



Testautomation i agila projekt

Modellbaserad test

Testautomation i agila projekt

- Helena Lind

- Prolore

Process, Organisation och Utbildning inom
Testautomation, Prestandatest och
Testmanagement

Agenda

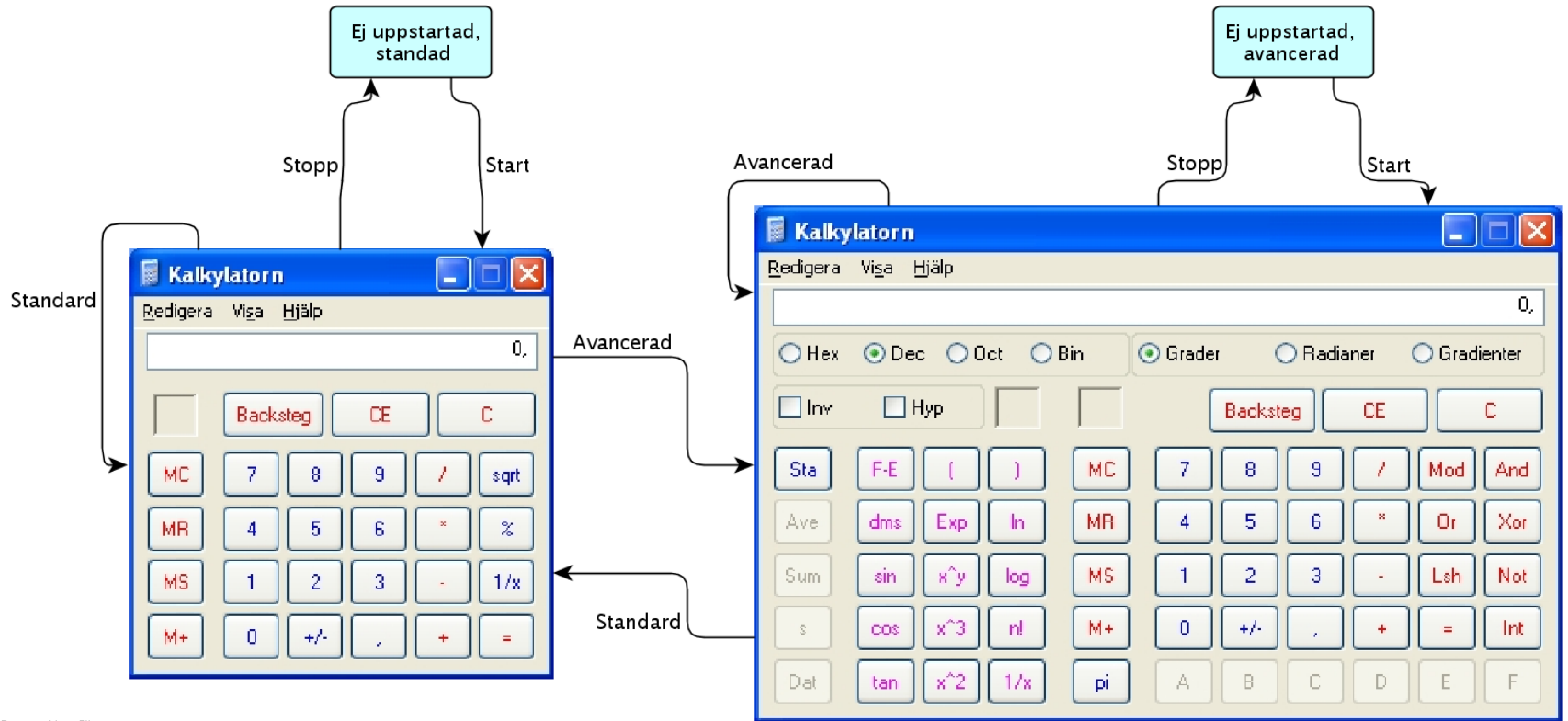
- Vad är MBT?
- MBT vs Traditionell automatisering
- Verifiering
- Presentation av resultat
- Utmaningar och lösningar

Vad är MBT?

