

Testa risker tidigt Upp till bevis för test

© Copyright Prolore 2007. All Rights Reserved.

Vem är jag?

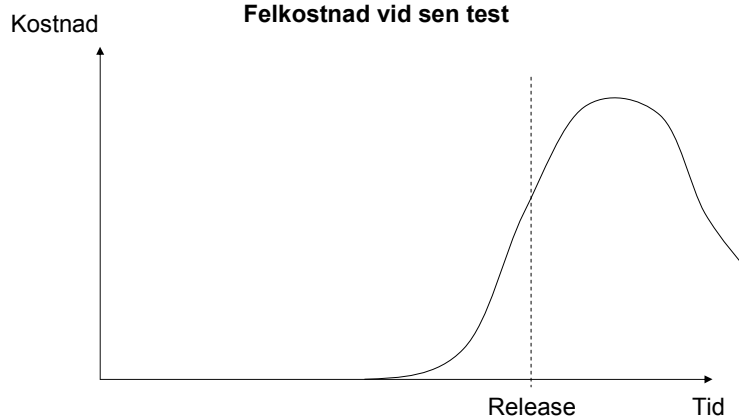
- Stefan Rungardt
- Sedan 1993 i test
- Teknisk testledning och testautomatisering
- Prolore, testautomatisering & prestandatest...

Testa tidigt

- Riskbaserad test handlar väldigt mycket om att testa kritisk funktionalitet eller arkitektur mycket och **tidigt**
- Test har i alla tider predikat om vikten av att testa tidigt
- Men vad gör vi i test när vi faktiskt får chansen?
- Att misslyckas med tidig test är dyrt

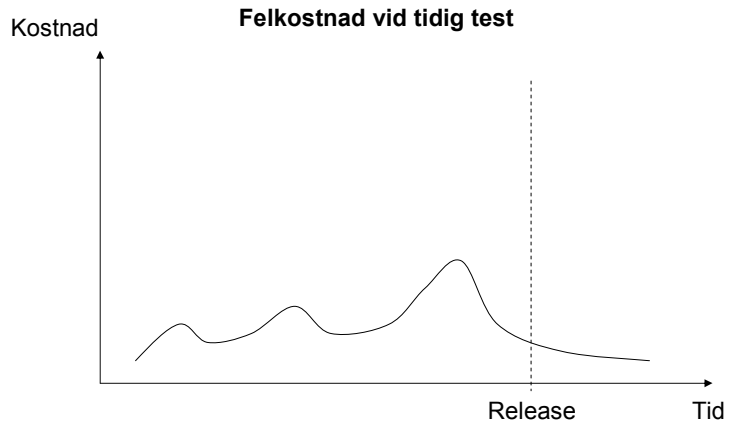
prolore

Testa tidigt



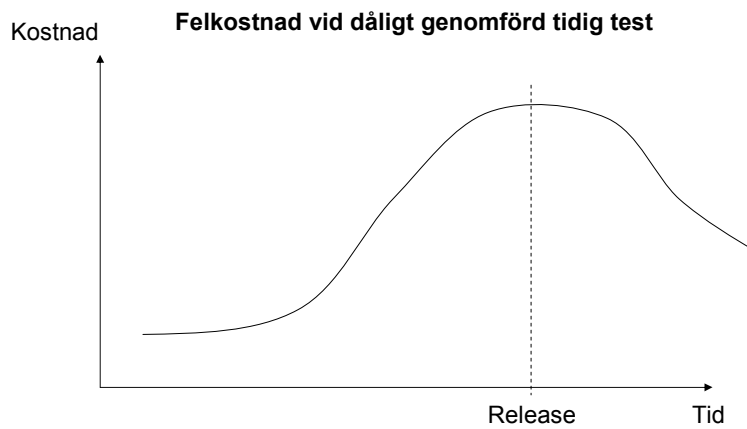
prolore

Testa tidigt



prolore

Testa tidigt



prolore

Agenda

- Varför är det svårt?
- Avgränsningar
 - Hur får man något att testa?
 - Hur testar man det?
- Praktikfall
- Skapa förutsättningar
- Hantering av brister i förutsättningar
- Framgångsfaktorer

prolore

Varför är det svårt?

