

Experts in agile software engineering

## ***One suite that suits them all***

**Leif Frenzel**

andrena objects ag

[leif@andrena.de](mailto:leif@andrena.de)

**Marc Philipp**

andrena objects ag

[marc@andrena.de](mailto:marc@andrena.de)

Project Usus

Once  
upon a  
time





Once  
upon a  
time



... in an IT office...

This page is intentionally left blank.

A couple of hours later...



This page is intentionally left blank.

Some days  
later....





Some days  
later....

(Unusually early in the morning...)



Some days  
later....

(Unusually early in the morning... I mean, for software people... ;-)

This page is intentionally left blank.

Some days later...





Some days later...

... it's a rainy day



Some days later...

... it's a rainy day,  
wind's howling...



This page is intentionally left blank.

# Classpath Suite

<http://johanneslink.net/projects/cpsuite.jsp>

Provides a proxy test suite that finds all test cases on the classpath.

Additional include/exclude filtering by name pattern.

```
import org.junit.extensions.cpsuite.ClasspathSuite;
import org.junit.extensions.cpsuite.ClasspathSuite.ClassnameFilters;
import org.junit.runner.RunWith;

@RunWith( ClasspathSuite.class )
@ClassnameFilters( { ".*Test", "!.*IntegrationTest", "!.*PdeTest" } )
public class AllTests {
    // nothing here
}
```



# Classpath Suite

Needs to load all classes on the class path to analyze them:

- ▶ gets slow on large workspaces
- ▶ speed hit scales with number of classes (not number of tests)

# Classpath Suite

Needs to load add classes on the class path to analyze them:

- ▶ gets slow on large workspaces
- ▶ speed hit scales with number of classes (not number of tests)

## **Statistics from a customer project**

No. of files looked at by CPSuite: 9313

No. of classes loaded by CPSuite: 4716

Test classes: 533

Test methods: 2552

Time for finding test cases: 43.6s

Time for entire JUnit run: 142.4 s

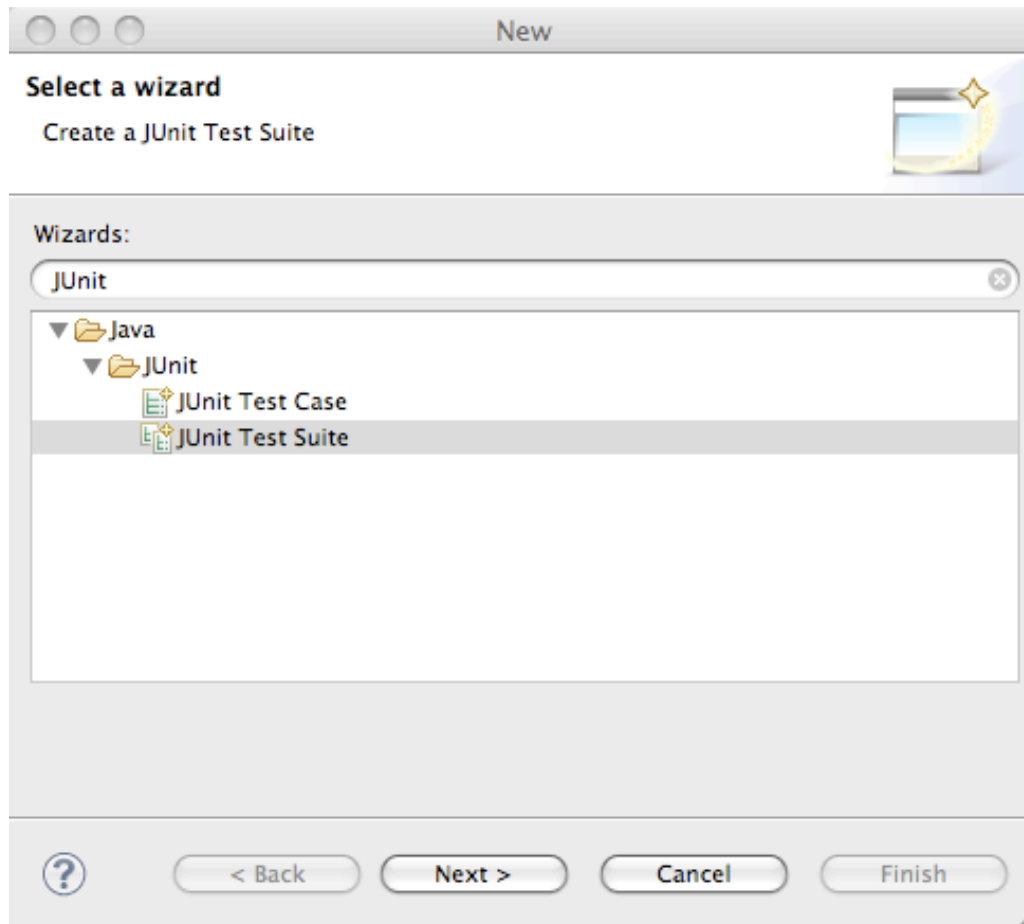
# Classpath Suite

Needs to load add classes on the class path to analyze them:

- ▶ gets slow on large workspaces
- ▶ speed hit scales with number of classes (not number of tests)
- ▶ executes static code (with all side-effects that may have)

