JigZaw

”Verifiser Forventet Funksjonalitet”

Teststategi utviklet av

Erik Drolshammer
Bård Lind

Bård Lind

Java siden 1997

Arkitekt siden 2000

JavaBin siden 1999

Enterprise Domain Repository

og JigZaw-teststrategi

Arkitekt i Telenor CID/CSS siden 2009

Twitter: baardl

bard.lind@telenor.com

CANTARA

Main components

Agile and Software Architecture
Agile Mindset and Methodology
Strategies
Tactics

Popular Pieces

JigZaw
Enterprise Maven Infrastructure
Installation and Deployment Automation
Maven FAQ
Test Frameworks and Tools
Lightweight alternatives to heavyweight technologies

Resources

Presentation repository
Glossary

Page Operations

View
Edit
Attachments (1)
Zip Upload
Doc Import
Info
Watchers
Favourites

Browse Space

Add Content

KM: Agile Software Development

JigZaw

Added by erik.drolshammer, last edited by Bård Lind on Mar 11, 2010 (view change)
Labels: strategy featured test maven ci EDIT

JigZaw - Agile Testing done Right

Concept

Documentation

- Introduction
- Test Levels
  - Categorization Matrix
- Design Principles and Drivers
- Mapping between requirements and tests
- Timeline - when to run and write tests

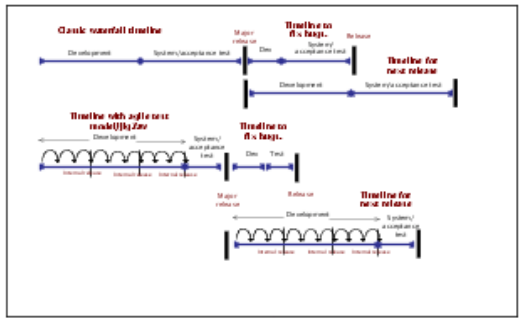
Tools and Implementation

- JigZaw Software Stack
- JigZaw Service Test Tactics
  - JMS Testing according to JigZaw
- System Test - Logical Overview

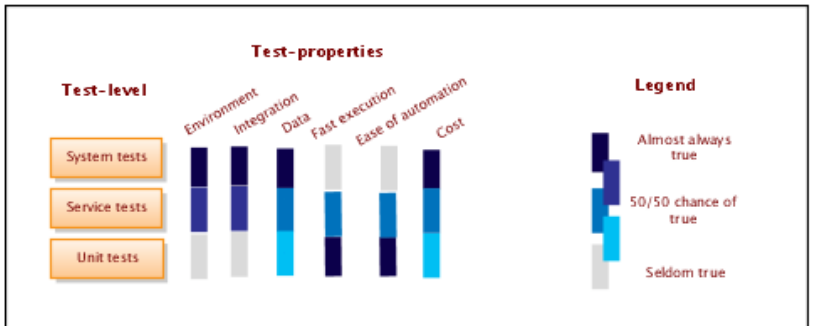
Grouping and Categorizing

- How to use JUnit Categories to implement JigZaw
- How to use TestNG groups to implement JigZaw
  - Group management
  - CI phases
- Standalone application example - not relevant in this form
- Web Application JigZaw - not relevant in this form

Spend less time on integration testing.



Test categorization



http://wiki.cantara.no/display/smidigtonull/JigZaw

Hva vi ønsker å oppnå

•Ultimat mål: kortere tid fra idee til produksjon

•Bruke mindre tid på testing totalt sett.

•Hvordan:

- Splitt og hersk (Divide and Conquer)
- Grupper kan være avhengig av hverandre.
- Verktøy tilgjengelig lar deg teste det du ikke kunne teste før.
- Sum av tester verifiserer forventet funksjonalitet.

Forutsetning

- Enhetstesting med Junit brukes i prosjektet
- CI server brukes i prosjektet
- Automatisk bygg via Maven, eller Ant

- Prosjekt == Produkt, kostnader slutter ikke når 1. leveranse er gjort

Felles språk

- Ende-til-ende test
- Klassisk trelagsarkitektur
- Unit vs manuell/automatisert-manuell test.
- Stub og Mock er vanlig
- Verifiser Forventet Funksjonalitet
- JigZaw i Telenor Terminologi
  - Hva er en Integrasjonstest?
  - Systemtest?, SI Test? VKT?