Wikipedia:

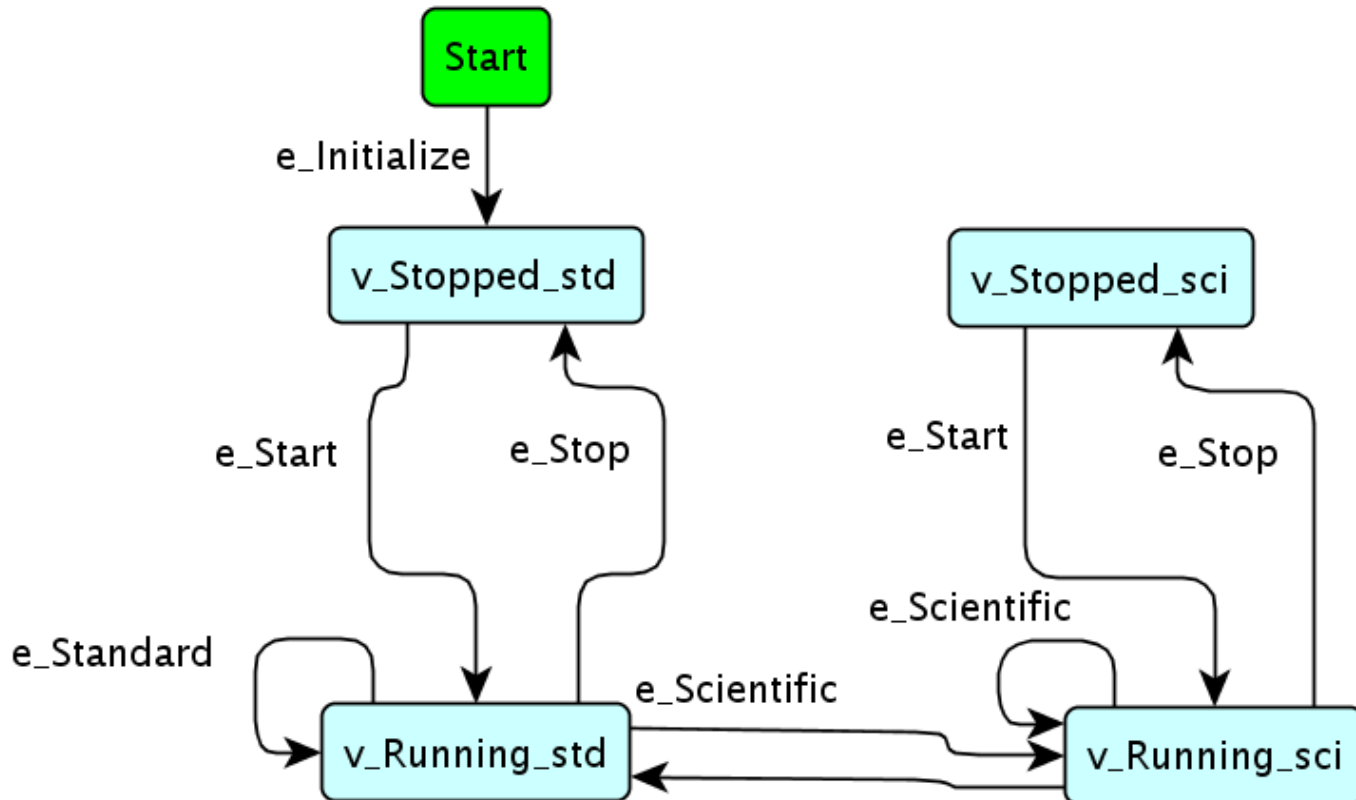
“Model-based testing is software testing in which test cases are derived in whole or in part from a model that describes some (usually functional) aspects of the system under test (SUT).”

Vad är MBT?

