





**THOMAS PLAYS AROUND WITH
POWERPOINT 2007 AGAIN!**



This is Theme "*Metro*"

**THOMAS PLAYS AROUND WITH
POWERPOINT 2007 AGAIN!**



← What the heck is that bar over there?



**THOMAS PLAYS AROUND WITH
POWERPOINT 2007 AGAIN!**



Design -> Far right -> Check Hide Background Graphics

TIP O' THE DAY!

Voila!

TIP O' THE DAY!

Thomas Ferris Nicolaisen (Objectware) presents

OPEN SOURCE TESTING STUFF

- i. Some real-life besserwissening
- ii. Technology 1: Selenium for web-testing
- iii. Technology 2: Hudson for Continuous Integration
- iv. Testing Strategies (The Big Picture)

MORE SPECIFICALLY

thomafni

MSc from ifi spring 2006

inf5750 spring 2005

Blog here: <http://tfnico.blogspot.com>

Mail here: tfnico <at> gmail.com

MORE BOUT ME



```
hisp.info
svn co http://www.hisp.info/svn/trunk/dhis-2/
mvn install
building...
...lotsa deps
tests are running..
... and running...
```

oi, a test failure! :

Failed tests:

```
testExport(org.hisp.dhis.importexport.ixf.IXFExportServiceTest)
```

```
jaja, -Dmaven.test.skip=true
```

still on webwork 2.2.6? tsk, tsk, well so are we anyway

mvn eclipse:eclipse and import into Eclipse..

Not too many warnings. Nice project structure. Good root pom.

MORE BOUT YOU

- Active mailing lists
- Improved Wiki
- Code looks nice
- An active Trac
- Demo up!
- An installer that works!

MORE BOUT YOU

In short

MORE BOUT YOU

BEAUTIFUL!

Well done.

MORE BOUT YOU

SO, TIME FOR BUSINESS

SELENIUM

*Disclaimer:

This presentation was originally made for Storebrand developers (in Norwegian).

SELENIUM*

Warmup

- Web developers?
- Tried..
 - JUnit
 - FIT(nesse)
 - Watir
 - Selenium



A trip down memory lane

2005

January: Selenium is released

Quickly gains attention through conferences, articles, etc

A trip down memory lane

2006

October: Storebrand starts using Selenium tests

- *Verifying use-cases*
- *Regression testing*
- *Basis for performance tests*

A trip down memory lane



2007

Selenium development hibernation (only 1 release)

November/December: Storebrand's tests are freshed up

A trip down memory lane

2008

Storebrand still runs Selenium tests

Selenium popularity increases (1.0 beta released)

"Google has 51k Selenium tests running accross 50 different teams"

<http://www.youtube.com/watch?v=EDb8yOM3Vpw>

Why test?

People are slow!

Tests can be automated (and monitored)



Errors in existing code introduced
(regression_tests)