- Ofärdiga lösningar
- Ofärdiga krav
- Föränderliga krav och lösningar
- Metoder och processer som ej har satt sig
- Infrastruktur ska byggas upp
- Kompetens skall säkras

prolore

Testledarens roll

Finns det optimala projektet?

prolore

Testledarens roll

Istället för att förutsätta att förutsättningar ges:

- Inventera förutsättningar
- Påtala brister och konsekvenser
- Välj strategi utifrån förutsättningar
- Kontinuerligt förbättringsarbete

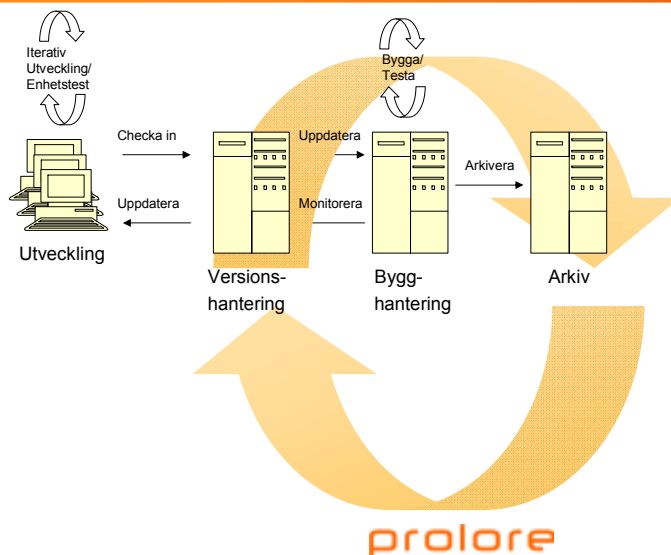
prolore

Kontinuerlig Integration

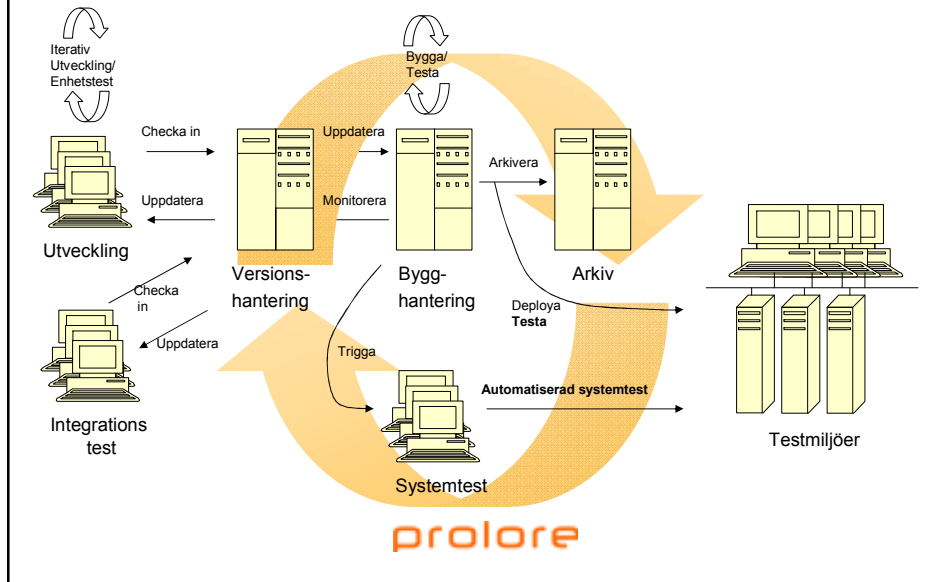
- Ju tidigare test desto mer regressionstest
- Ju mer regressionstest desto bättre förutsättningar för testautomatisering att bli lönsamt
- Automatiserad leveransprocess ger bättre bättre förutsättningar att få något att testa...som är testbart
- Kontinuerlig Integration

prolore

Kontinuerlig Integration



Kontinuerlig Integration



Praktikfall

- Avrapportering och utredning av händelser och brott
- Ett av de största utvecklingsprojekten Rikspolisstyrelsen har haft
- Bra krav på verksamhetsnivå + krav på 2 generella tjänster, Regeltjänsten och Persontjänsten.

