The future of enterprise testing

Aslak Knutsen
JBoss, by Red Hat
aslakknutsen

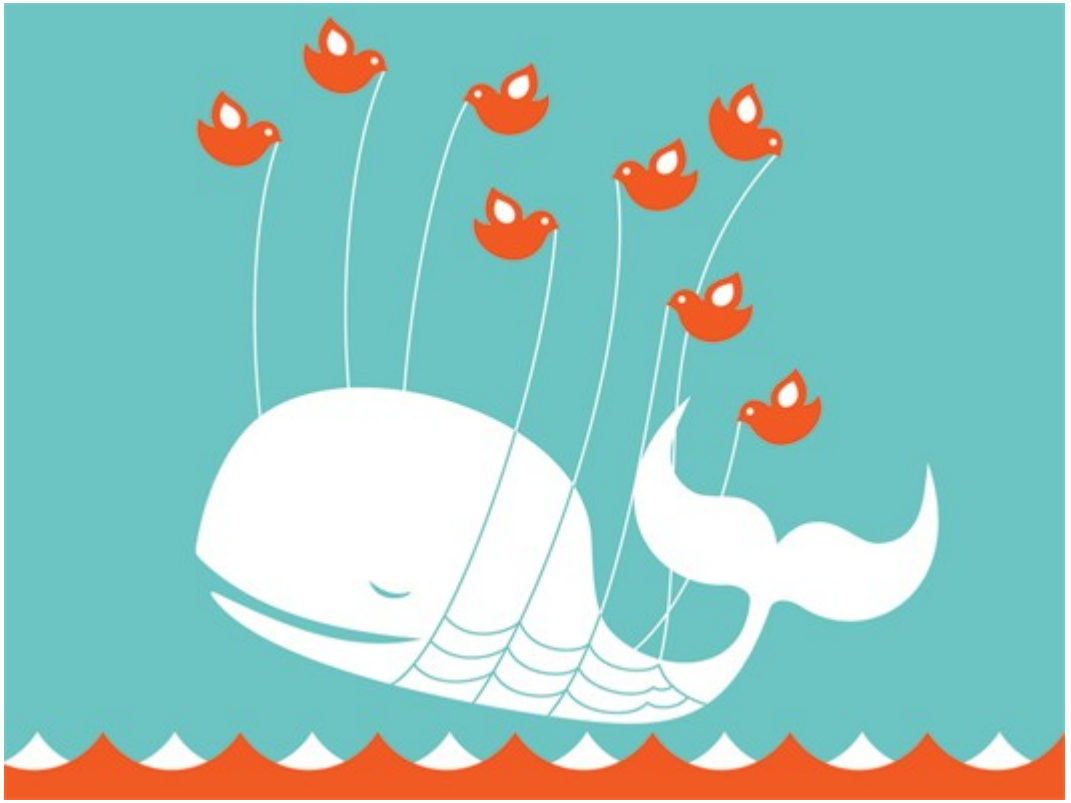
Dan Allen
JBoss, by Red Hat
mojavelinux

Agenda

#arquillian

- Why Java EE is harder than it should be
- How component models apply to testing
- Tools that help you develop & test with confidence
  - **ShrinkWrap** - *Skip the build*
  - **Arquillian** - *Test in-container*
- Demo, demo, demo
- What's coming next, with demos