## Related work

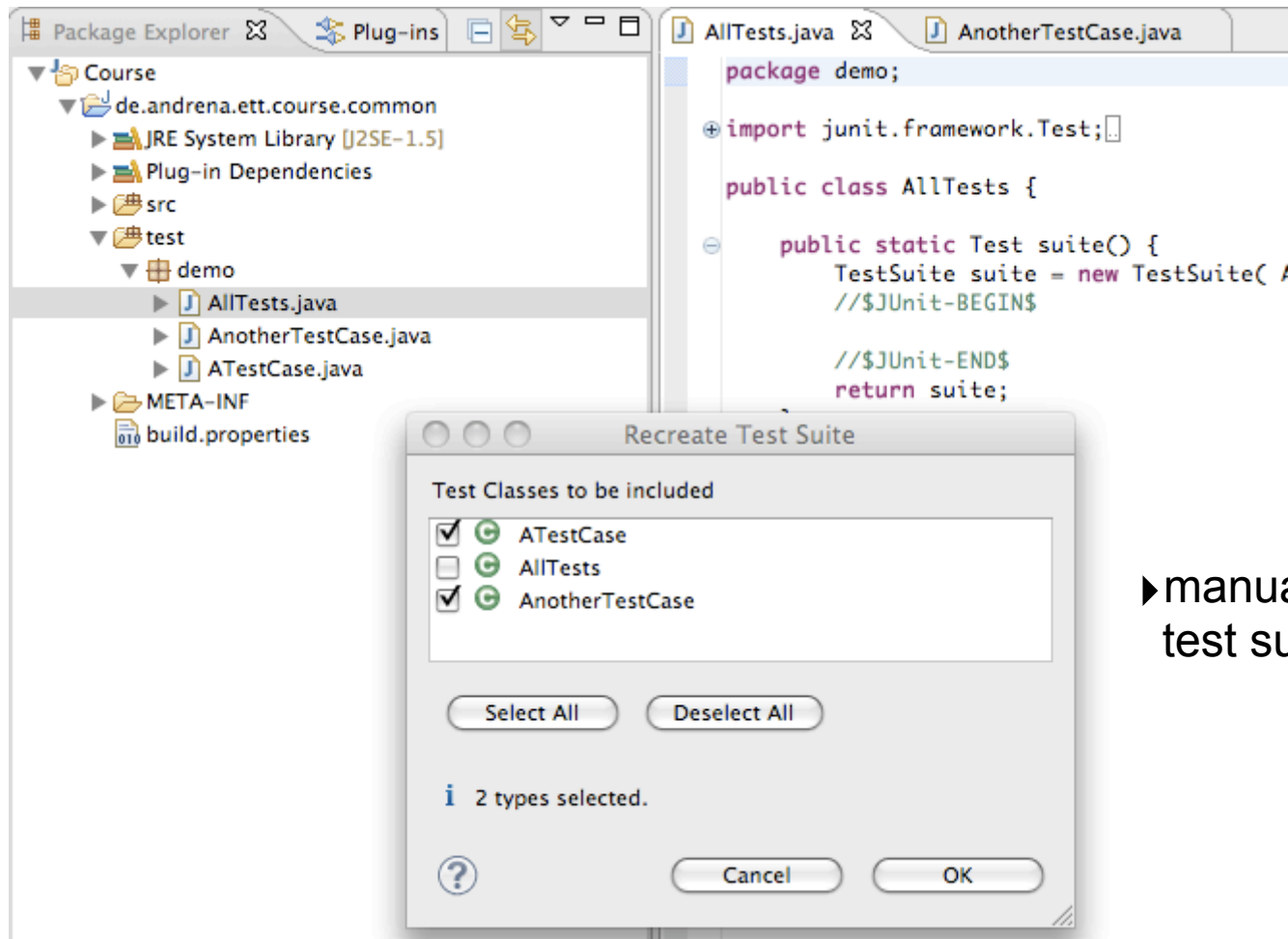
# JDT's New JUnit Test Suite wizard





## Related work

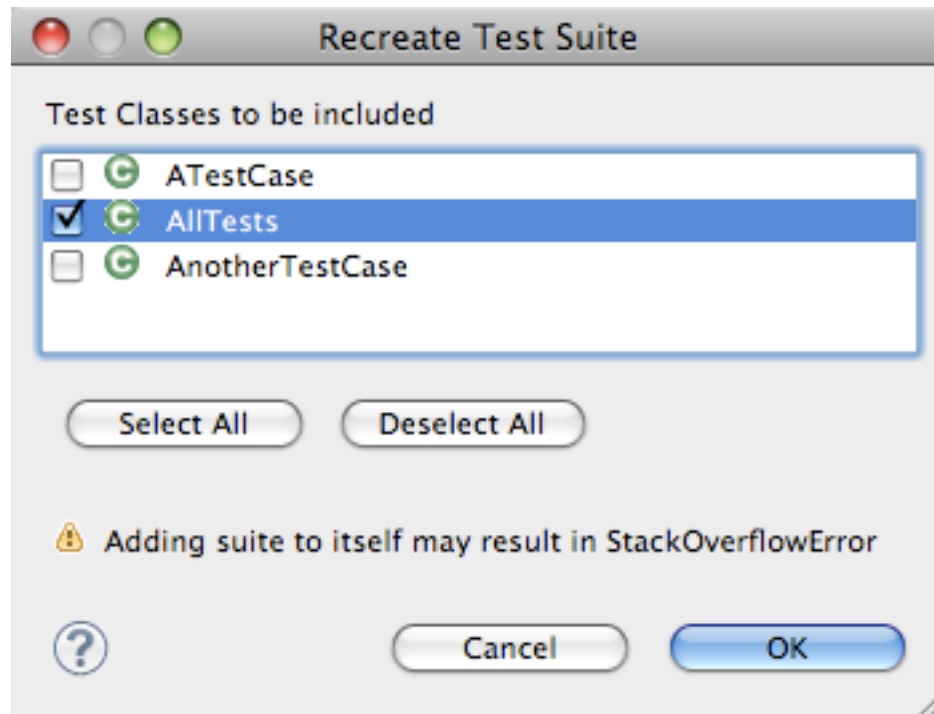
# JDT's New JUnit Test Suite wizard



► manually re-create test suites

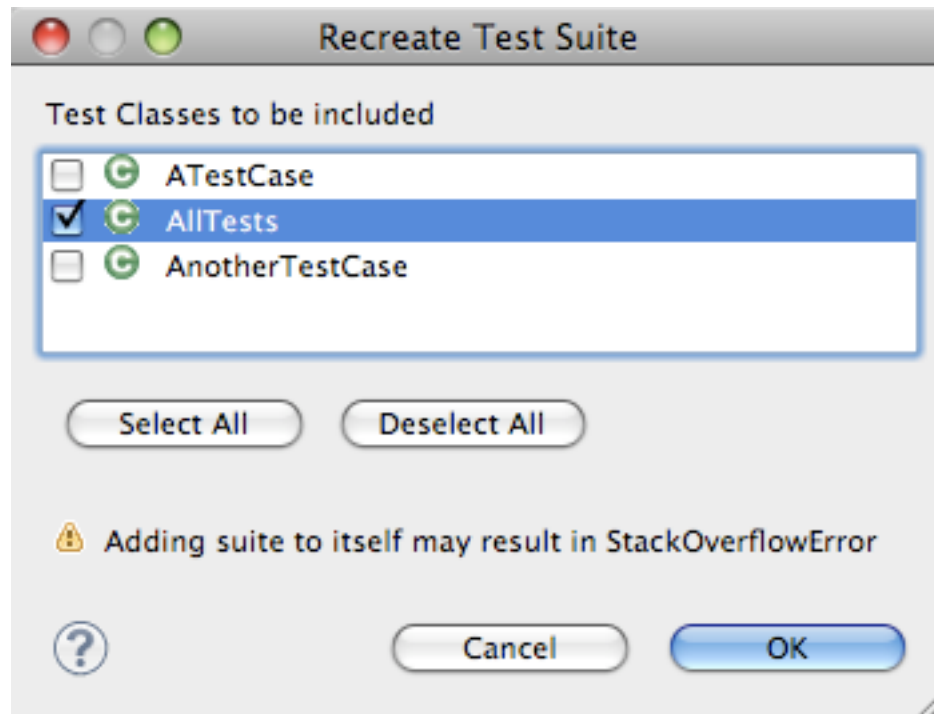
## Related work

### JDT's New JUnit Test Suite wizard



## Related work

### JDT's New JUnit Test Suite wizard



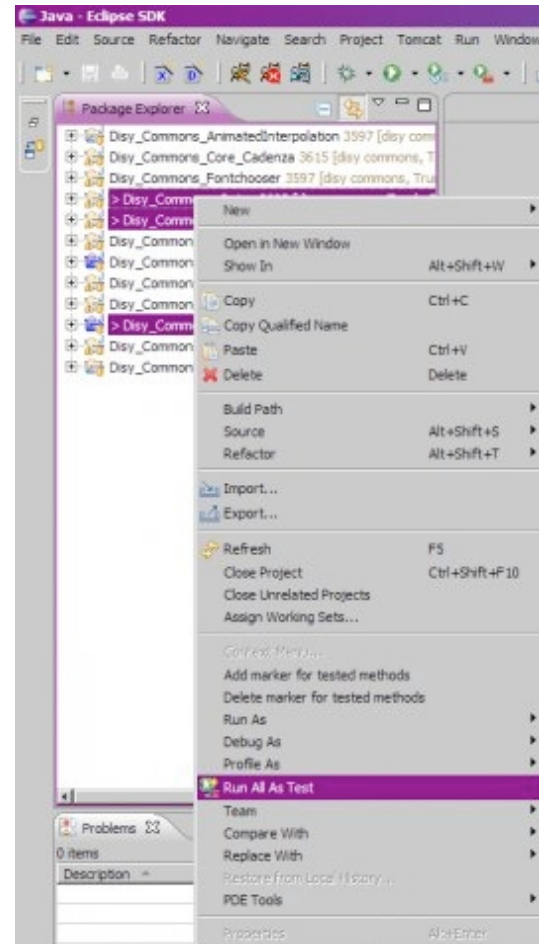
► works only for JUnit 3.x

## Related work

### E-rat (Eclipse - Run All Tests plug-in)

<http://e-rat.sourceforge.net/>

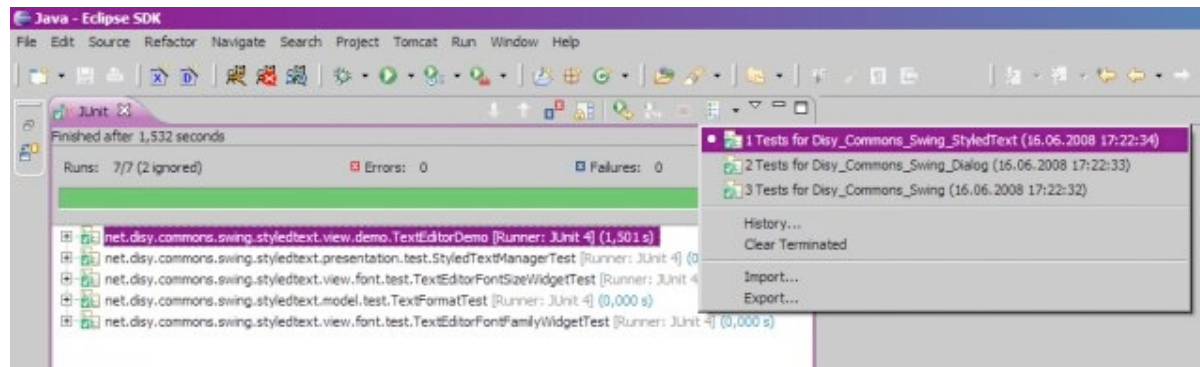
