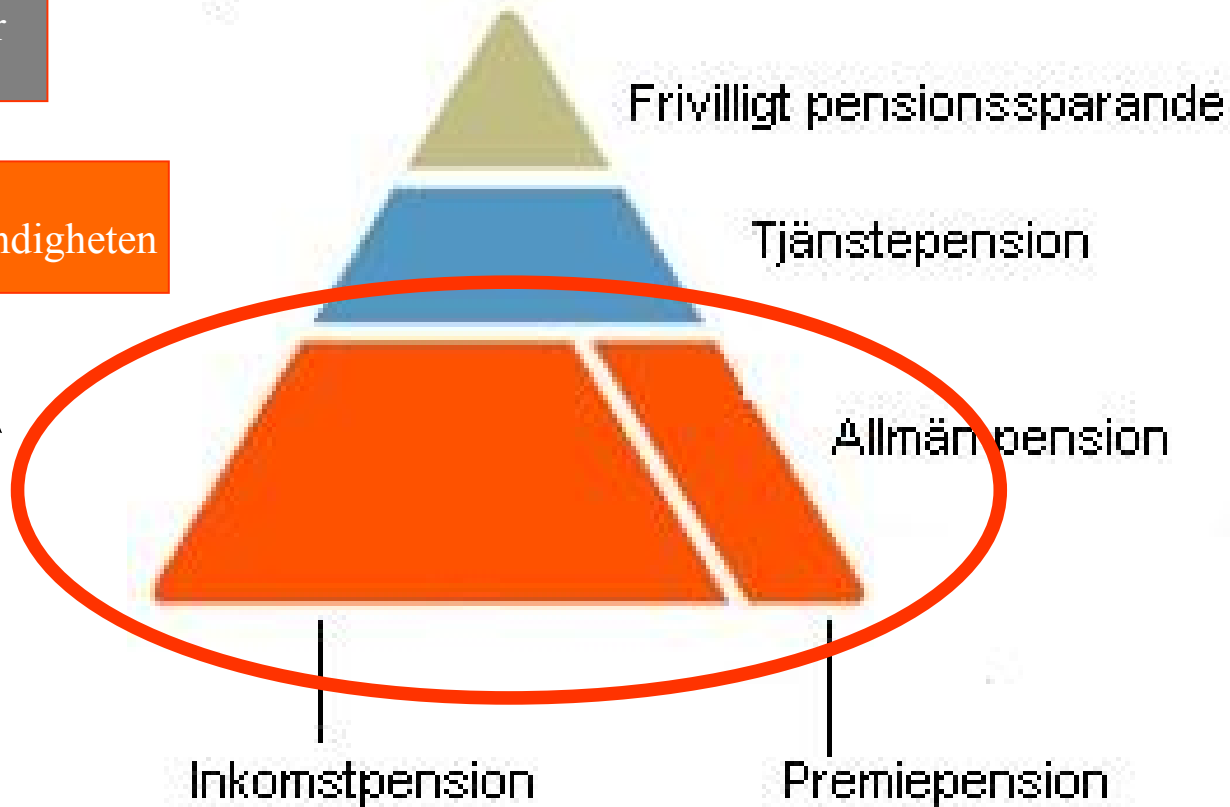
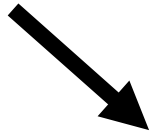


PPM + FK = nästan sant

~~PPM~~

315 dagar
kvar

Pensionsmyndigheten



PPM idag



- Ca 220 anställda
 - 77 på it-avdelningen
 - 11 på testenheten
- 6,1 miljoner kunder
 - 560 000 pensionärer
- Fondvärde ca 300 miljarder kronor
- 500 miljoner rader i transaktionstabellen

PPM enl Regleringsbrevet



PPMs uppgift är bl a

- Administrera och förvalta premiepensionssystemet
- Verka för ökad kunskap om premiepensionen

Syftet med 2K

Externa aktörer

- Ge information om innehav på sitt premiepensionskonto på extern aktörs webbplats
- Möjliggöra fondbyten, på sitt premiepensionskonto, via en extern aktörs webbplats
- Skapa en säker lösning för att hantera stora mängder fondbyten som förvaltningsföretag gör för p-sparares räkning



Tekniken i projektet

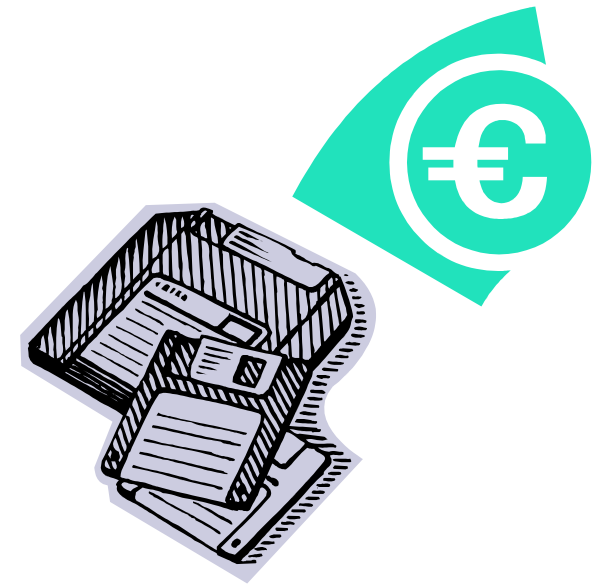
- Synkrona- samt asynkrona meddelanden
- Webservices (SOAP)
- Filöverföringar (batchkörningar)
- XML fram och tillbaka, hit och dit

Syftet med 2K - IT-projektet

Utveckla och produktionssätta tekniskt gränssnitt och infrastruktur som gör detta möjligt!

Projektets förutsättningar

- Budget IT - 25,8 miljoner kronor
- Ca 28 000 mantimmar (blir 35 000)
- Ca 20 konsulter och 10 anställda
- Scrumprojekt!!! (4 team)



Varför testautomatisera?

- Testa varje gång det har skett en förändring!
- Hitta fel så snabbt som möjligt
- Bli lika agila som utveckling (!?)

Naturligtvis finns det tester som inte kommer att automatiseras.