Når passer JigZaw

- Remote-centric arkitektur.
- Standard webapplikasjon.
- Enterprise testing.

Teori

A. Oppdeling i Ansvar
   - Gi hver test ett eksplisitt ansvar

B. Grupper testene dine

C. Timeline

•Summen av tester Verifiserer Forventet Funksjonalitet.

A. Oppdeling i Ansvar

- Symptomer på trøbbel i horisonten.
- DP1. Single Responsibility Principle
- DP2. Divide and Conquer

Symptomer på trøbbel…

A. Oppdeling i Ansvar

- Symptomer
- DP1. Single Responsibillity Principle
- DP2. Divide and Conquer

- Komplekst prosjekt.
- Vanskelig å skrive gode tester.
- Tungt å vedlikeholde tester.
  - Utvikling av funksjonell kode hemmes av testendringer.
- Testing tar tid.
- Krøkete å kjøre tester til ”grønt”.

- Vanskelig å diskutere tester.

A. Oppdeling i Ansvar

- Symptomer
- DP1. Single Responsibillity Principle
- DP2. Divide and Conquer

Single Responsibillity Principle

- Test kun en funksjon om gangen.
- En test skal kun ha en grunn til å endres.
- Minimer overlapp mellom tester.
- Eksempel

A. Oppdeling i Ansvar

•Symptomer

•DP1. Single Responsibillity Principle

•DP2. Divide and Conquer

Divide and Conquer

•Del opp ansvar i enkle blokker

•Når noe er vanskelig, del opp, løs kjerneproblemet.

•Tester fra GUI til DB vil feile!