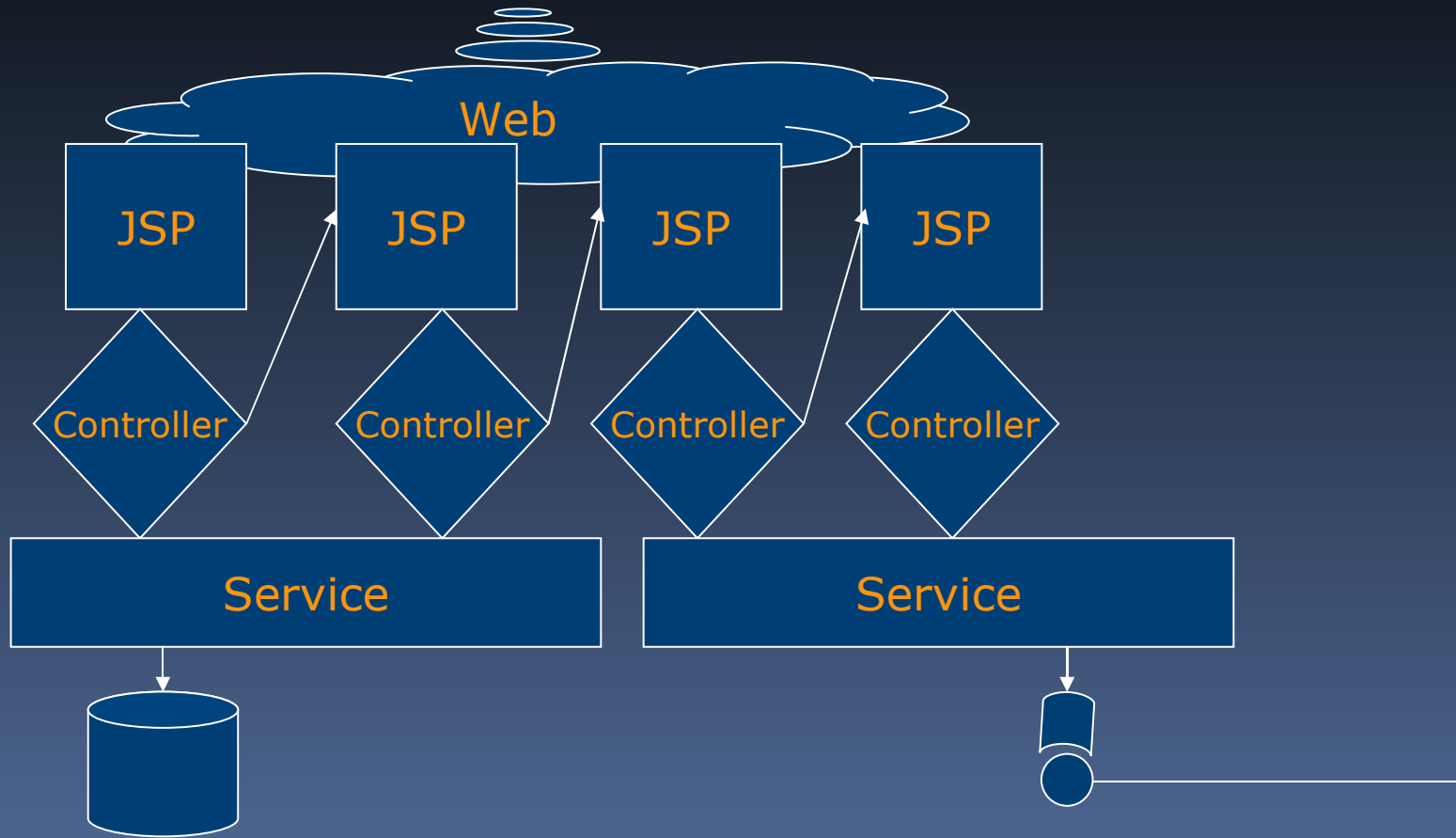
Verify functionality (use-cases)