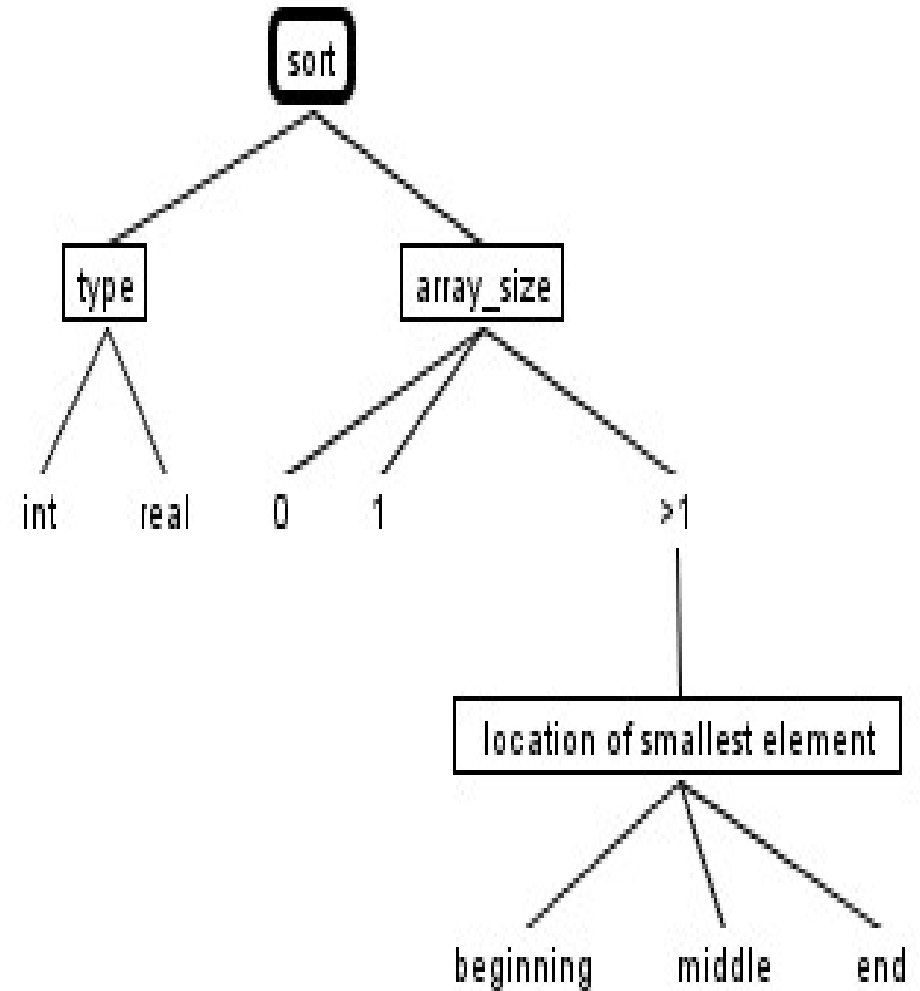
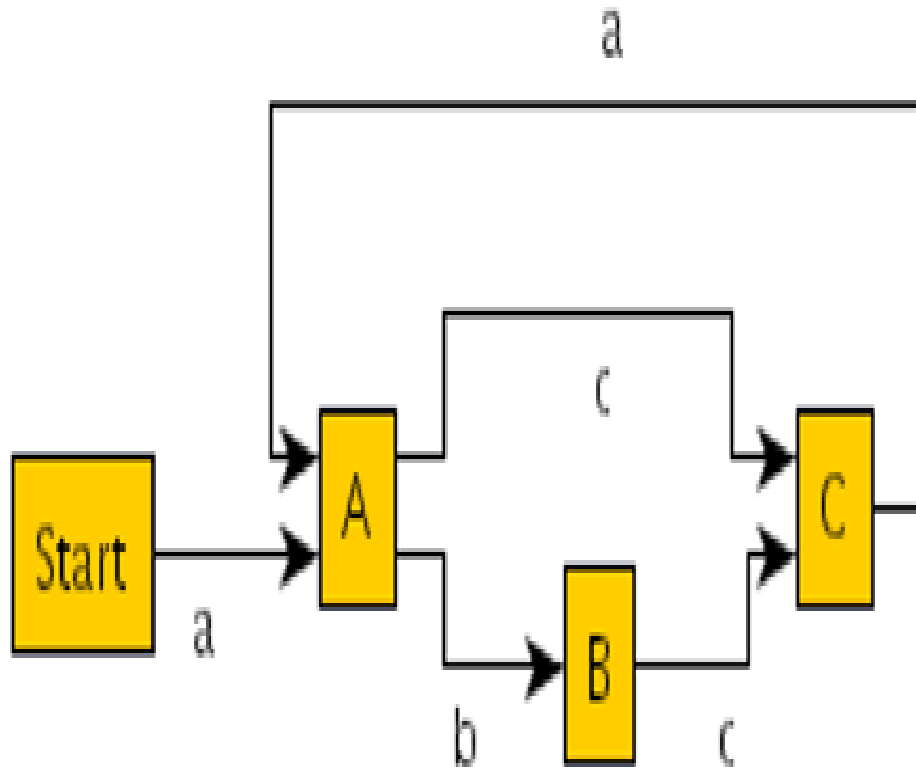
Tidigare försök...

- Försök har gjorts
- Underskattat insatsen – inte avsatt så mycket tid och resurser som verkligen krävs
- Verktygssinriktat

Varför gick det bättre nu?

- Testautomatisering är inget verktyg
- Det är en process
- Verktyg, metoder och personer är lika viktiga

Modell-baserad testning



Vad är MBT?