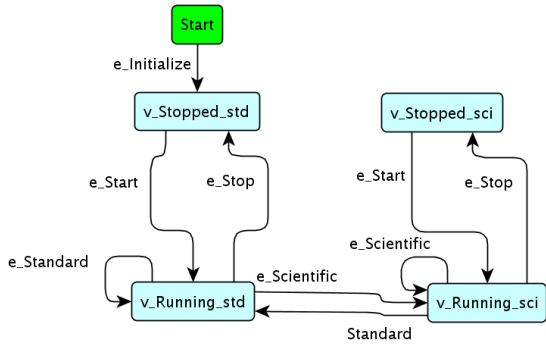


Powered by yfiles

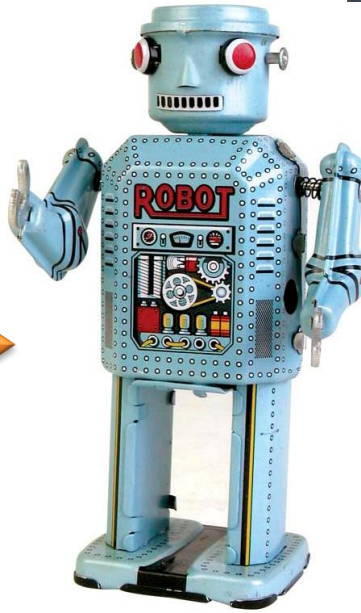
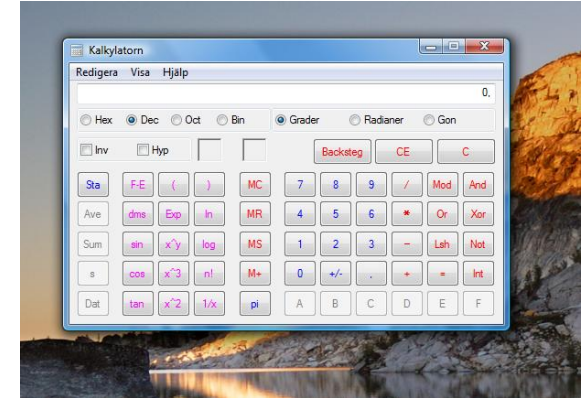
Vad är MBT?



Vad är MBT?