- ▶ allows to select multiple workspace projects and runs all their test cases
- ▶ recently updated to run on Eclipse 3.6



## Related work

### E-rat (Eclipse - Run All Tests plug-in)

<http://e-rat.sourceforge.net/>



- ▶ no single launch - launches each project separately
- ▶ no accumulated result in JUnit view

## Related work

### Build systems (Ant, Maven, ...)

```
<target name="test-html">
  <junit fork="yes" printsummary="no" haltonfailure="no">
    <batchtest fork="yes" todir="${test.reports}" >
      <fileset dir="${classes}">
        <include name="**/*Test.class" />
      </fileset>
    </batchtest>
    <formatter type="xml" />
    <classpath refid="test.classpath" />
  </junit>
</target>
```

- ▶ typically relies on naming conventions
- ▶ weak IDE integration
- ▶ tends to be used for build server and neglected in the IDE



## Review

- ▶ explicit AllTests suite

  - Pro:** run with single launch

  - Con:** maintain manually, error-prone

- ▶ Build system/script (\*\*/\*Test)

  - Pro:** run with single launch

  - Con:** often needs naming convention, weak IDE integration

- ▶ Classpath Suite

  - Pro:** run with single launch

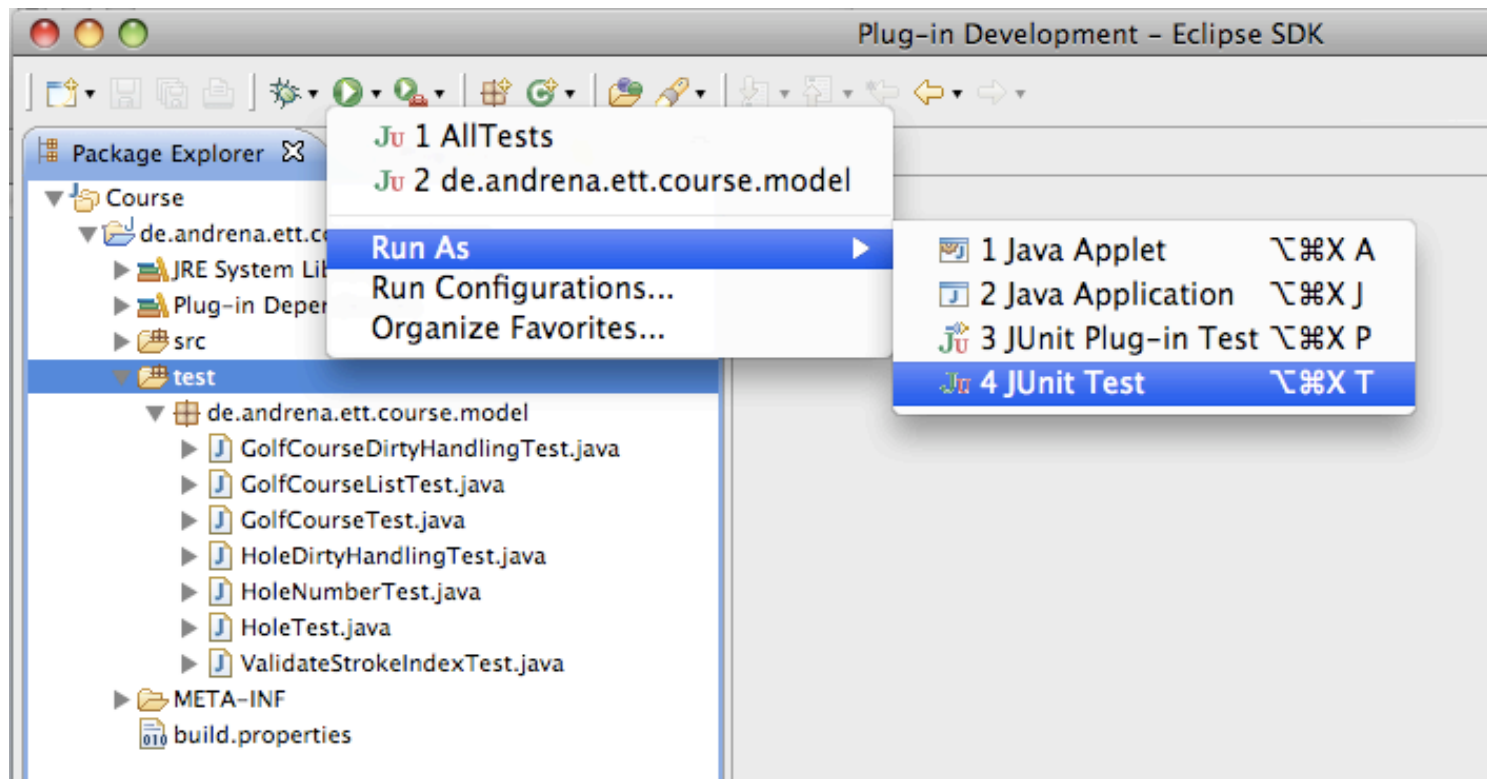
  - Con:** slow on large workspaces, executes static code