Wikipedia

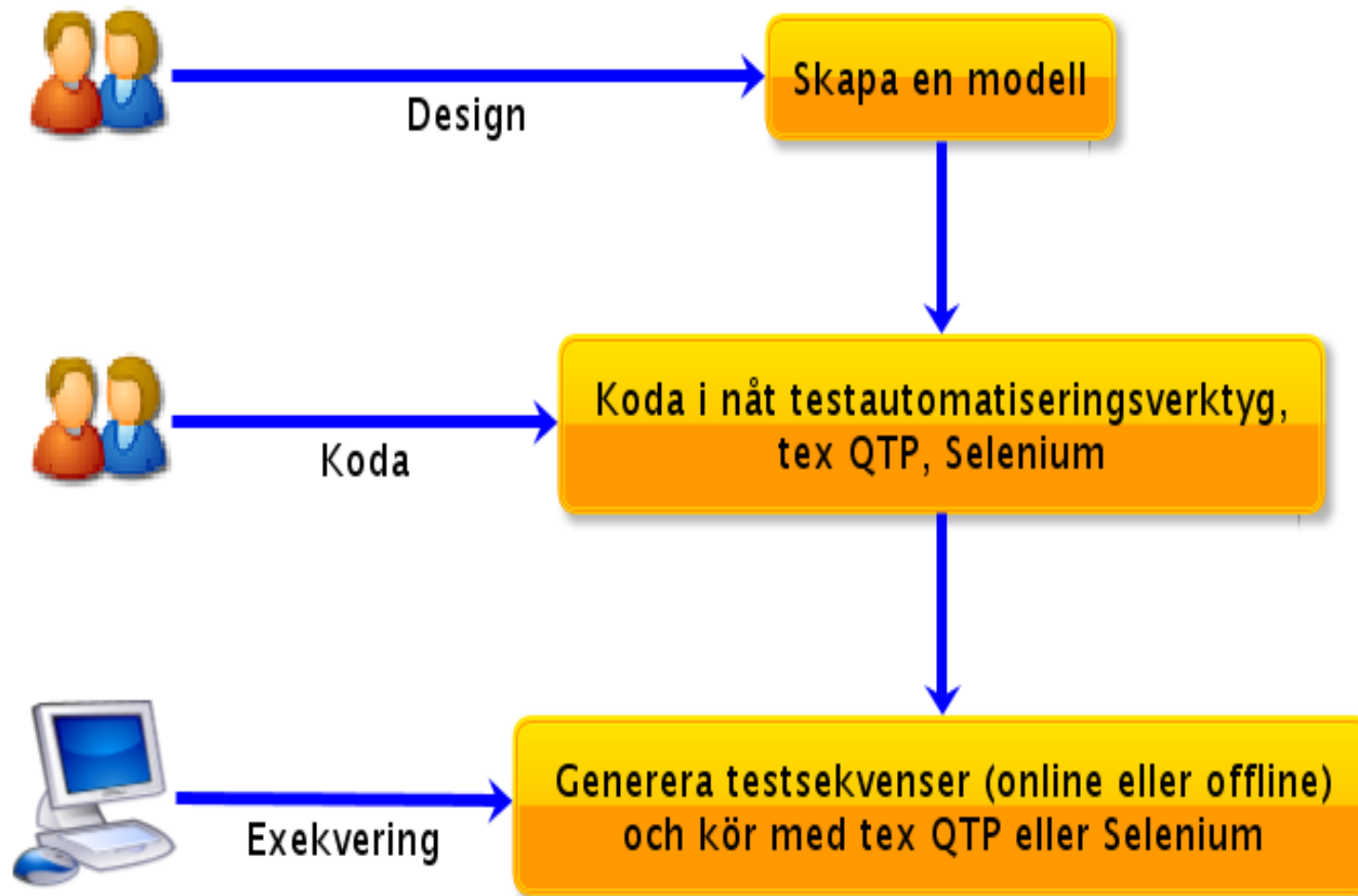
Test case generation by:

- ***theorem proving***
- *constraint logic programming*
- *model checking*
- *symbolic execution*
- *using an event-flow model*
- ***using an Markov chains model***

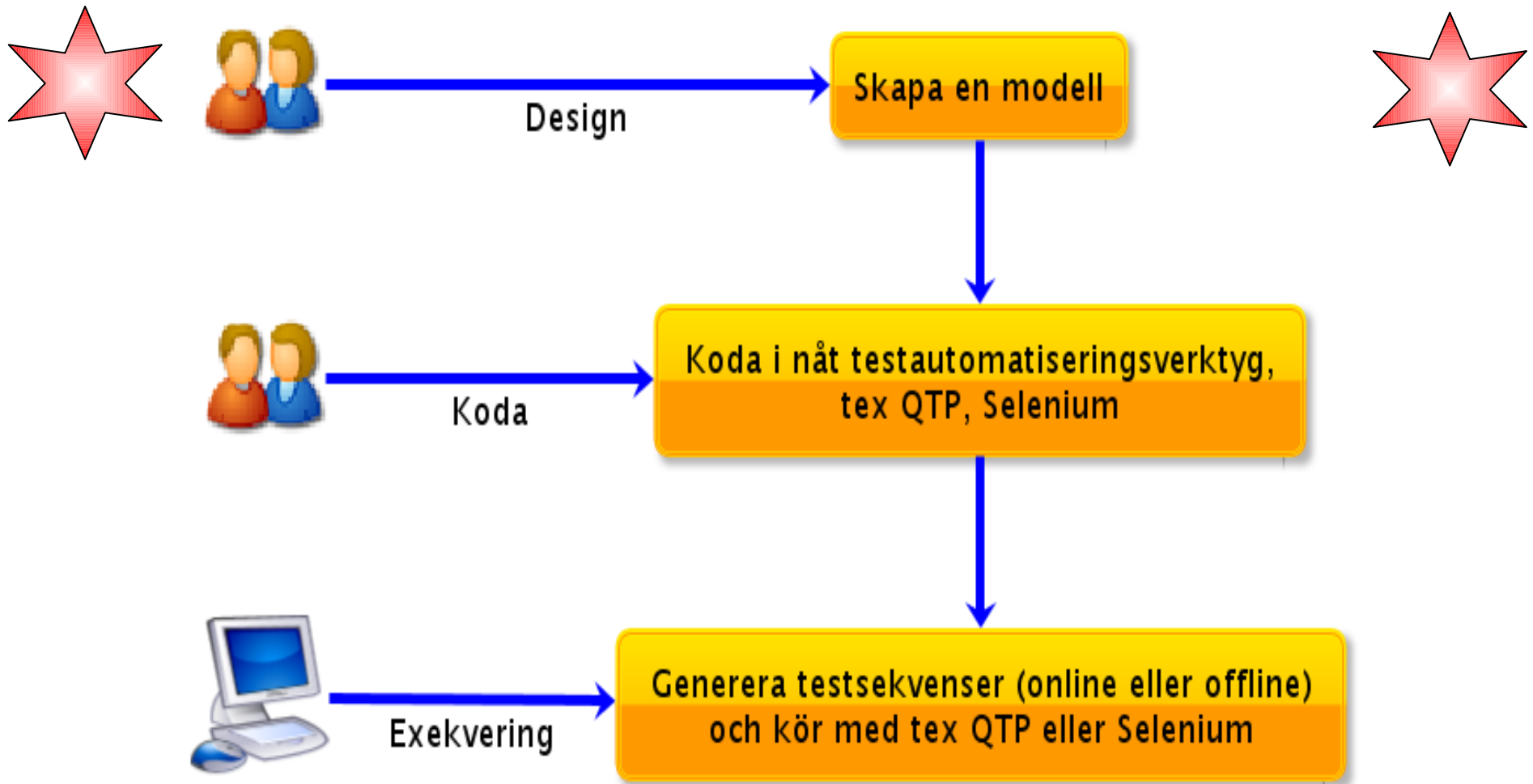
Varför MBT?

