



#arquillian

Java. Cloud. Leadership.

High Octane Development

JBoss AS 7 with Arquillian

Aslak Knutsen
@aslakknutsen





**AHH, IT'S TEST
HELL ON EARTH!**





**CAN WE TAKE
BACK CONTROL?**



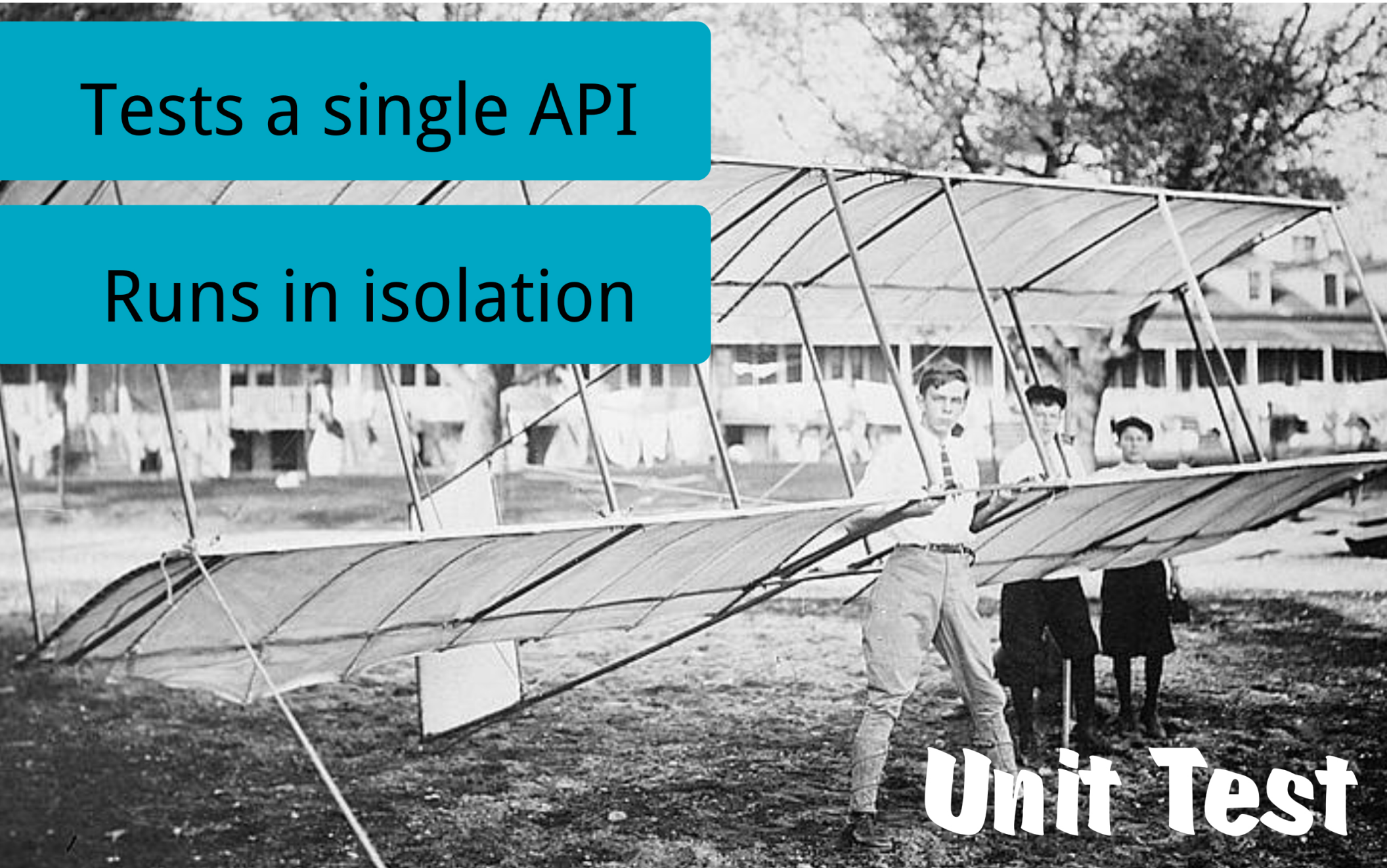


How do we prevent
software failures?