prolore

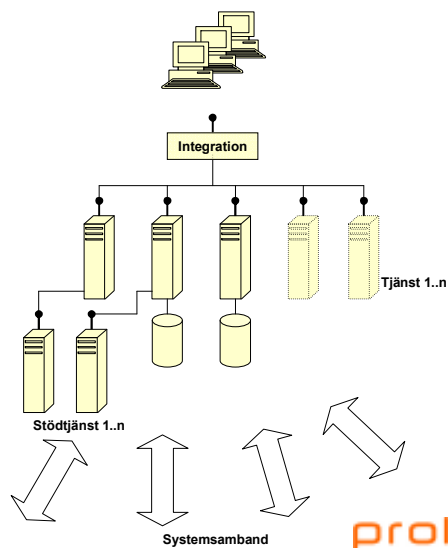
Tjänstebaserad arkitektur SOA

En tjänst har:

- En väl avgränsad funktionalitet
- En tydlig koppling till verksamhetsnytta
- Ett standardiserat gränssnitt
 - Typiskt webbtjänster baserade på XML

prolore

Praktikfall



Teknik:

J2EE
WebServices
Java RMI
BEA WebLogic
Oracle
HPUX
MQ

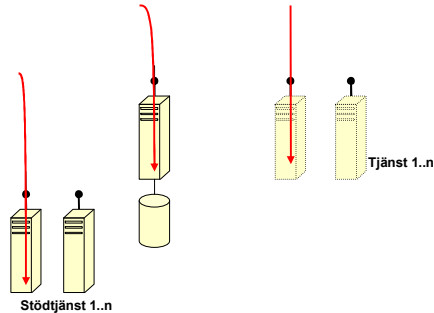
Projektmetodik:

RUP

prolore

Praktikfall

Teknik
•Java RMI



Praktikfall

- Hur systemtestar man en tjänst?
 - Testa utifrån en generell kravbild
 - Utveckla generella testklienter
 - Java RMI -> Java-testklienter
- Ingen Java-kunskap i testdisciplinen
 - Stor uppbackning från utvecklingsdisciplinen
 - Samma teknik som enhetstest. Eclipse och JUnit.
 - Med lite generella programmeringskunskaper så kommer man ganska långt. Testkod kan göras ganska enkel.

prolore

Praktikfall

- Nej, vi hjälper inte till med enhetstest
- Definition av testnivåer
 - Systemtests syfte är att verifiera krav och tester ska utföras i produktionslik miljö
 - Enhetstests syfte är att verifiera design och testerna kan utföras på objektfiler
- Startkriterier för systemtest
 - Kodtäckningsgrad minst 80% på radnivå (EMMA)
 - Leverablerna skall vara körbara i den ordinarie testmiljön
- Prestandatest med LoadRunner

prolore

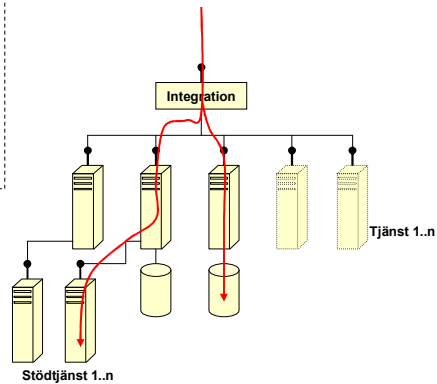
Praktikfall

- 2-veckors iterationer
- Nightly Build
 - Med kodtäckningsmätning. 16% första gången.
 - Med system/integrationstest mot ordinarie testmiljö

prolore

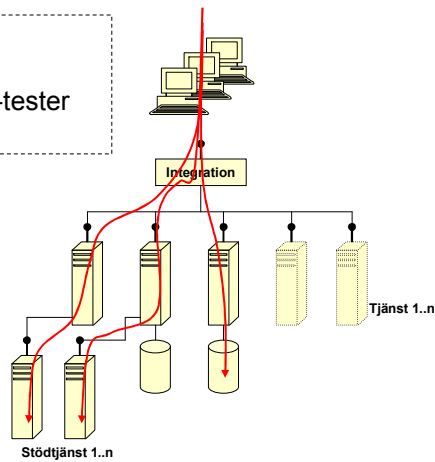
Praktikfall

- Java metदानrop med kravmappning mot verksamhetskrav
- Java RMI -> WebServices
- JUnit
- LoadRunner
- Nightly Build



Praktikfall

- Verifiering av verksamhetskrav
- Automatiserade GUI-tester
- Nightly Build



prolore

Vad lyckades?