- Modeller är enklare än de system som de avbildar, därmed enklare att förstå
- Visuella!
- Enklare att designa tester
- Lyfter bort logiken från skripten
- Lättare att underhålla

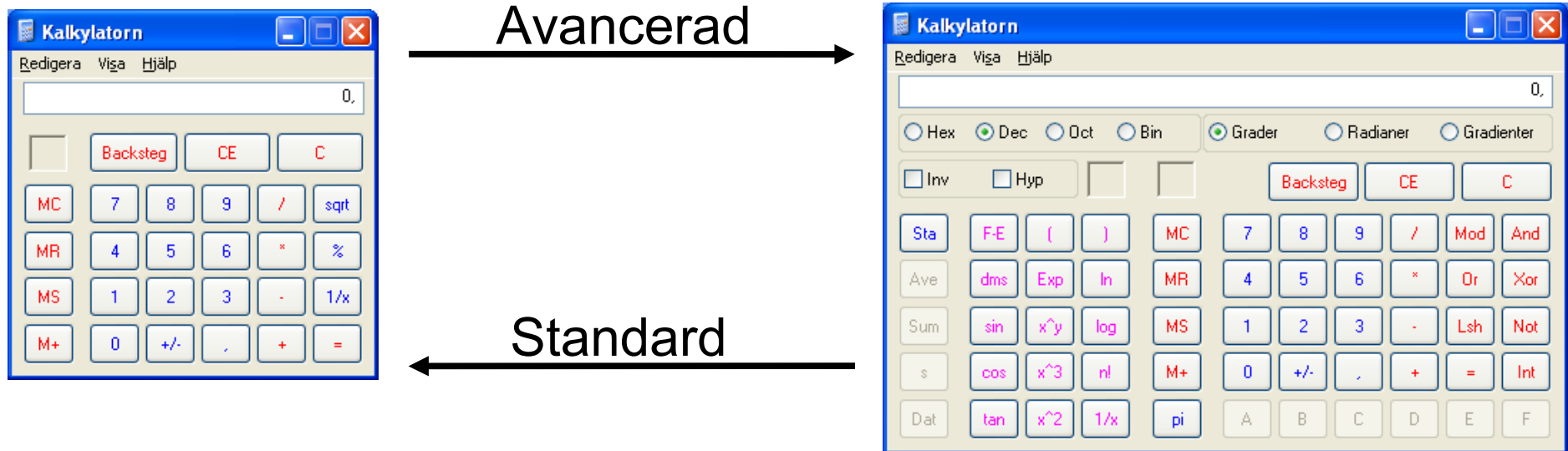
MBT Crash course - Process



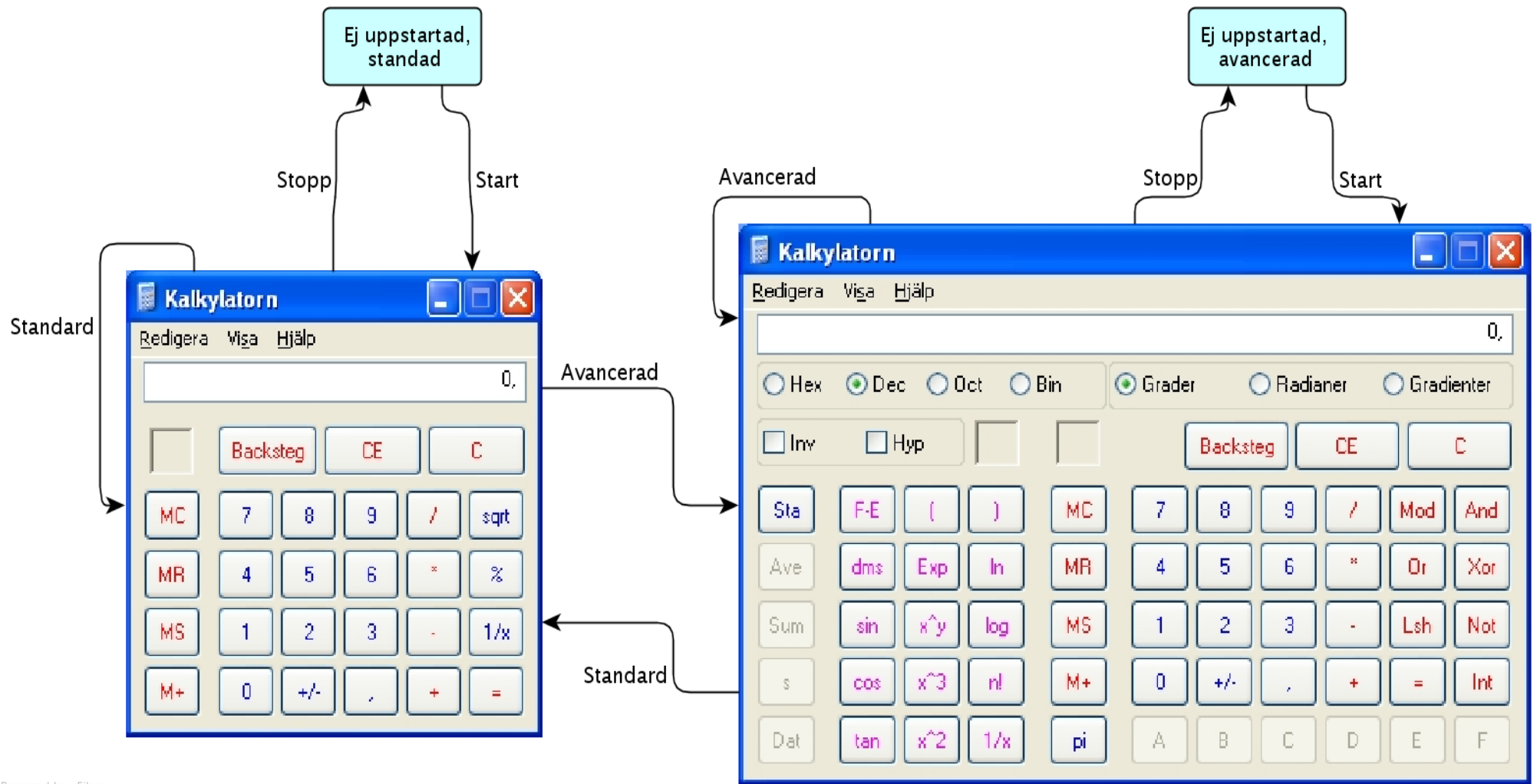
MBT Crash course - Process



MBT Crash course - Design