Tests a single API

Runs in isolation

Unit Test

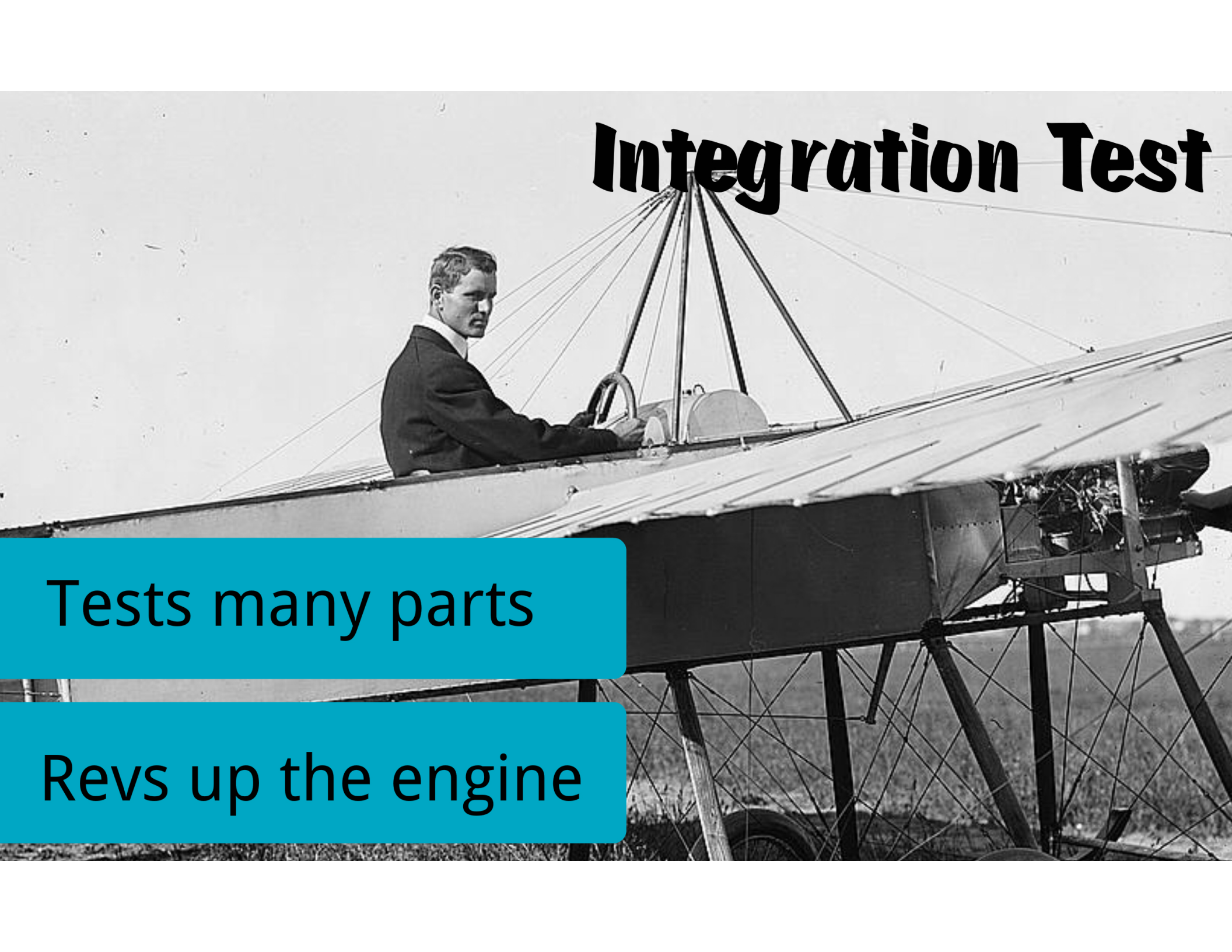


Unit tests are so
fine-grained.

Fast, simple
and easy.
Just as
testing
should be.



Integration Test

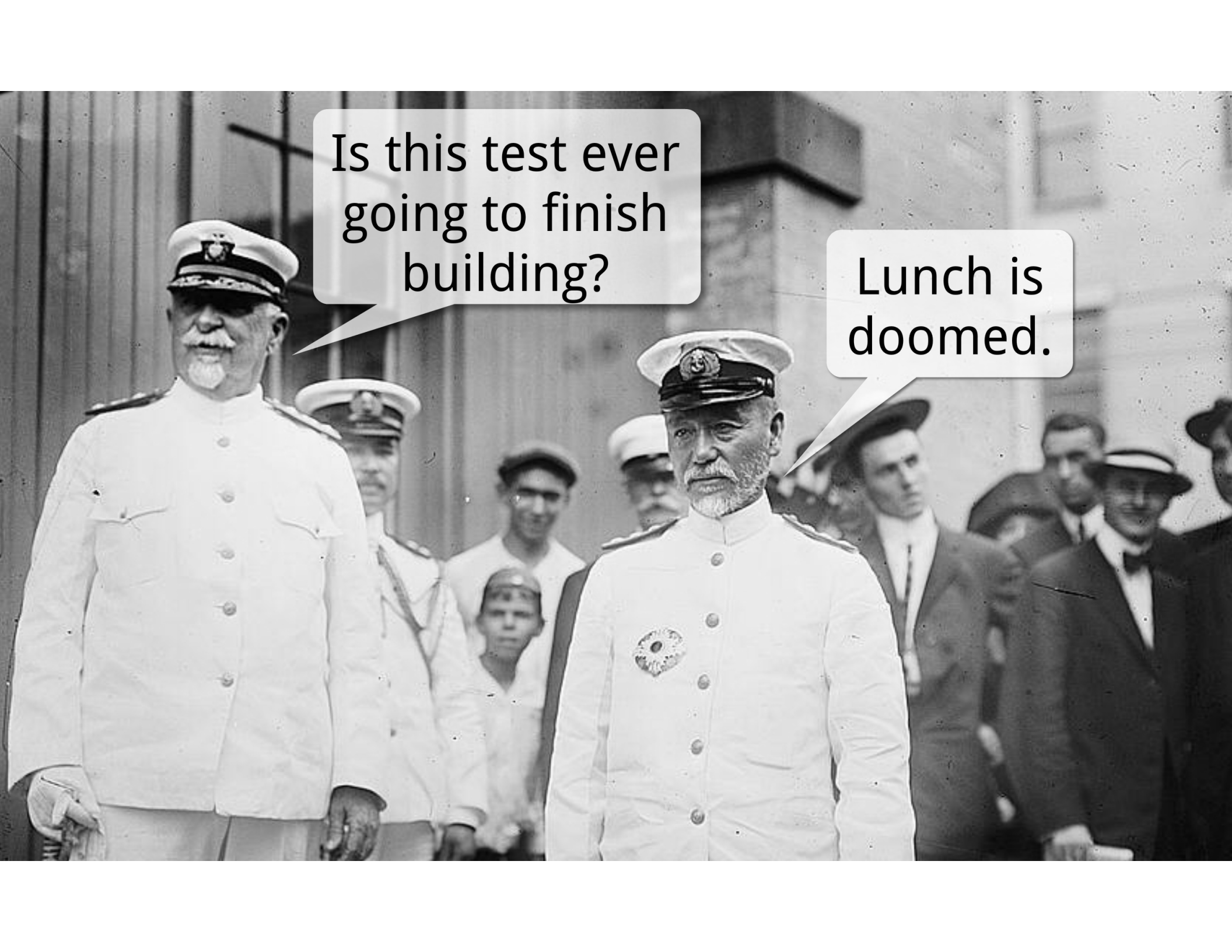
A black and white photograph of Orville Wright in the cockpit of the Wright Flyer. He is wearing a dark suit and is looking forward, holding the controls. The aircraft is a biplane with a high-wing configuration, supported by a complex system of struts and wires. The background is a clear sky.