No tests→→ #fail



Unit tests vs integration tests

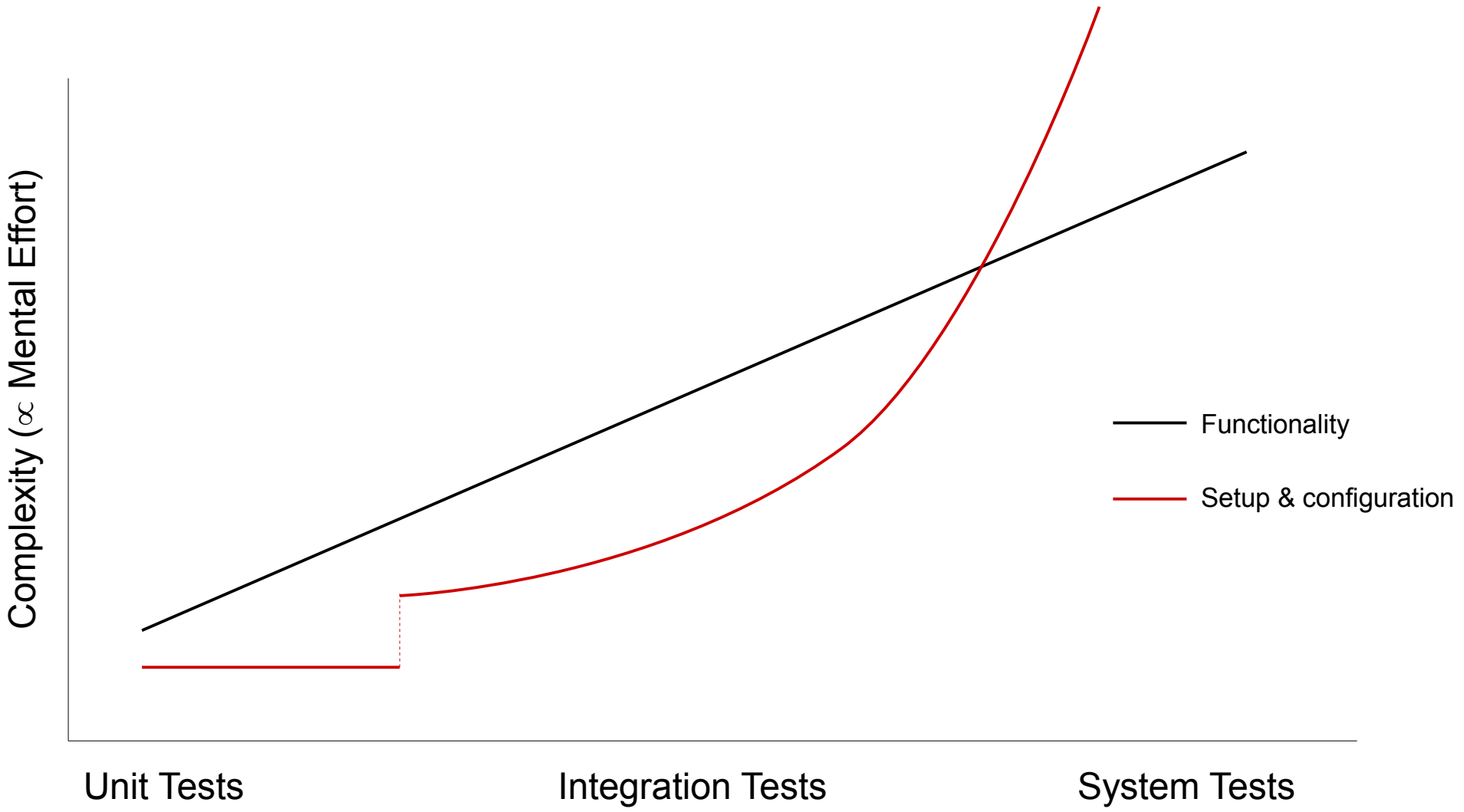
Unit

- Attributes
  - Fine-grained
  - Simple
  - Single API call
- Perception
  - Fast, fast, fast
  - Easily run in an IDE

Integration

- Attributes
  - Coarse-grained
  - Complex
  - Component interactions
- Perception
  - Slooooooooow
  - Run in an IDE? How?

The testing “bandgap”



What if integration testing could be...?

- as easy as writing a unit test
- run in the IDE (incremental builds, debugging, etc)
- ramped up in phases
- portable

Component models make life easier

- *Component*
  - Follows standard programming model
  - Encapsulates business logic
  - Packaged in deployable archive

- *Container*
  - Host process for deployed applications
  - Provides services and a runtime for components
  - Gives you *powerful mechanisms for free*