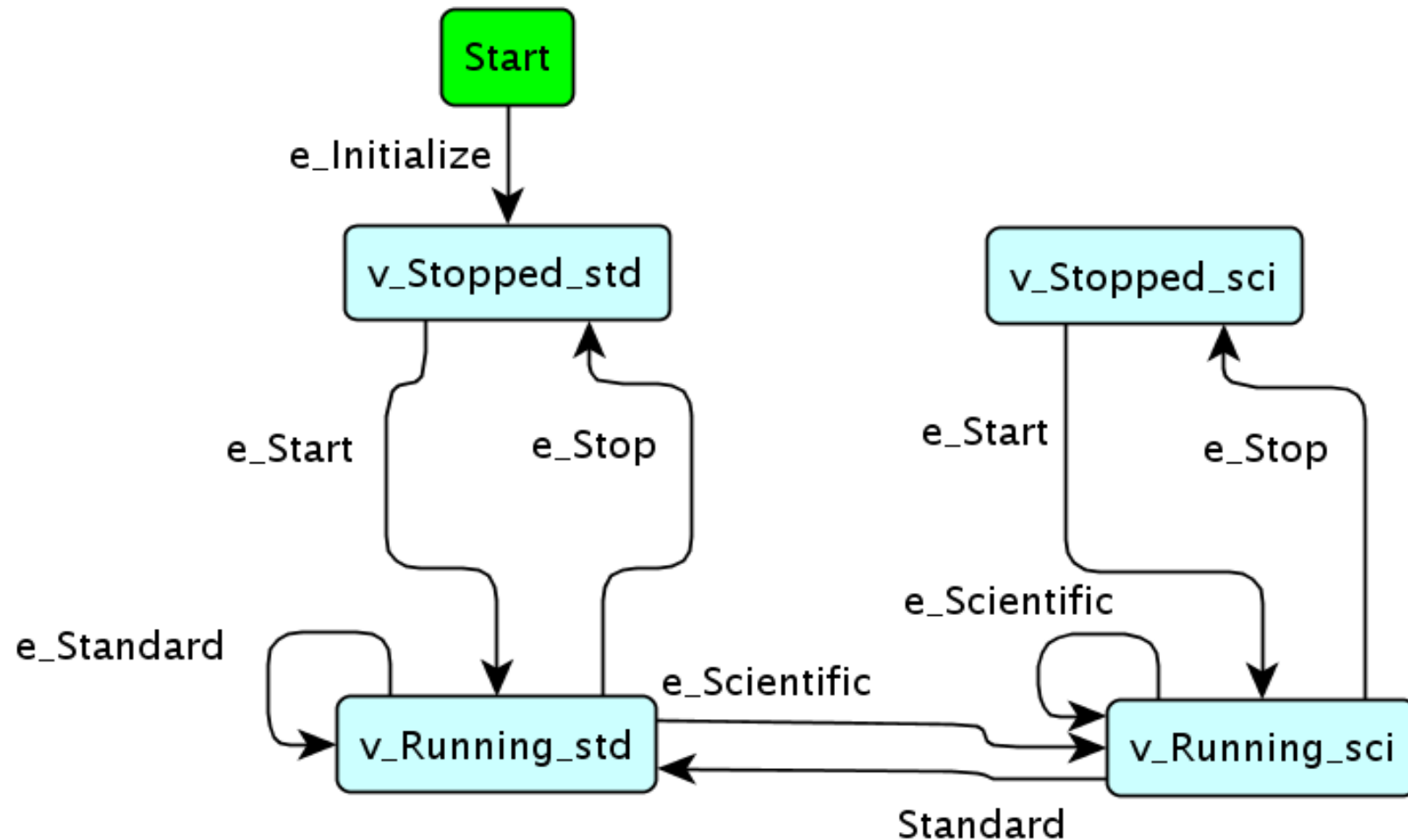
MBT Crash course - Design



Powered by yFiles



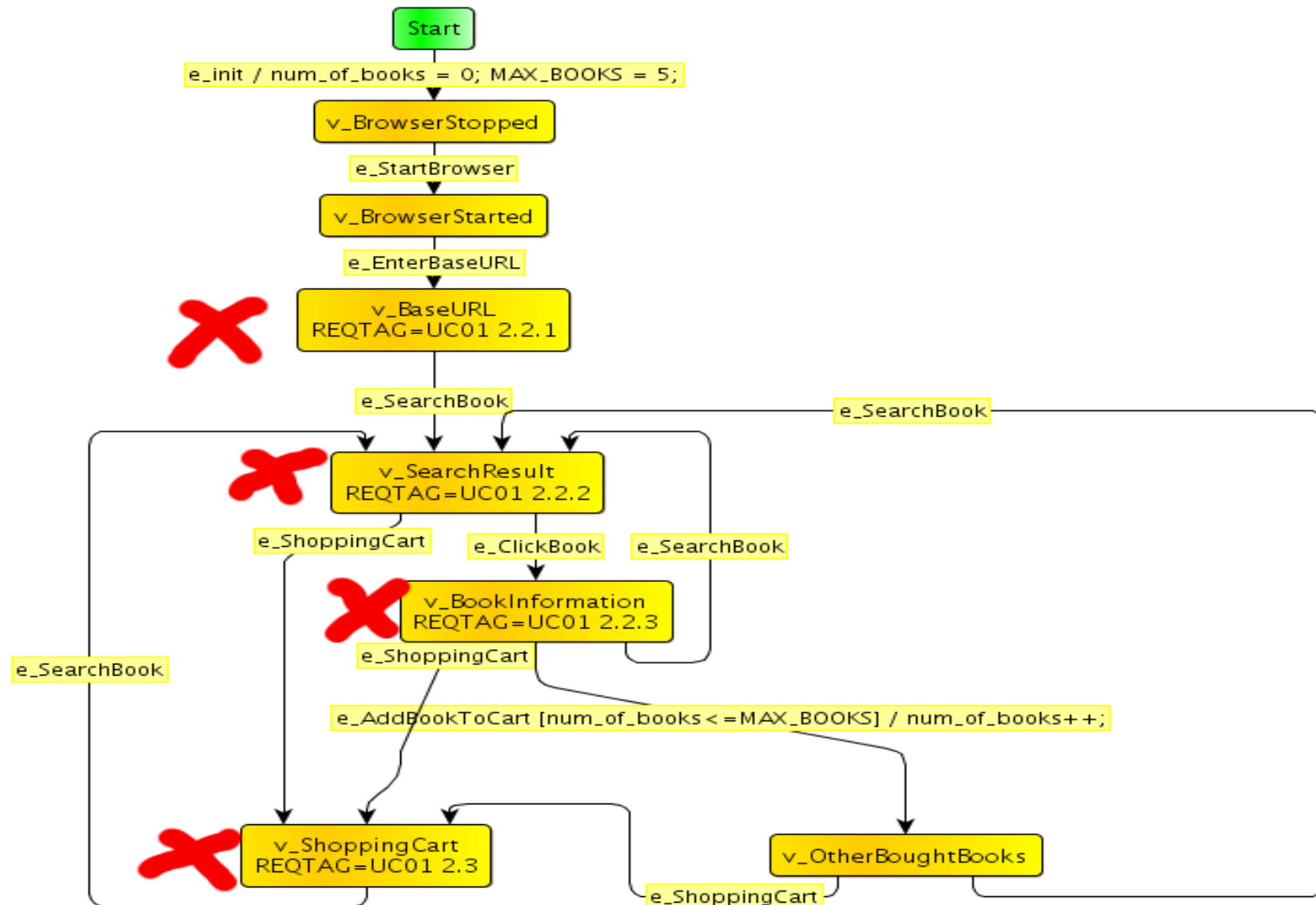
MBT Crash course - Design



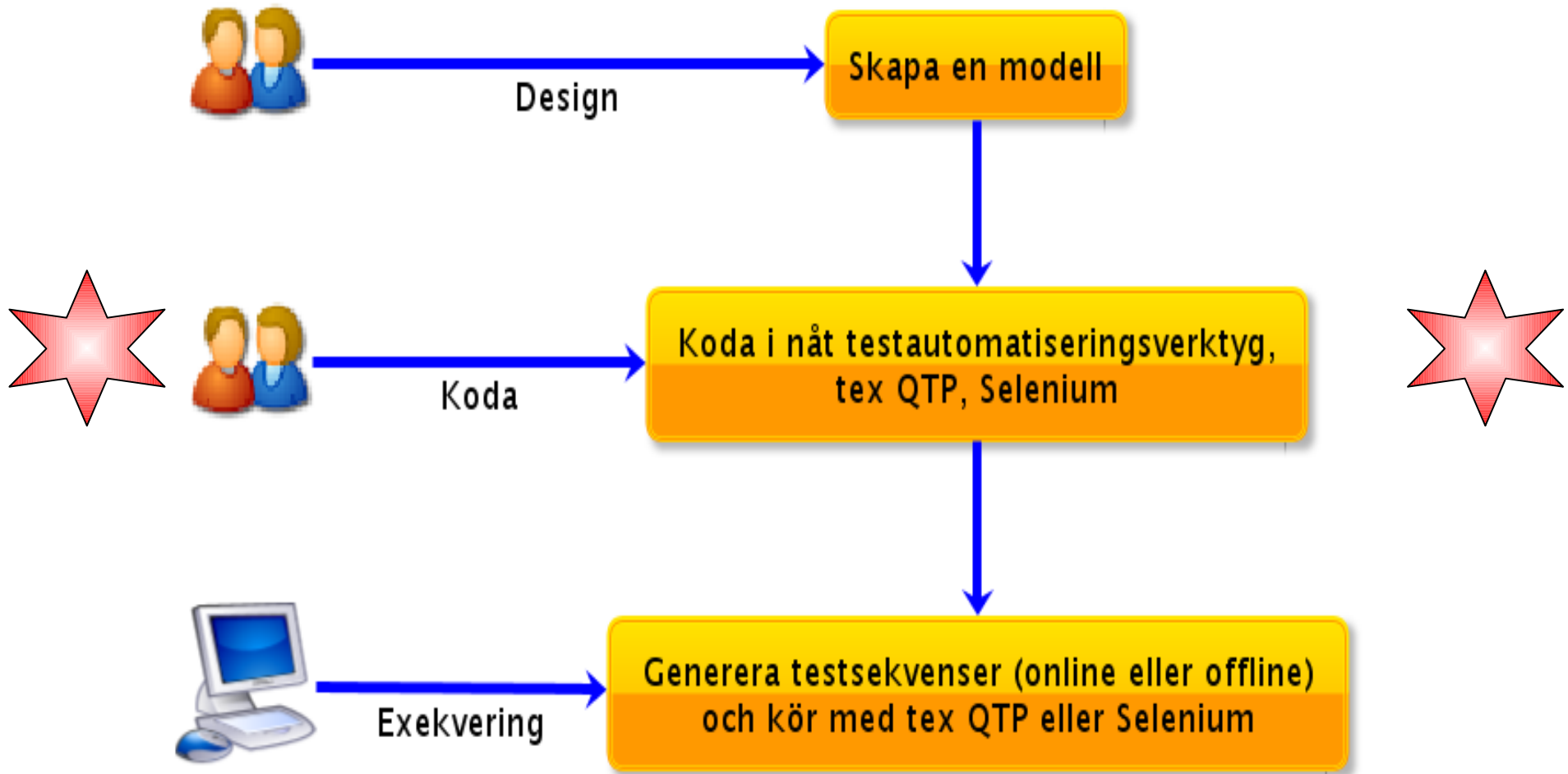
Powered by yFiles



MBT Crash course - Design



MBT Crash course - Process

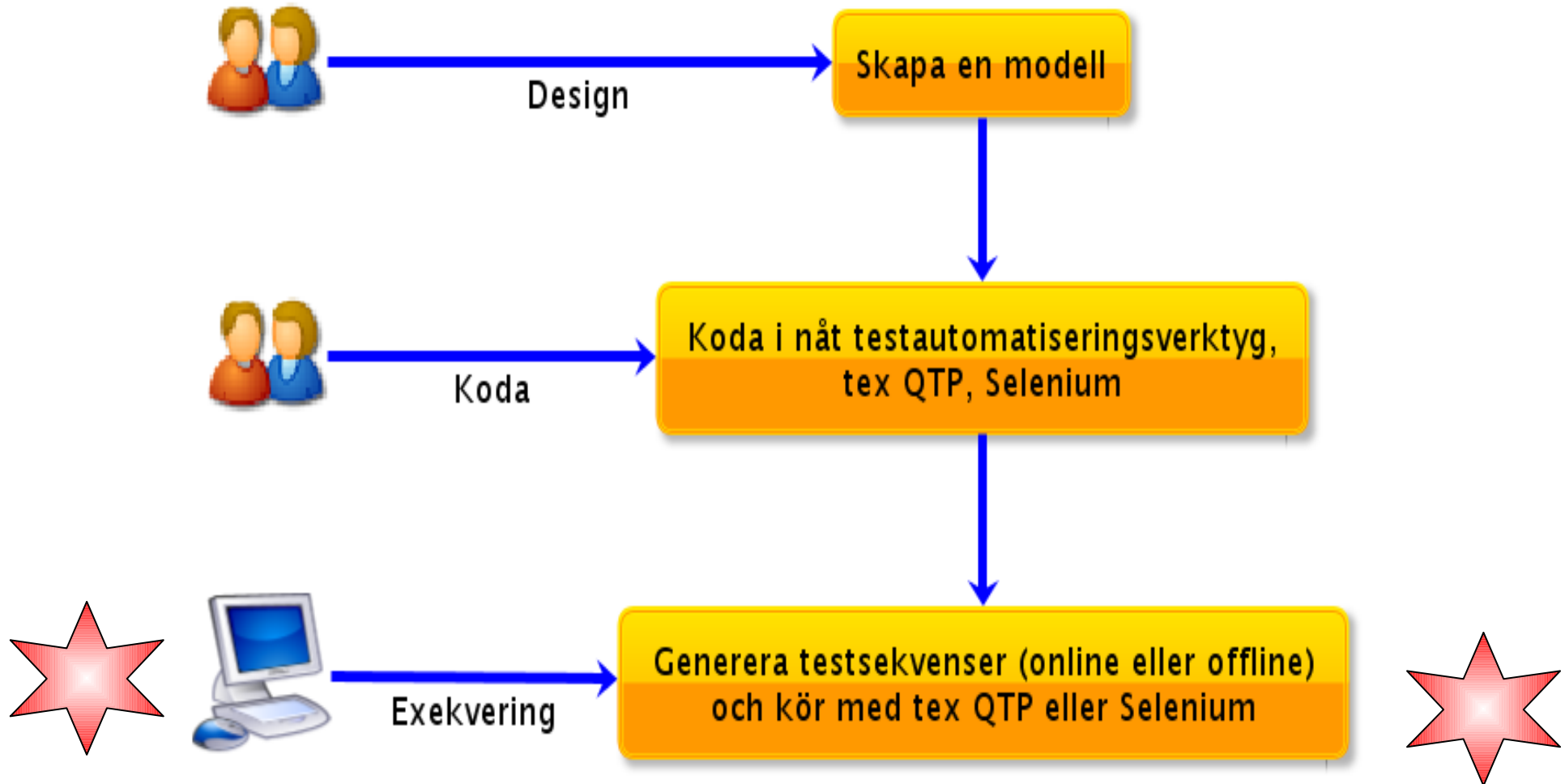


MBT Crash course - Koda