Tests many parts

Revs up the engine

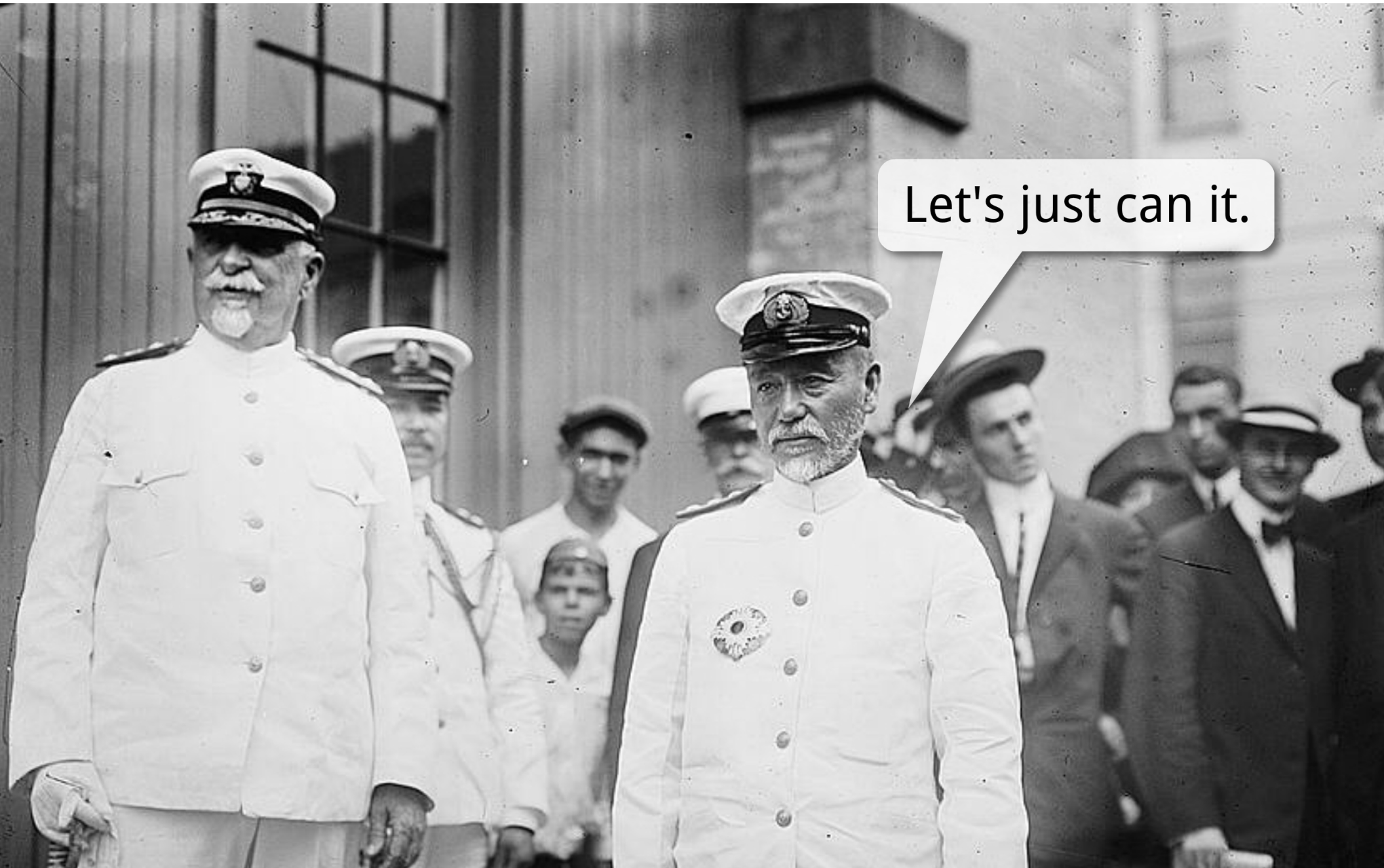
Nice if they
could run in an
IDE.