- ▶ JDT's Run As . . . on workspace project

  - Pro:** run with single launch

  - Con:** works only on single project or source folder

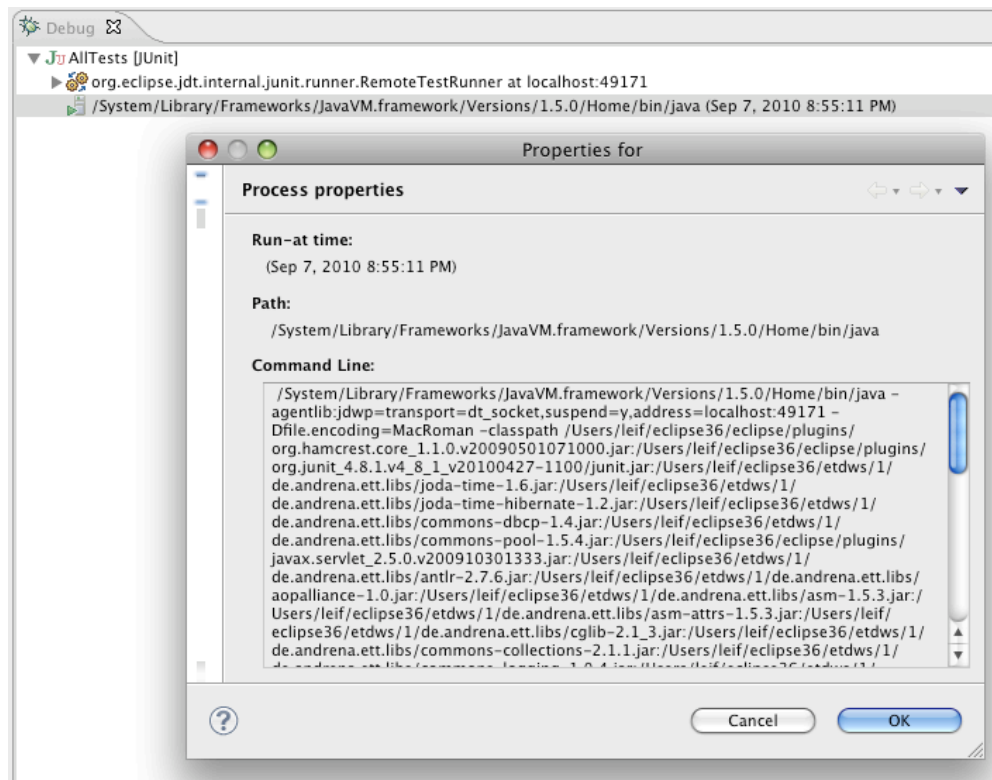
## JDT's Run As . . . on workspace project



Why can I do this only on a single project (no multi-selection)?

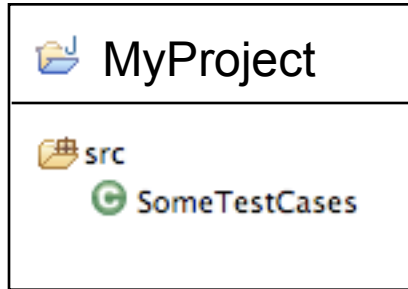
# JDT's Run As . . . on workspace project

Launch must run in the context of a single project in order to build the classpath for the VM runner.



► Cannot run classes from unrelated projects: potential classpath conflicts.

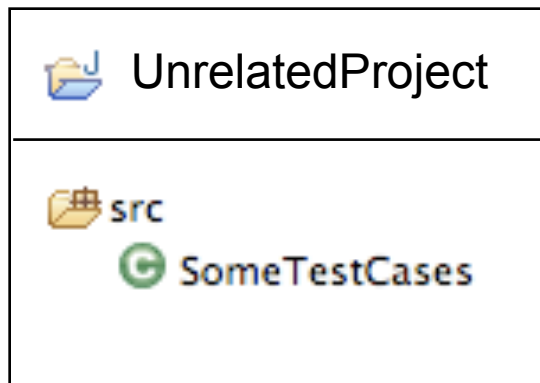
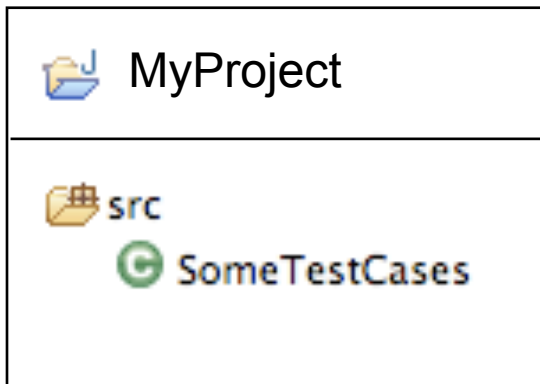
## JDT's Run As . . . refined



◀  Run As... JUnit Test

Runs all tests in MyProject.