- Systemtest från första iterationen
- Tidig prestandatest
- Hög automatiseringsgrad
- Automatiserade tester i produktionslik miljö
- Parallell utveckling av testkod. Frysta krav och API:er levererades tidigt i iterationerna.
- Automatisk kodtäckningsmätning.

prolore

Vad misslyckades?

- Projektet nedlagt
- Rörliga krav, många intressenter
- Rörlig implemetation
- Tillräckligt bra var inte gott nog
- Integrationsplanering kom igång för sent
- Eviga diskussioner om vad test är
 - "Ni ska ju hjälpa till med enhetstesterna"
 - "Nej bry er inte om det. Det finns där av tekniska skäl." Brist på kravfokusering.
- Riskbaserad endast i Test
- Organisationen ej SOA-mogen

prolore

Skapa förutsättningar

- Var befinner vi oss idag?
 - Vilken kompetens har vi?
 - Vilken infrastruktur har vi?
 - Hur jobbar vi med integration och integrationsplanering?
- Var brister det?
- Hur hanterar vi bristerna?
 - Vad klarar vi oss utan?
 - Vad måste vi förbättra?
 - Risk och konsekvensanalys, Business Case

prolore

Förbättringsarbete

- Inventera nuläge
- Identifiera svaga länkar
- Prioritera förbättringsåtgärder
- Vision, Etappmål
- Anpassa mot brister i förutsättningar samtidigt som man jobbar för att skapa förutsättningar
- Ett steg i taget. Tålamod.
- "TPI"

prolore

Ex: Kompetensbrist

Brist i förutsättningar: Utvecklingskompetens i testgruppen

Möjlig åtgärd 1:

Ta in utvecklingskompetens

Möjlig konsekvens 1:

Systemtest blir enhetstest

Möjlig åtgärd 2:

Beställa testklienter

Möjlig konsekvens 2:

Sårbarhet. Tester kan bara utföras om testklienter levereras.

prolore

Kompetens

- Utvecklingskunnande en nödvändighet
- Testautomatisering på alla testnivåer
- Erfaren testledning
 - Förmåga att identifiera förutsättningar, också utanför testdisciplinen
 - Förbättringsarbete
 - Timing

prolore

Ex: Föränderlig värld

Brist i förutsättningar:

Krav och arkitektur förändras

Möjlig åtgärd 1:

Utforskande test

- Specifikation och utförande på samma gång
- Testerfarenheter styr hur testerna utvecklas
- Ställer krav på testarens kreativitet

Möjlig åtgärd 2:

Testa senare

Möjliga konsekvenser?

prolore

Ex: Automatisering

Brist i förutsättningar:

Kompetens eller infrastruktur för automatisering av leverans och byggprocess och test saknas i projektet

Möjlig åtgärd:

Manuell hantering

Möjlig konsekvens:

Längre iterationer (Big Bang integration)

Degenererande regressionstest alt. minskande tid för nya tester

prolore

Ex: Utvecklare

Brist i förutsättningar:

Tunnelseende i utvecklinsdisciplinen

Möjlig åtgärd:

Svårt. Utbilda och tjata. Visa på konsekvenser. Tvinga med hjälp av checklistor och byggmiljö

prolore

Testa sent?

Brist i förutsättningar: Förutsättningar saknas

Möjlig åtgärd: Testa sent

- Med riskmedvetande och riskminimering kan sen test vara bästa lösningen
- Om inte annat så kan man skrämman fram resurser med hotet om att testa sent

prolore

Sammanfattning: Framgångsfaktorer

- Kontinuerlig förbättring
- Automatisering
- Ställ krav på enhetstest, kodtäckningsgrad
- Ta chansen att ifrågasätta

prolore

Frågor?

?

stefan.rungardt@prolore.se

0705-829860

prolore