Ha! In your
dreams.

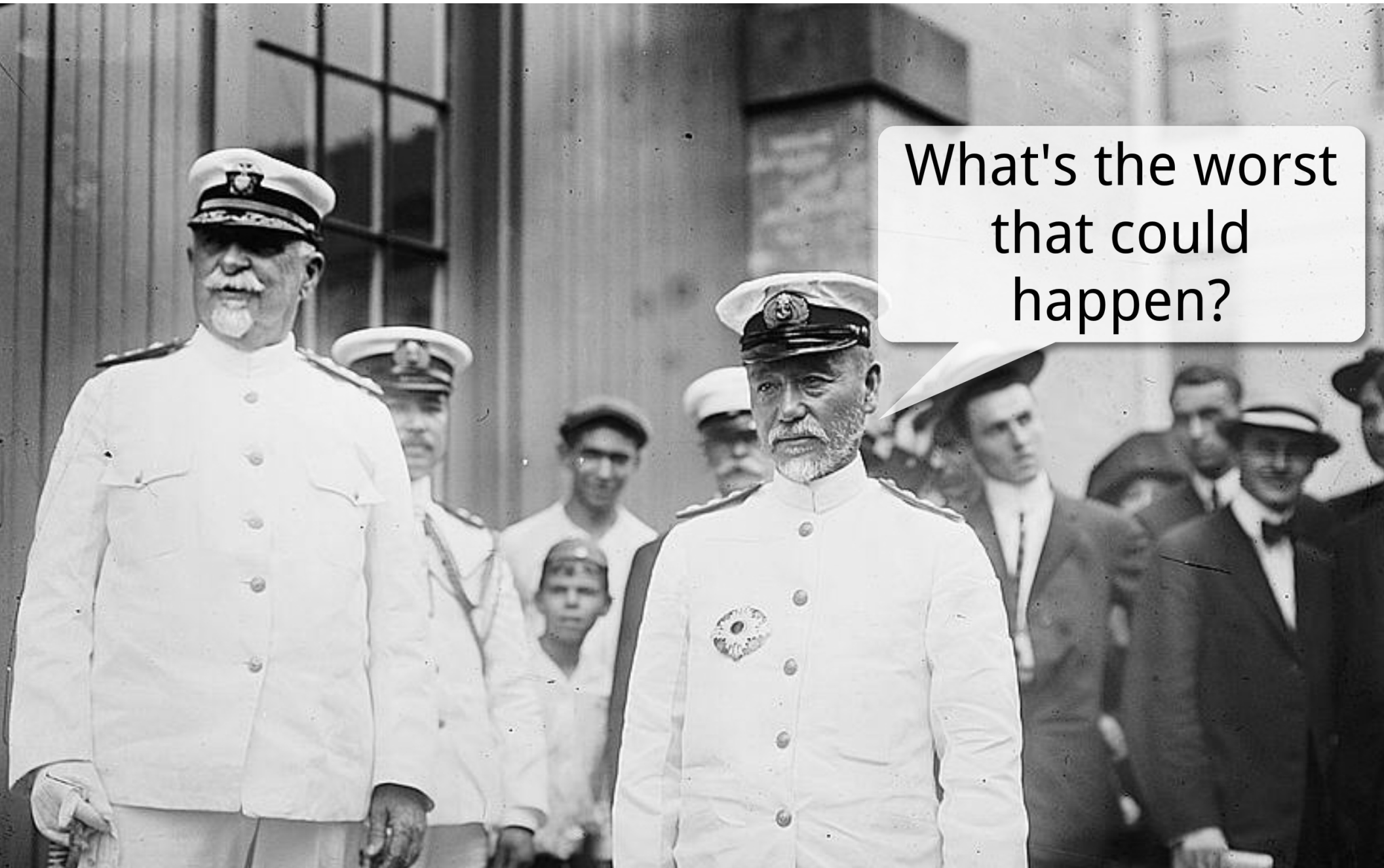


Is this test ever
going to finish
building?

Lunch is
doomed.



Let's just can it.