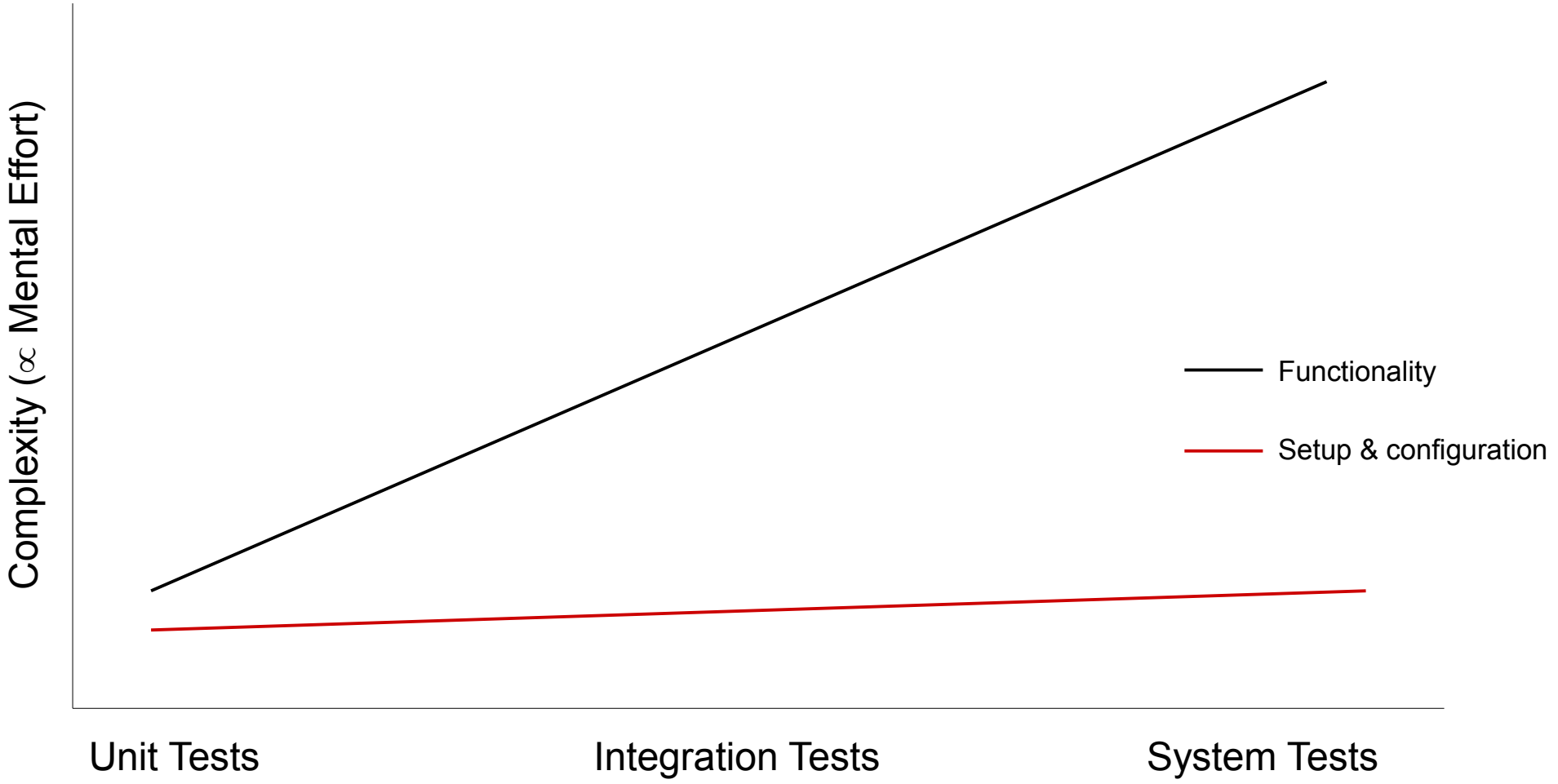
missing

What's been ^ from Java EE?

A *component model* for your tests

Reducing enterprise testing to child's play

Arquillian's test continuum



How do we get there?


Skip the build!
Test in-container!
arquillian

*Project lead: **A**ndrew **L**ee **R**ubinger*

ShrinkWrap http://jboss.org/shrinkwrap

n. *a fluent API for creating archives such as JARs, WARs and EARs in Java*

Benefits of ShrinkWrap

- Incremental IDE compilation
  - Save and re-run
  - Skip the build!
- Simple, fluent API
- Container deployment adapters
- Micro deployments
- Export and debugging

Fluent archive creation

```
JavaArchive archive =
   ShrinkWrap.create(JavaArchive.class, "slsb.jar")
      .addClasses(Greeter.class, GreeterBean.class);
System.out.println(archive.toString(true));
```

Yields output:

```
slsb.jar:
/com/
/com/acme/
/com/acme/app/
/com/acme/app/ejb3/
/com/acme/app/ejb3/Greeter.class
/com/acme/app/ejb3/GreeterBean.class
```

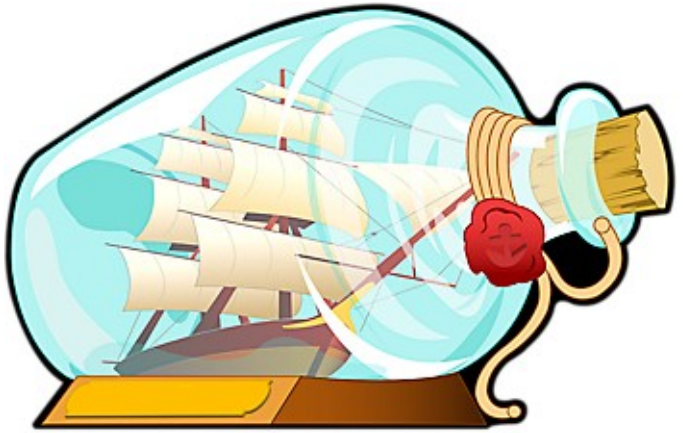
Project lead: ***Aslak*** *Knutsen*

http://jboss.org/arquillian

n. *a container-oriented testing framework that enables developers to create* ***portable integration tests*** *for enterprise applications; manages the lifecycle of a container and enriches, deploys and runs tests in the container or as a client*

An in-container approach to integration testing

1. Start or connect to a container
2. Package and deploy test case to container
3. Run test in-container
4. Capture and report results
5. Undeploy test archive



Bring your test to the runtime...