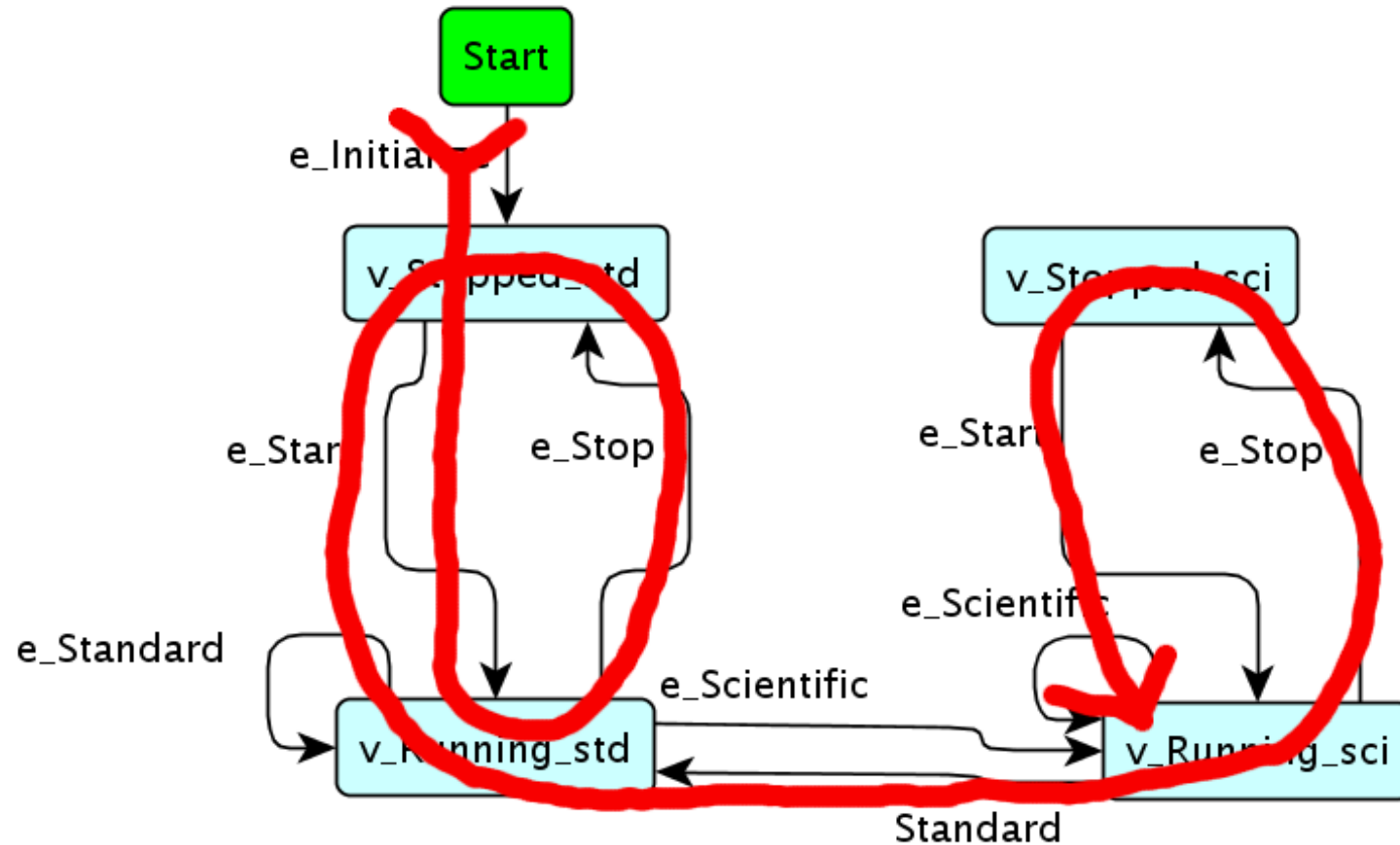
Exempel på funktion **e_Scientific** implementerad i QTP

```
'@Description Implements the Edge 'e_Scientific'  
Public Function e_Scientific()  
  
    Window("Calculator").WinMenu("Menu").Select  
    "View;Scientific"  
  
End Function
```

MBT Crash course - Process



MBT Crash course - Exekvering



Powered by yFiles

MBT Crash course - Exekvering

Generering av testfallssekvenser

Generering

- Slumpat (Fyllegång)
- Optimerat

Stopkriteria

- Samtliga övergångar
- Tid
- Längd
- Specifikt tillstånd
- Specifikt övergång

MBT Crash course - Exekvering

Offline vs. Online

Offline (statisk)

Testsekvensen genereras före testet.