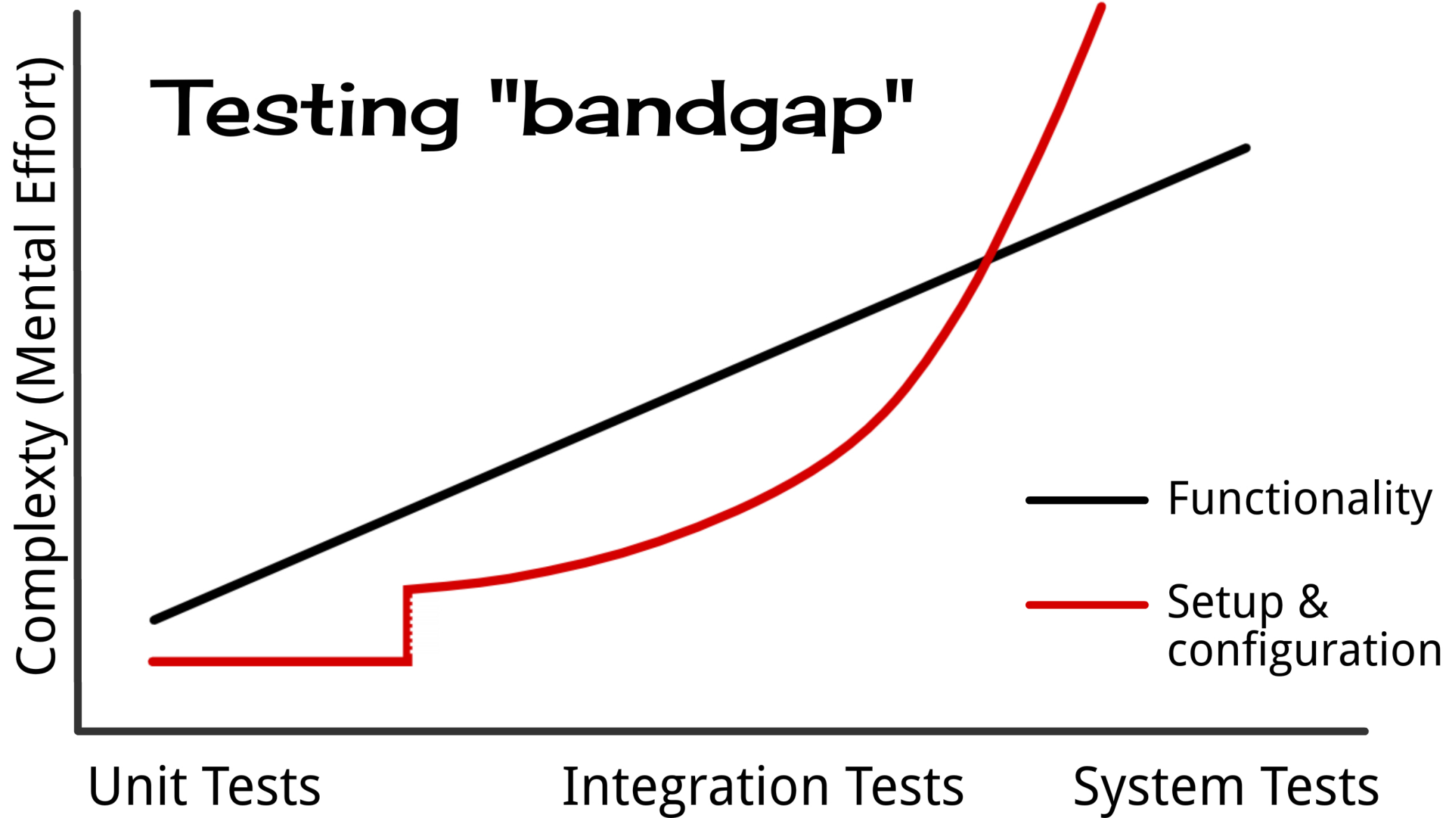


What's the worst
that could
happen?

#fail



Testing "bandgap"



*graph is theoretical



We've got to
close this gap.



What would help us
write tests more
efficiently?

It's called a
component model
for testing.





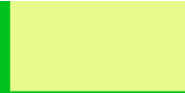
Adheres to a
standard
programming
model



Gives you
powerful
mechanisms
for free

Keeps you **focused** on the test at hand





How can integration
testing speed up
development?



Code

V00000

Build



Test



Skip the build.





Code



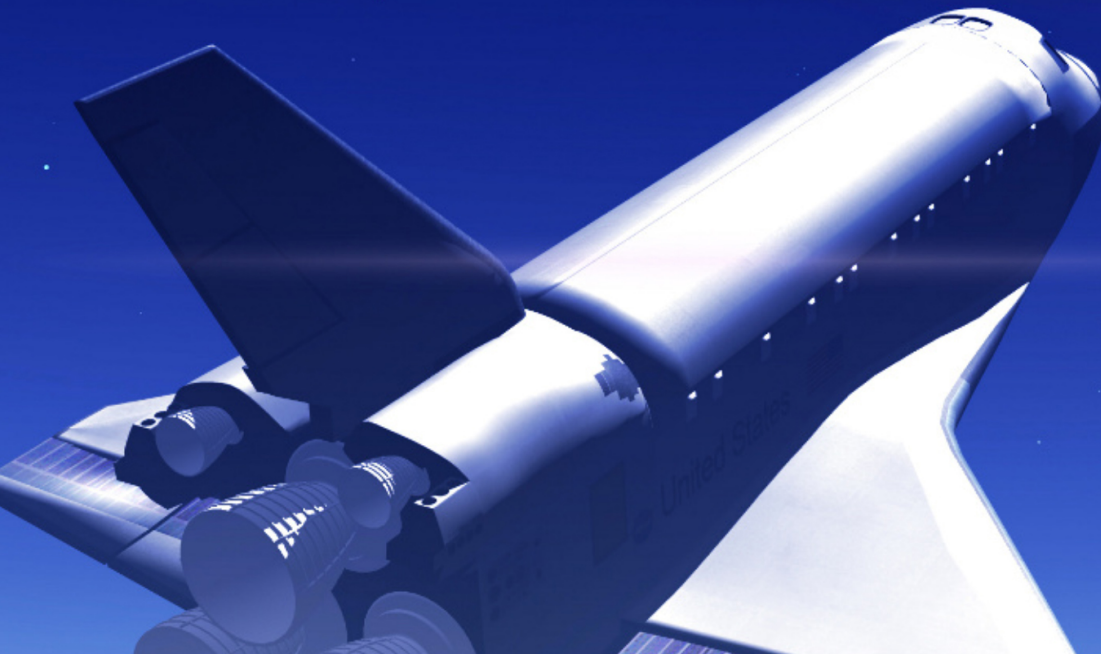
Test



Bring your test to the runtime...