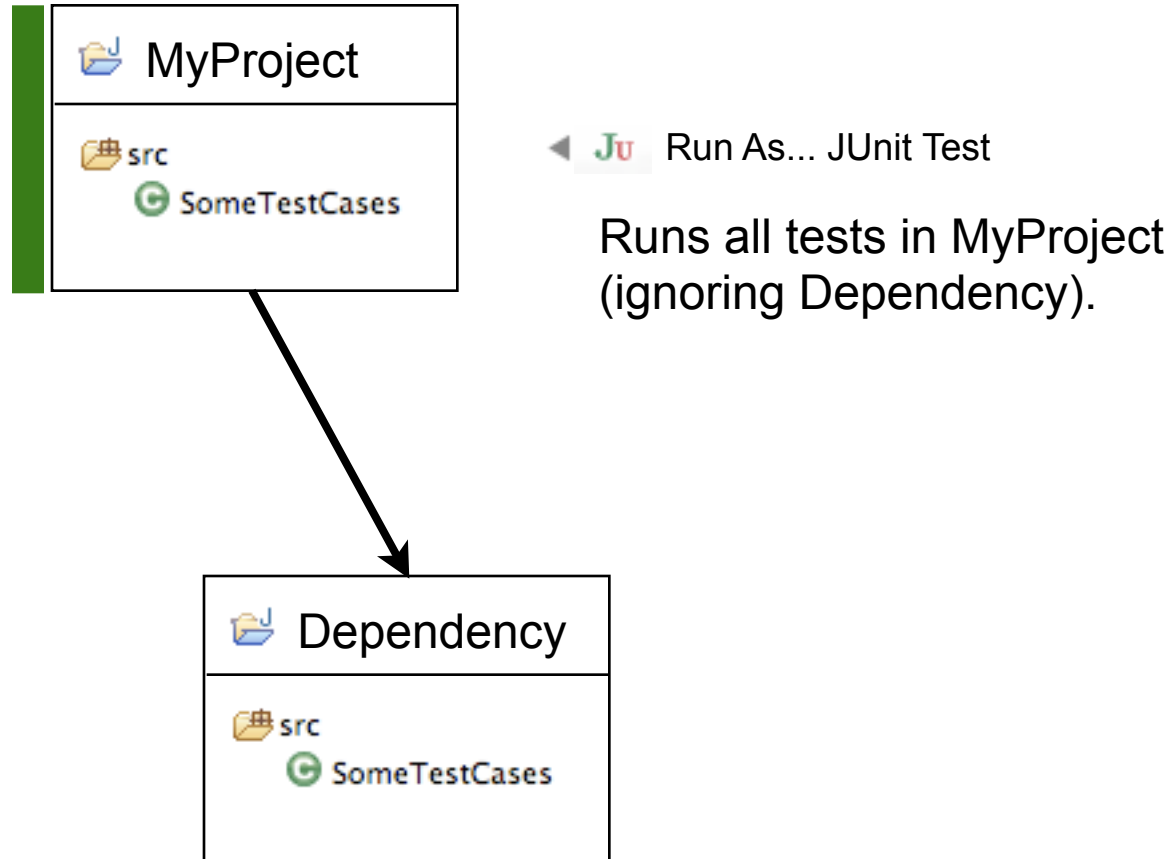
## JDT's Run As . . . refined



◀ JUnit Run As... JUnit Test

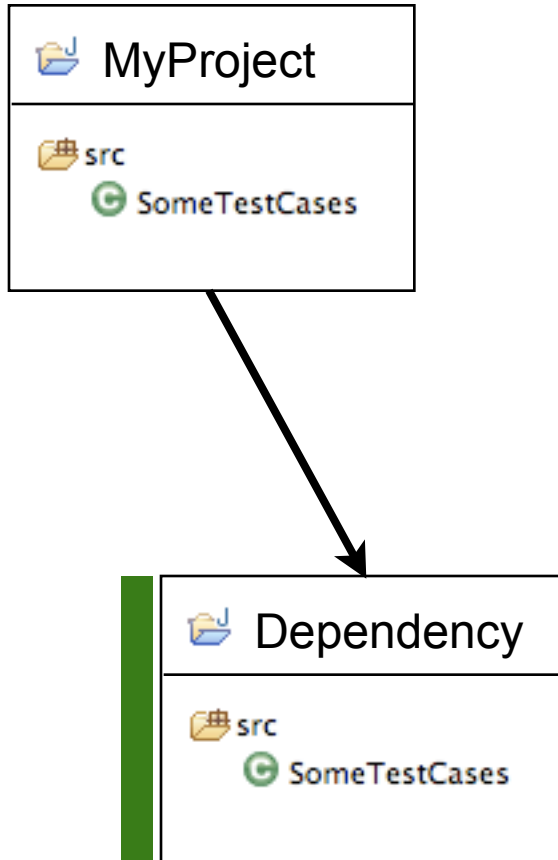
Potentially incompatible  
classpaths.