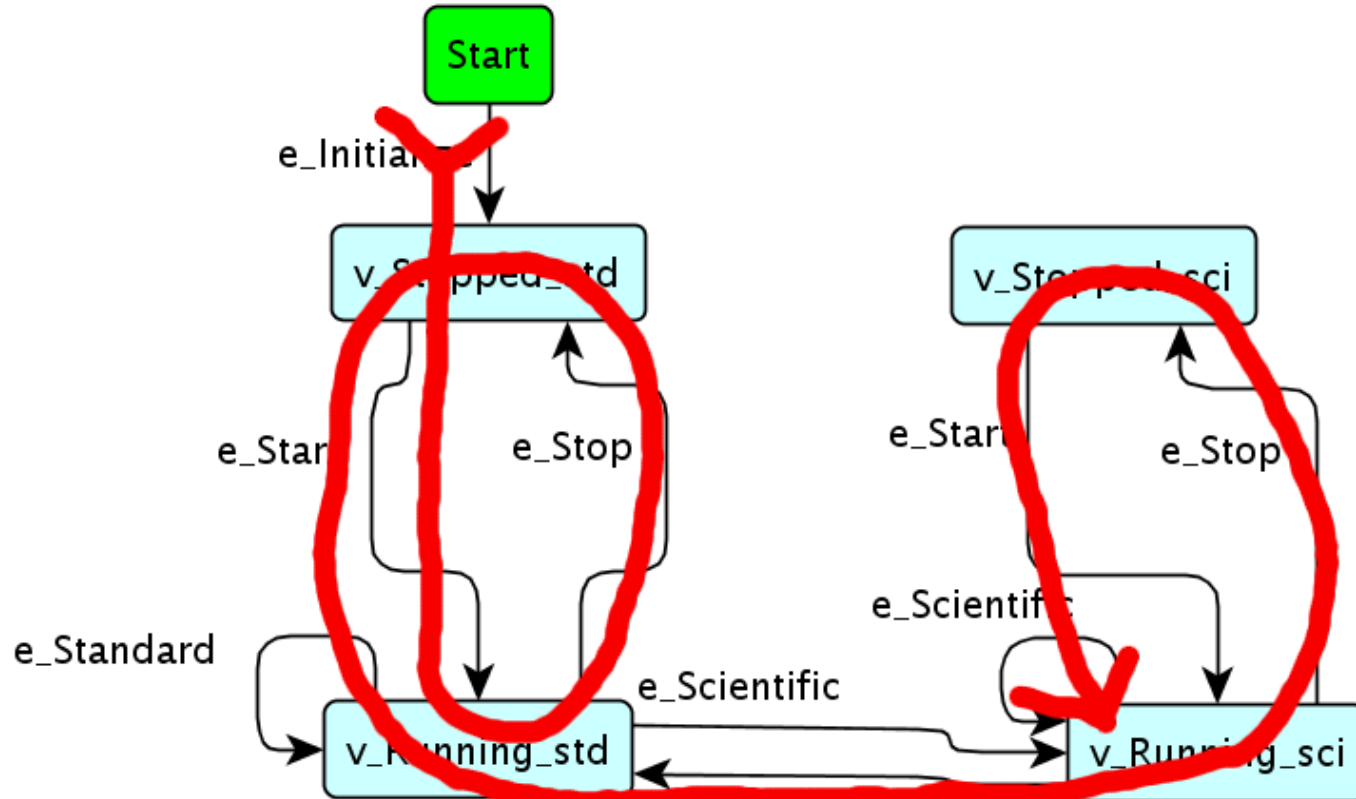


Powered by yFiles

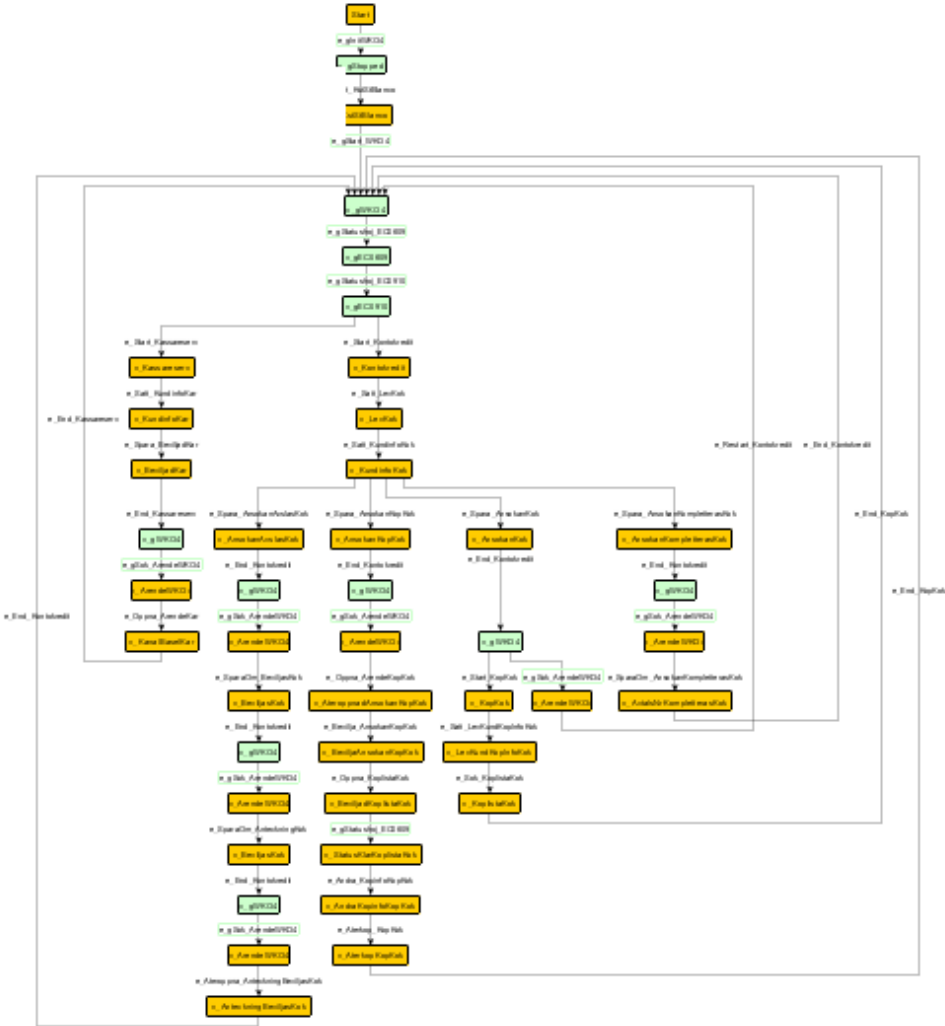


GraphWalker
The Open Source Model-Based Testing Tool

Vad är MBT?