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



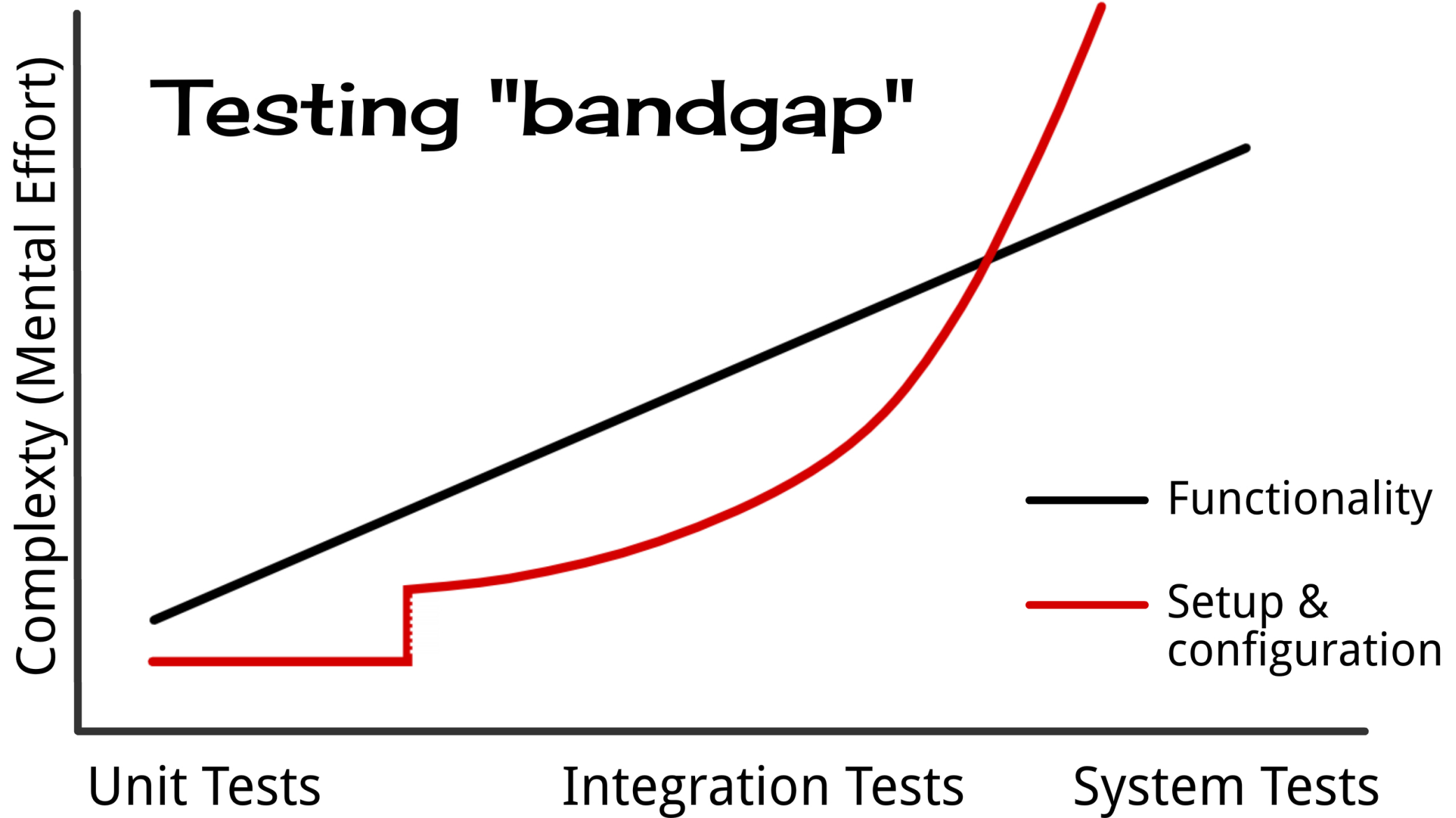
<http://arquillian.org>



arquillian