- Samma sekvens varje test
- Lätta att implementera

Online (Dynamisk)

Testsekvensen genereras vid varje test

- Unik sekvens för varje test
- Skapar bättre testtäckning

MBT Crash course - Exekvering

Testfallssekvens

```
e_Initialize  
v_Stopped_std  
e_Start  
v_Running_std  
e_Scientific  
v_Running_sci  
e_Stop  
v_Stopped_sci  
e_Start  
v_Running_sci
```



MBT på PPM

1(4)

Design av tillståndsmodeller

- Med hjälp av krav och verksamhetskunskap ritas en tillståndsgraf i yEd (editor)
- Kraven noteras i modellen (för spårbarhet och testrapportering)

MBT på PPM

2(4)

Gemensam granskning av modellerna

- Innan man börjar koda, så granskas modellen.

MBT på PPM

3(4)

Implementering av kod

- Klient byggd i java testas mha QTP
- Tester mot webben tillverkas i Selenium
- Mocka bort beroenden!!!

MBT på PPM

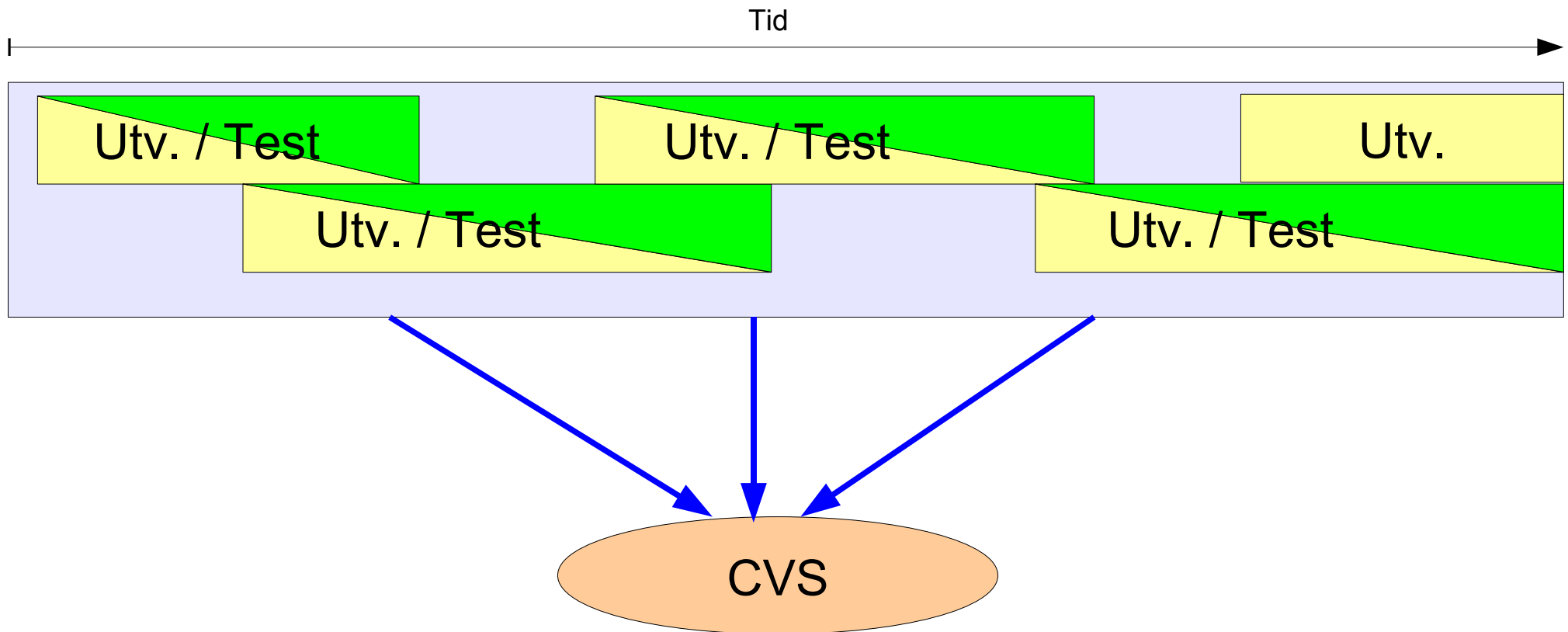
4(4)

Leverans av testerna till Integrationsteamet

- När testerna förväntas fungera i den integrerade testmiljön, så levereras dom
- Testerna anpassas för att köras integrerat, dvs med beroenden av andra system

MBT på PPM

En sprint



MBT på PPM

Nattliga byggen

- Körs varje natt
- Testar allt som är incheckat
- Testerna går i en integrerad miljö (omockad)



Reflektioner från omgivningen

- Natliga byggen exponerar fel tidigt. Detta kan leda till frustration, då man kan tycka att: "Ingenting fungerar!"
- Överflöd av tester. Vadå 238.000 testfall?
- Begreppet 'Planerade testfall' existerar inte
- Test och utveckling har fått en ny relation
- Mätetal! Hur redovisar man resultat på ett vettigt sätt?

Resultat

- 69 felrapporter, på 8 månader, på 10 testare
- Om vi hade skrivit 1.000 felrapporter, och om varje felrapport kostat 2 timmar?
- Många fel löstes enkelt, utan att behöva skriva felrapporter
- Icke-expert designar tester
- Varje natt testas allt:
 - ▶ Enhetstester
 - ▶ Funktionella API-tester
 - ▶ GUI-tester
 - ▶ Prestandatester



Framgångsfaktorer

- En duktig Scrum master
- Experthjälp
- Nyfikna, öppna, intresserade och "villiga" testare
- MBT - Modellerna
- Mockade (stubbade) testmiljöer
- Vid sprintplanering: 1 timme utv = 1 timme test
- Fungerande nattliga byggen och deploy



Resurser

Verktygen

<http://mbt.tigris.org>

<http://www.yworks.com>

Model-Based Testing

http://www.geocities.com/model_based_testing/

<http://www.goldpractices.com/practices/mbt/index.php>

http://en.wikipedia.org/wiki/Model-based_testing



Litteratur och annat

Litteratur

*Model-Based Software Testing and Analysis with C#
by Jonathan Jacky, Margus Veanes, Colin Campbell, Wolfram
Schulte 2008*

*Practical Model-Based Testing: A Tools Approach
by Mark Utting and Bruno Legeard, Morgan-Kaufmann 2006*

Video

Model-Based Testing: Black or White? - Google Talk (Utting)

[http://video.google.com/videoplay?
docid=5521890509476590796&q=mark+utting&total=545&start=
0&num=10&so=0&type=search&plindex=0](http://video.google.com/videoplay?docid=5521890509476590796&q=mark+utting&total=545&start=0&num=10&so=0&type=search&plindex=0)



Frågor