Modellen

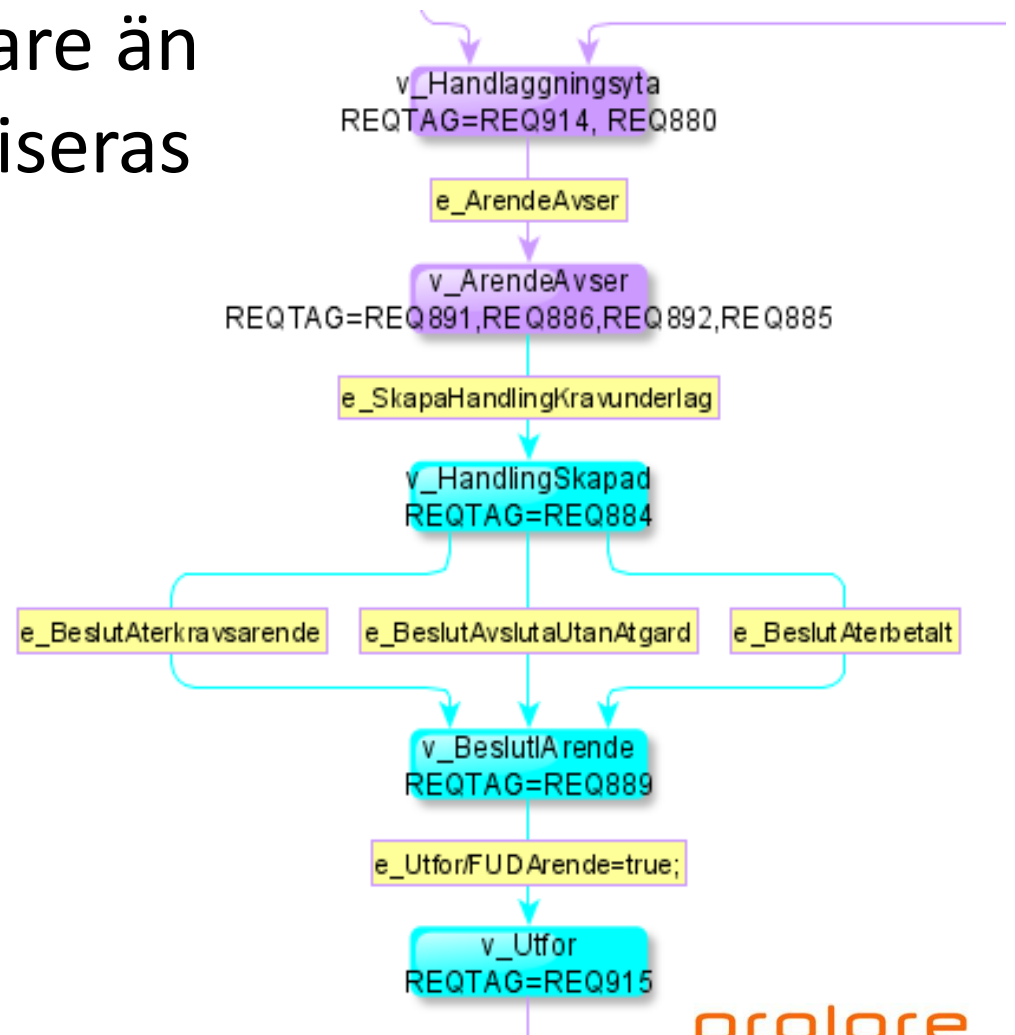


MBT vs traditionell testautomation

- Modellen gör testet visuellt – buggar hittas redan i designstadiet
- Lyfter bort logiken från skripten
- Enklare att underhålla - Lätt att se i modellen
- Dynamiska stopkriterier
- Tid till första testkörningen
- Testar alltid lite olika