Handy for debugging

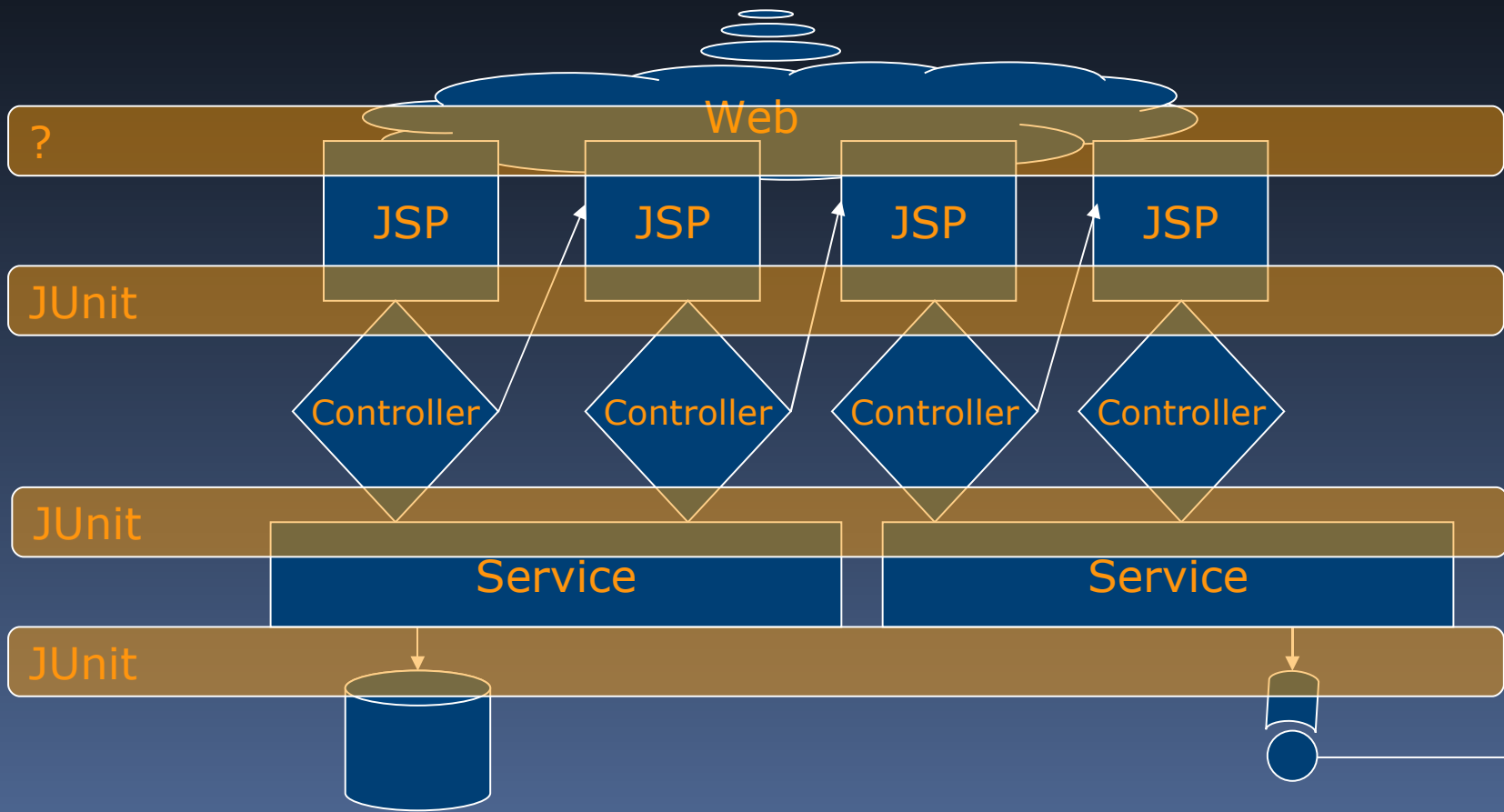
Why test web?

- Some things you can't test with JUnit (easily)
 - JSP
 - HTML/Web-design
 - JavaScript
 - Navigation/urls (struts.xml)
 - Entire scenarios (log in and buy insurance)

Some architecture



Some test architecture



Web-testing tools

	<i>Pros</i>	<i>Cons</i>
Selenium (JS-engine)	Can test JS Good Recorder	Browser dependant
Watir (Ruby)	Powerful language Runs native IE	New language Browser dependant
HtmlUnit (Java)	Mature Virtual browser buffer	No recorder

Selenium

- Selenium IDE (Firefox plugin)
- Selenium Core
- Remote Control
- Grid

Good things about Selenium

- Active community
- Scaringly easy
- Good AJAX/JavaScript support

A normal day in the web-testers life

- Do a quick demo on local DHIS2
- (Or on the demo site)
- Show the screencast

maven-selenium-plugin

- take it for a spin

What can go wrong

- Too much time
- Too much "dill"
- Tests break without anything being wrong (dialog boxes, browser upgrades)
- Too many tests
- Too long, slow tests
- Testing the wrong thing

Remedies

- Use it, and get help
- Good urls, good id's
- Test suites:

20 favourites

Integration

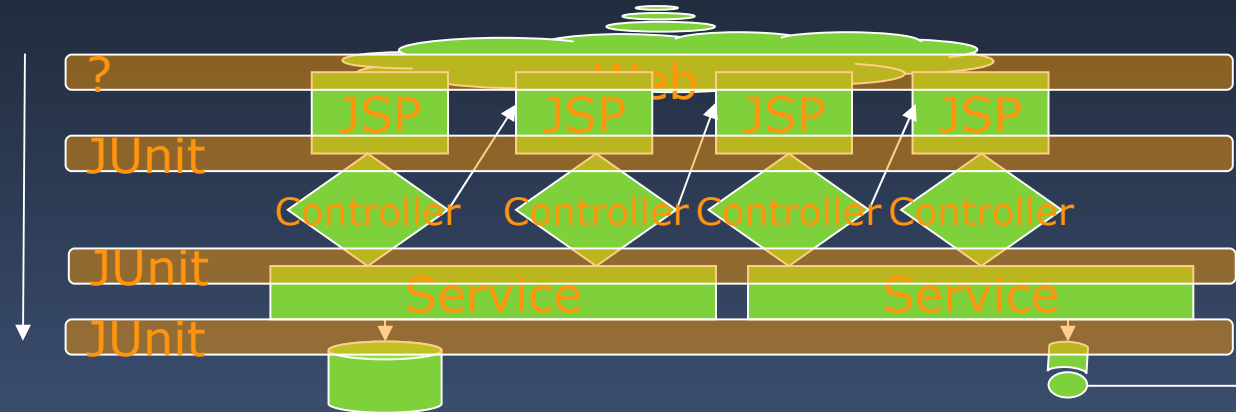
Life

Don't test these

- Performance (unøyaktig måling)
- Data (too static)
- Business rules (imprecise)
- Integration (too slow)

More tips

- Push tests “downward”
- Be pragmatic



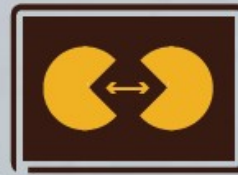
- Don't test everything with Web/GUI-tests
- Have a strategy
- Make bugreporters record traces when they report errors

Testing Strategies

- Selenium? -----
- Fit? -----
- TestNG? -----
- JUnit? -----



Human to Application services



Application to Application services



Aggregated Core services



Core services

Testing Strategies

- Selenium? -----



Human to Application services

	No data	Data	No environment	Environment	Location
Unit test	Fast (1-5)	n/a	Fast (1-5)	n/a	
Service test	Fast (1-5)	Slow (2-5)	Fast (1-5)	Slow (2-5)	
System test	Slow (2-5)	Slow (2-5)	Slow (2-5)	Slow (2-5)	



Core services

And now..

CONTINUOUS INTEGRATION

Questions to you

- Should we run clean install before every commit?
- How long should the build take?

Continuous Integration

- Don't run the tests yourself
- "Security-net"



Take it for a spin

- Show off hudson locally

Continuous Integration

- Let the machines do the testing
- Earliest possible feedback!
- The build is broke!
- What's broken?
- Who broke it?

Continuous Integration

- Continuum (from the Maven lads)
- Hudson (Koshuke, the devil)
- CruiseControl (good old)
- Bamboo (Atlassian)
- TeamCity (ditto..)

Things to consider

- One product, one build?
- (you could build site..)
- Lots of different things to test
- Many testing frequencies
 - Each commit
 - Hourly
 - Nightly

My advice

- For every combination (you want to do)
- Make a job: build goal plus frequency
 - 2 minute build
 - 10 minute build
 - database-integration tests
 - web-tests
 - every-2-hour-integration tests
 - nightly site/snapshot

Broken or unstable?

- Is it broken if only 1 test doesn't run?
 - What does broken mean?
 - Is that test relevant to all the other modules?
- Should we keep building the next modules

Broken or unstable?

- Make the CI see the difference
- Hudson and Bamboo shows **new** test-failures
- Yellow versus green builds

Conclusion

- Today:
 - Install Selenium IDE
- Next weeks:
 - Maintain a handful of tests
- When/if you learn to love them
 - Throw them into a CI

Questions?

THAT'S IT

If you've got any questions, please mail me at [tfnico <at> gmail.com](mailto:tfnico@gmail.com).

THANKS !