## JDT's Run As . . . refined





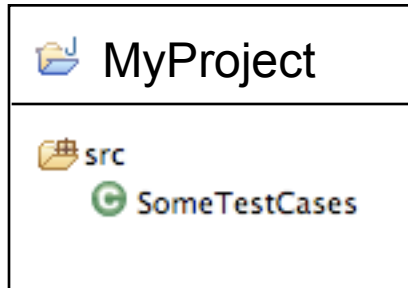
## JDT's Run As . . . refined



◀  Run As... JUnit Test

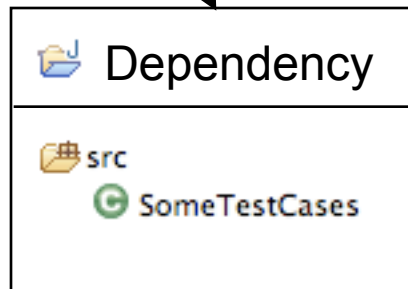
Runs all tests in Dependency  
(ignoring MyProject).

## JDT's Run As . . . refined

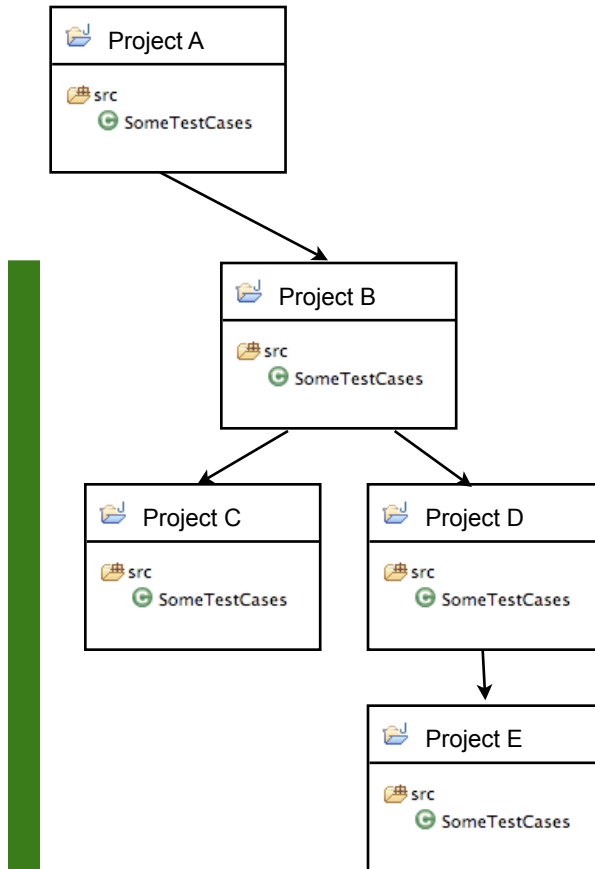


◀ JUnit Run As... **Refined** JUnit Test

Runs all tests in MyProject  
**and** Dependency.