...instead of managing the runtime from your test.

Arquillian project mission

Make integration testing a breeze!

Prove it.

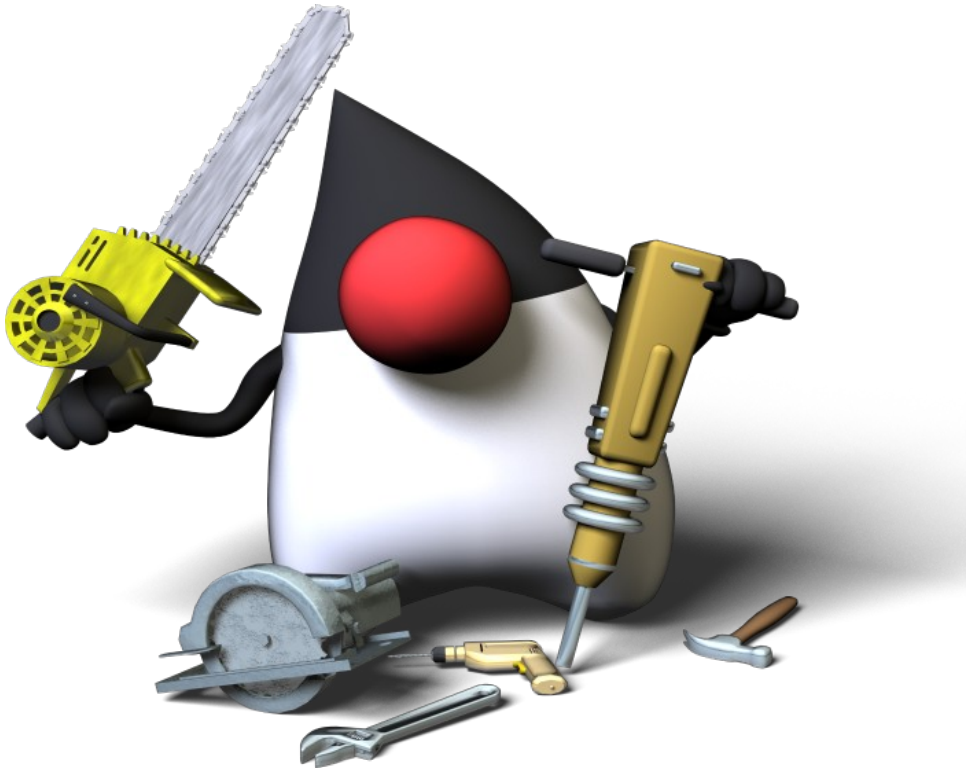
```
@RunWith(Arquillian.class)
public class GreeterTestCase {

    @Deployment
    public static Archive<?> createDeployment() {
        return ShrinkWrap.create(JavaArchive.class)
            .addClasses(Greeter.class, GreeterBean.class);
    }

    @EJB Greeter greeter;

    @Test
    public void shouldBeAbleToInvokeEJB() {
        Assert.assertEquals("Hello, Earthlings",
            greeter.greet("Earthlings"));
    }
}
```


What's on the way?

- **Full-on, container invasion**
  - Multi-deployment, multi-container, multi-node ***per test***
  - Sophisticated configuration
- **ShrinkWrap descriptors**
  - Fluent API for creating/modifying XML descriptors
- **Container metadata**
  - Configuration (e.g., ports, urls)
  - Deployment information (e.g., context path)
- **Dependency management**
  - Isolated container dependencies
  - Adding libraries to test archive

Power tools: test framework integration

- Test frameworks are services too!
- Extends test component model
- Examples:
  - JSFUnit*
  - Cobertura*
  - Spock*
  - Selenium*
  - HTTPUnit
  - DBUnit



* available

Experience Ike

Benefits of Arquillian

- Write less (boilerplate) test code
- As much or as little “integration” as you need
- Looks like a unit test, but you're in a true environment!
- Run same test in multiple containers
- It's a ***learning*** environment

Arquillian...

- is a container-oriented testing framework
- provides a component model for tests
- handles test infrastructure & plumbing
- ships with a set of container implementations
- saves your fail ;)

Get involved!

- Download us!
  - **ShrinkWrap** - http://jboss.org/shrinkwap
  - **Arquillian** - http://jboss.org/arquillian
- Participate with us!
  - Community Space
- Fork us! github SOCIAL CODING
- Meet us!
  - #jbosstesting channel on irc.freenode.net
- Write for us!

  - Share your stories – Blog! Tweet! **#arquillian**
  - Document how it works