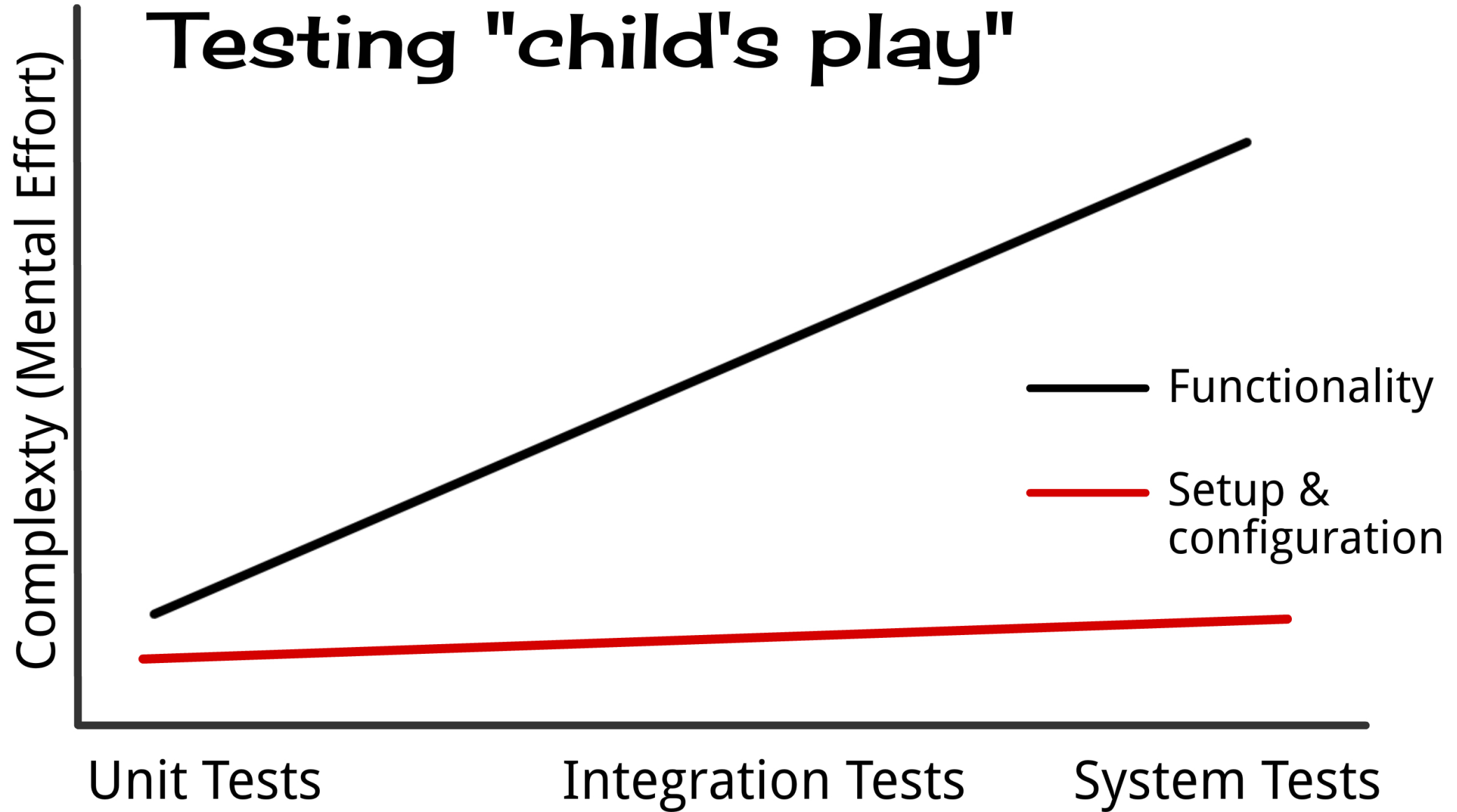
A container-oriented testing
framework

Testing "bandgap"



*graph is theoretical

Testing "child's play"