Verifiering

- ▣ Krav verifieras snarare än att testfall automatiseras
- ▣ Kraven läggs in i modellen



Presentation

- Resultat efter nattkörningar presenteras i dashboard



The screenshot shows a web browser window with the URL `http://localhost/dashboard/showDashboard.cgi?productId=13`. The browser's address bar and menu (File, Edit, View, History, Bookmarks, Tools, Help) are visible. The main content is a table with the following structure:

Requirement	Status	9	8	7	6	5	4
REQ804 P1.1 Värdeförändring - TVA Tidsviktad avkastning		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat
REQ805 P1.2 Värdeförändring - IRR Kapitalviktad avkastning		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat
REQ806 P1.3 Värdeförändring - Ansaffningsvärde		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat
REQ807 P1.4 Värdeförändring - Värdeförändring i kronor - På kontonivå, hittills i år		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat
REQ808 P1.5 Värdeförändring - Värdeförändring i kronor på kontonivå, totalt		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat
REQ809 P1.6 Värdeförändring - Värdeförändring i kronor på fiktiv fond-nivå, hittills i år och totalt		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat
REQ810 P1.7 Värdeförändring - Fondandelar		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat
REQ811 P1.8 Värdeförändring - Fondkurser		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat
REQ812 P1.9 Värdeförändring - Fondavgifter		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat
REQ813 P3.1 Pluto Enterprise - Filken P-sparare/Fondval		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat
REQ814 P3.2 Pluto Enterprise - Filken P-sparare/Sakbo		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat
REQ815 P3.3 Pluto Enterprise - Filken P-sparare/Sakbo		Testresultat	Testresultat	Testresultat	Testresultat	Testresultat	Testresultat

Utmaningar och lösningar

Utmaning 1:

- Lämna testfallstänk och tänk 'modell'

Lösning:

- Inställning – tänk 'krav'

Utmaningar och lösningar

Utmaning 2:

- Designa relevanta modeller

Lösning:

- Modellerna granskas av test/utv/PO
- Lätt att få feedback

Utmaningar och lösningar

Utmaning 3:

- Tidig testautomation
- Ofta ändringar

Lösning:

- Tätt samarbete inom och mellan scrumteam
- Prototyper från utvecklare
- Stöd vid modellering

Utmaningar och lösningar

Utmaning 4:

- Agil testautomation (scrum)

Lösning:

- MBT lätt att ändra/underhålla/förstå
- Förlåtande om man gör fel
- Återanvändning av kod
- Testdesign från dag 1

Utmaningar och lösningar

Utmaning 5:

- Hur undvika dubbelarbete och få erfarenhetsutbyte mellan testautomatiserare i flera scrumteam

Lösning:

- Testautomationmöten med Kanban 1ggn/v
 - Krav, Design, Realisering, Exekvering, Integration, Förvaltning

Utmaningar och lösningar

Utmaning 6:

- Testmiljöer

Lösning:

- Egen testmiljö för utv av skript
- Deploya efter eget behov
- Integrationstestmiljö för nattekörningar

Utmaningar och lösningar

Utmaning 7:

- Underhåll av skript

Lösning:

- Prio 1 - Underhåll
- Dynamiskt testdata

Utmaningar och lösningar

Utmaning 8:

- Daily builds - Nattekörningar

Lösning:

- Integrationstestteam – miljö och testexekvering
- Miljön är omockad för att hantera integration mellan system
- Teamen läser/analyserar själva resultaten

Utmaningar och lösningar

Utmaning 9:

- Visualisera produktstatus och kravtäckning

Lösning:

- Dashboard - utifrån testresultat gentemot krav

Summering

- Visualisera testerna via modell
- Testtäckning – stoppkriteria
- Verifierar krav
- Samarbete inom teamet/projektet möjliggör 'agil takt'
- Snabbt till första testkörning
- Lätt att underhålla
- Testkörningar varje natt ger tidiga testresultat

Q&A



presentationen finns på www.prolore.se