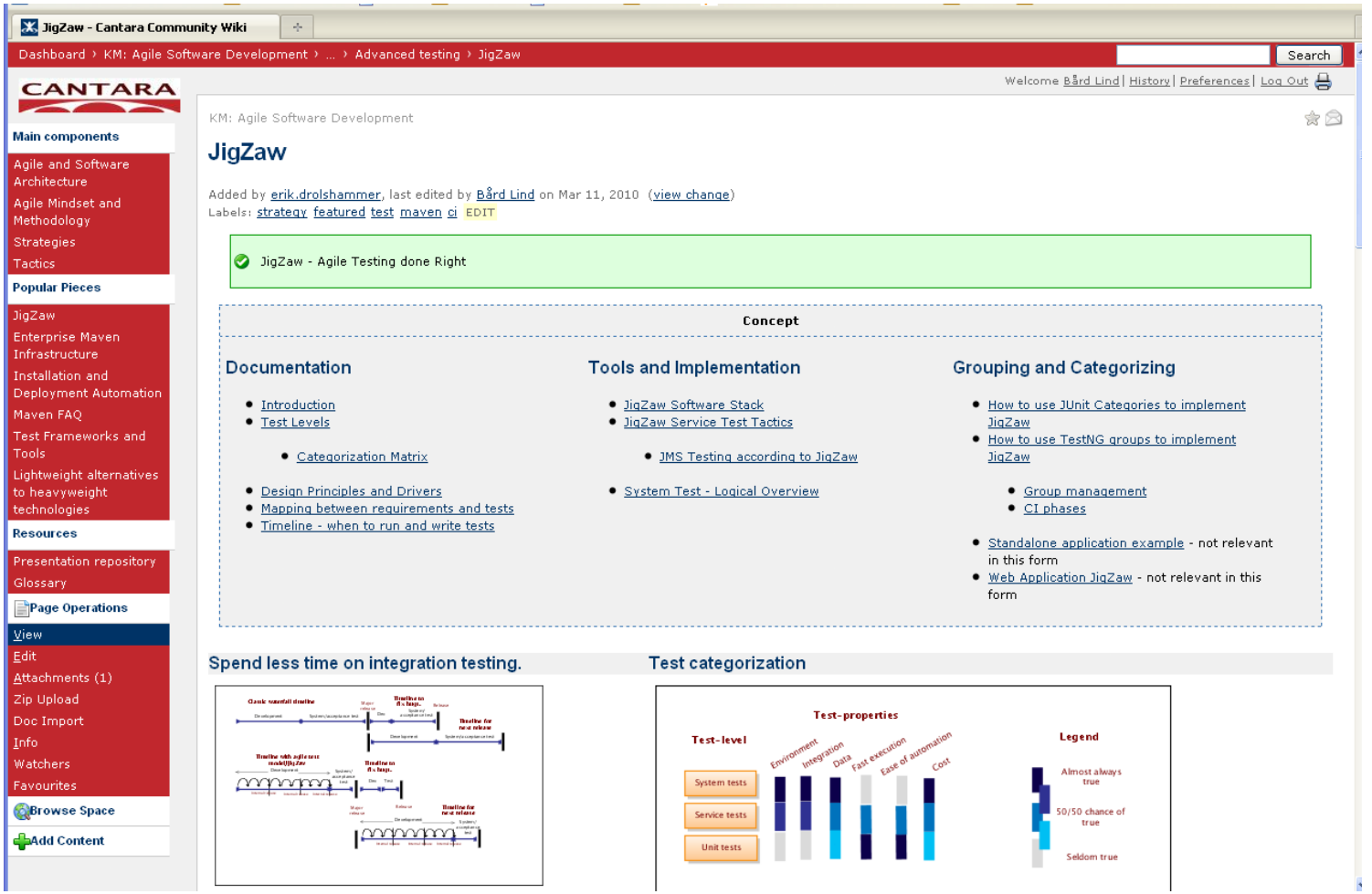
B. Grupper testene dine

- DP3. Grupper tester eksplisitt
- Forslag til grupper.

DP3. Grupper tester eksplisitt

- Basert på forutsetninger
  - Ressurser testen trenger
  - Tid det tar å kjøre testen
  - Datasett nødvendig for å sette pre/post conditions.
- Samme test kan være medlem av flere grupper.
- Test-dependency
  - En test kan være avhengig av en annen.

Gruppeeksempler



http://wiki.cantara.no/display/smidigtonull/JigZaw

Egenskaper å gruppere etter

Full isolasjon	Flere noder?	Med data
Uten Data	Med miljø	Uten miljø
Rask test	Treg test	Manuell