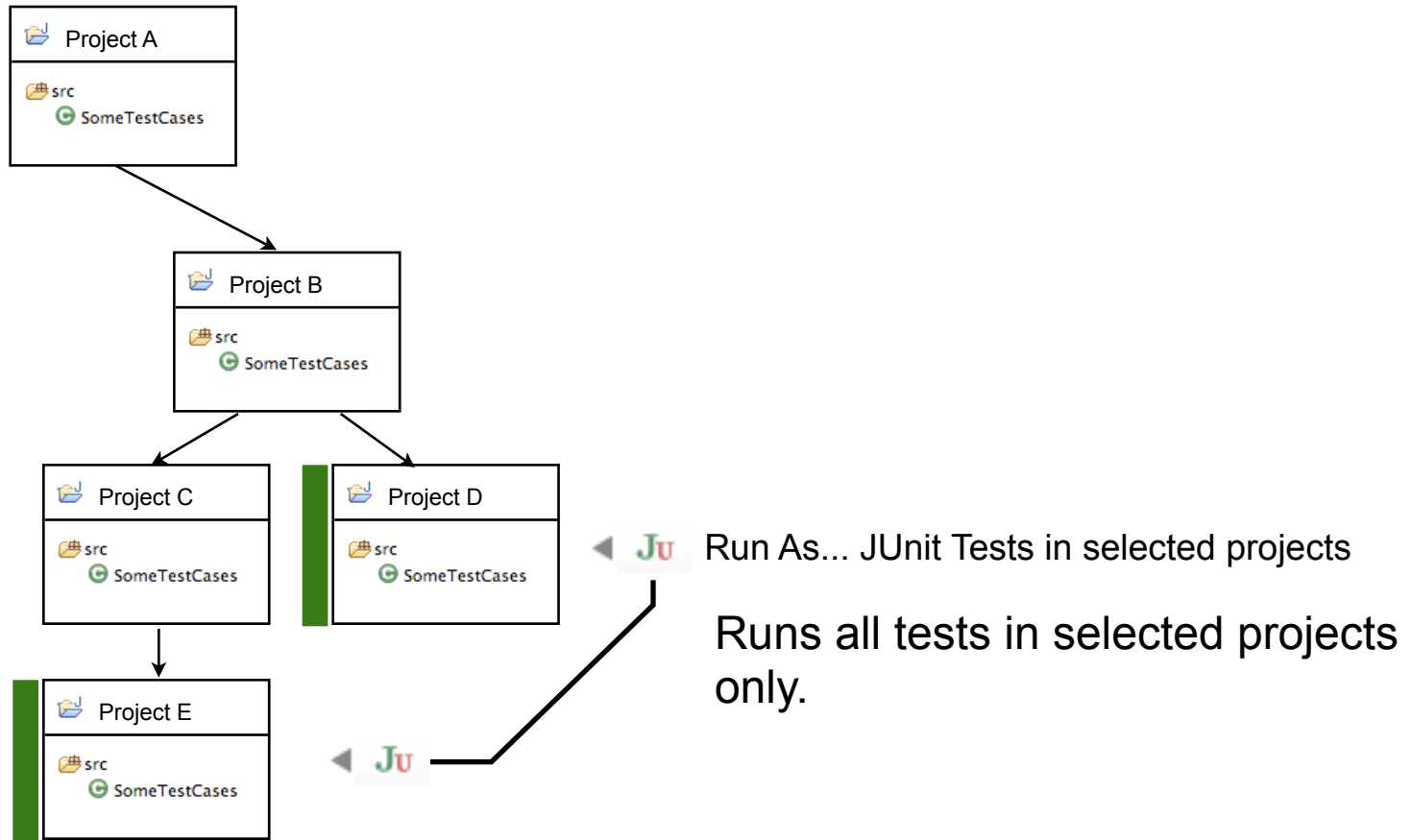
# JDT's Run As... refined



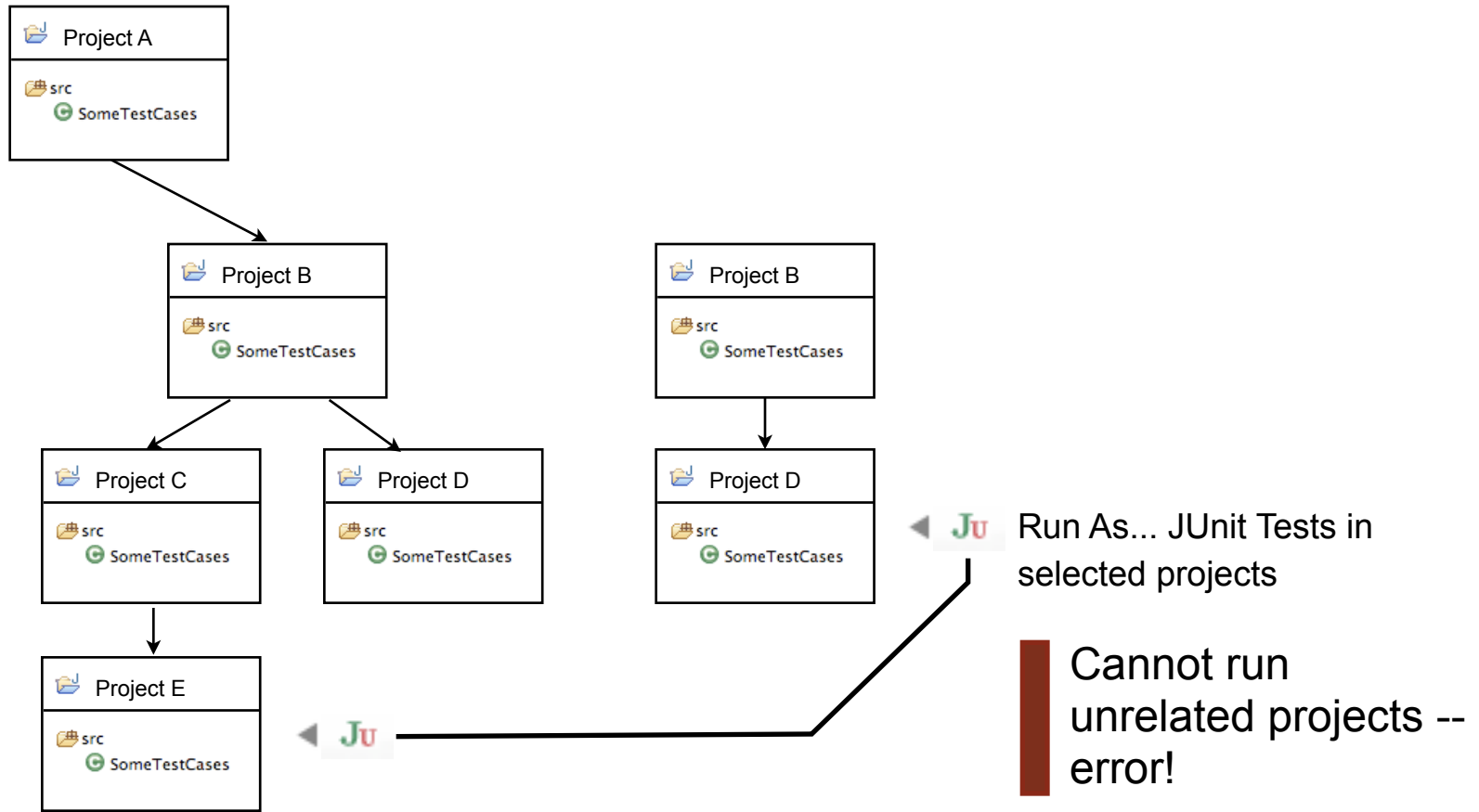
◀  Run As... **Refined** JUnit Test

Runs all tests in selected project and ***all dependencies***.

# JDT's Run As . . . refined



# JDT's Run As... refined



**Demo**



***Thanks for your attention!***

**andrena**  
OBJECTS

<http://andrena.de>

Project Usus

<http://projectusus.org>