*graph is theoretical

```
public class GreeterTestCase {
```

```
}
```

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

```
}
```



```
@RunWith(Arquillian.class)
public class GreeterTestCase {
    @Deployment
    public static JavaArchive createDeployment() {
        return ShrinkWrap.create(JavaArchive.class)
            .addClass(Greeter.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
    }
}
```

```
@RunWith(Arquillian.class)
public class GreeterTestCase {
    @Deployment
    public static JavaArchive createDeployment() {
        return ShrinkWrap.create(JavaArchive.class)
            .addClass(Greeter.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
    }

    @Inject Greeter greeter;

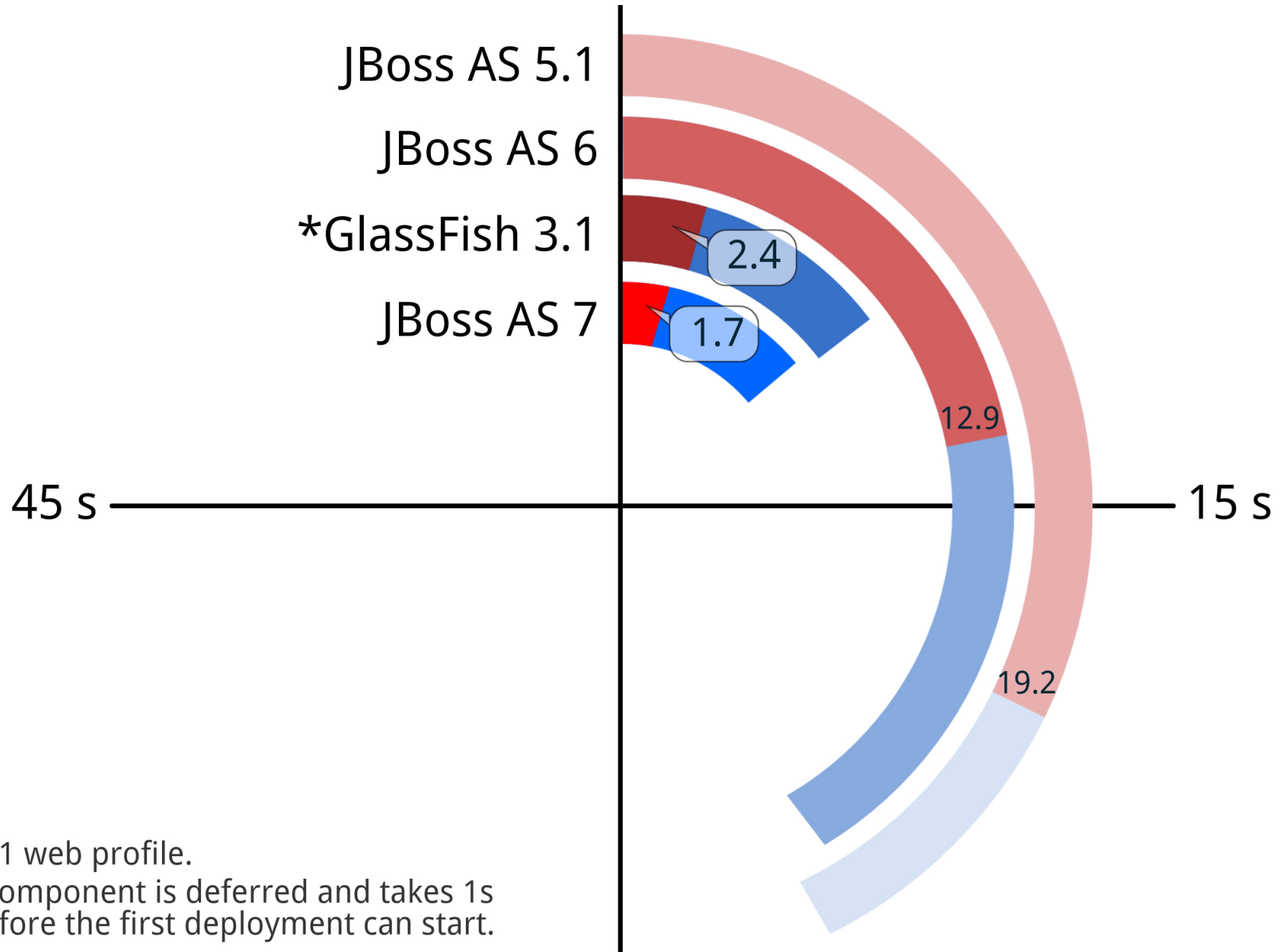
}
```

```
@RunWith(Arquillian.class)
public class GreeterTestCase {
    @Deployment
    public static JavaArchive createDeployment() {
        return ShrinkWrap.create(JavaArchive.class)
            .addClass(Greeter.class)
            .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
    }

    @Inject Greeter greeter;

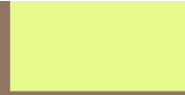
    @Test
    public void shouldBeAbleToInvokeBean() throws Exception {
        Assert.assertEquals("Hello, Earthlings", greeter.greet("Earthlings"));
    }
}
```





*GlassFish 3.1 web profile.

The web component is deferred and takes 1s to load before the first deployment can start.



Can we put this
productivity into
practice?

Prove it.



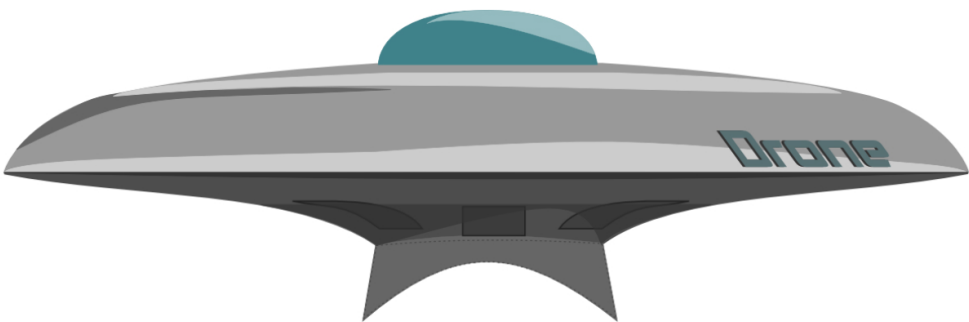


THE **IKE** EXPERIENCE

github.com/arquillian/arquillian-showcase







JSFUnit

jclouds

multi - cloud library

Infinispan



Byteman

- ✓ Less (boilerplate) test code
- ✓ As much or as little integration as you need
- ✓ Looks like a unit test, but executes in a true environment
- ✓ Same test can be run in multiple environments, including...
- ✓ It's a **learning** environment





- Rob Williams
jroller.com

**ARQUILLIAN:
END OF TEST HELL
ON EARTH**

**GO BURN
PRODUCTIVITY
RUBBER!**





THANK YOU!
(AND DON'T MOCK ME)



jboss.org/arquillian

Historical photographs courtesy of the Library of Congress



[flickr.com/photos/library_of_congress/](https://www.flickr.com/photos/library_of_congress/)