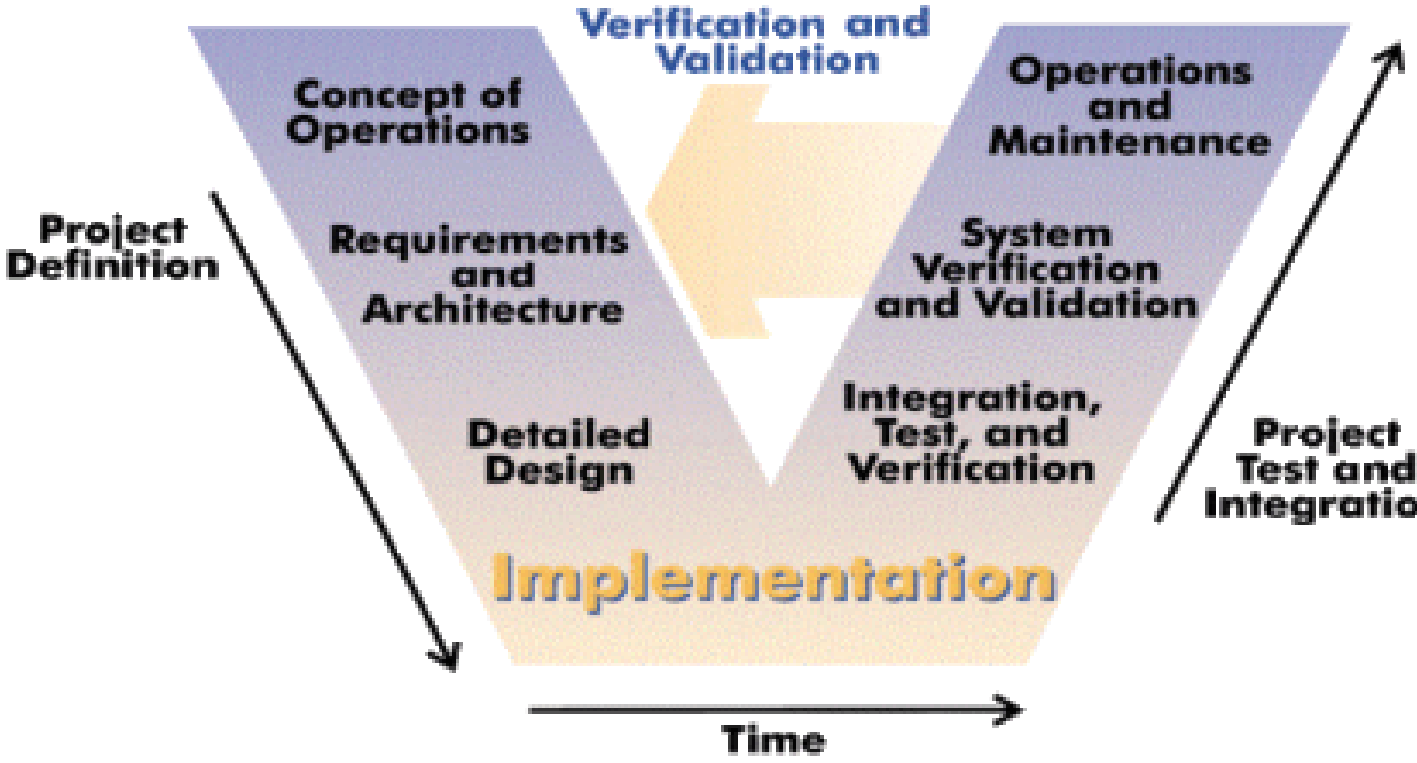
Eksempel testgrupper

Enhetstest (Unit)	Service test	Multi-service test	System test	System multi-node test
• Full isolasjon • Enkelte metoder	• Public tjeneste • Kode nær • White box (åpent innhold)	• Flere prosesser • En node	• Funksjonelle krav • Ikke-funksjonelle krav • Black box (lukket kode)	• Flere fysiske noder

C. Timeline

- DP6. Timeline
- Eksempel
- Performance tester

V-modellen test-nivåer



- Acceptance test
- System test
- Integration test
- Unti test

Ref. http://en.wikipedia.org/wiki/V-Model_(software_development)

Timeline eksempel

- Før Commit
- I sprinten
- Ved sprint demo/avsluttning
- Før produksjonsetting

Når kjøre hvilke tester?

	Før commit	10 min	Hver time	Daglig	Ukentlig	End-of-sprint	Systems integration	Accept-ance
Med Data		X	X	X				
Uten data	X	X	X	X				
Med Miljø		X	X	X				
Uten Miljø	X	X	X	X				
Rask test		X	X	X				
Treg test		X	X	X				
Manuell				X	X	X		
Load-test		?	?	?	?	X		

Test-properties vs Timeline

Verifiser Forvenet Funksjonalitet

A. Oppdeling i Ansvar
   - Gi hver test ett eksplisitt ansvar

B. Grupper testene dine

C. Timeline

Summen av tester Verifiserer Forventet Funksjonalitet.

Designprinsipper

- DP1. Single Responsibility Principle
- DP2. Divide and Conquer
- DP3. Grupper tester eksplisitt
- DP4. Velg den testmetode som gir lavest kost i levetiden til prosjektet.
- DP5. Test og testrapporter har forskjellig publikum.
- DP6. Timeline

Kommunikasjon

- Forbedret kommunikasjon i teamet.

Ukjent med applikasjonen

- Ny på teamet
- Driftsperspektiv

- Bruk mindre tid på å oppdage feilen.
- Bruk mindre tid på å finne årsaken.

Key take-aways

- Splitt og hersk (Divide and Conquer)
- Grupper kan være avhengig av hverandre.
- Verktøy tilgjengelig lar deg teste det du ikke kunne teste før.

Mer Info

http://wiki.cantara.no/display/smidigtonull/JigZaw

erik.drolshammer@gmail.com

bard.lind@telenor.com, twitter: baardl

Cantara

- *Mature and empower Norwegian software development.*-

Cantara er et åpent forum for deling av kompetanse på tvers av firmarelasjoner.