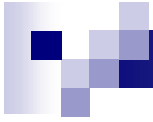


# Whitebox Testing

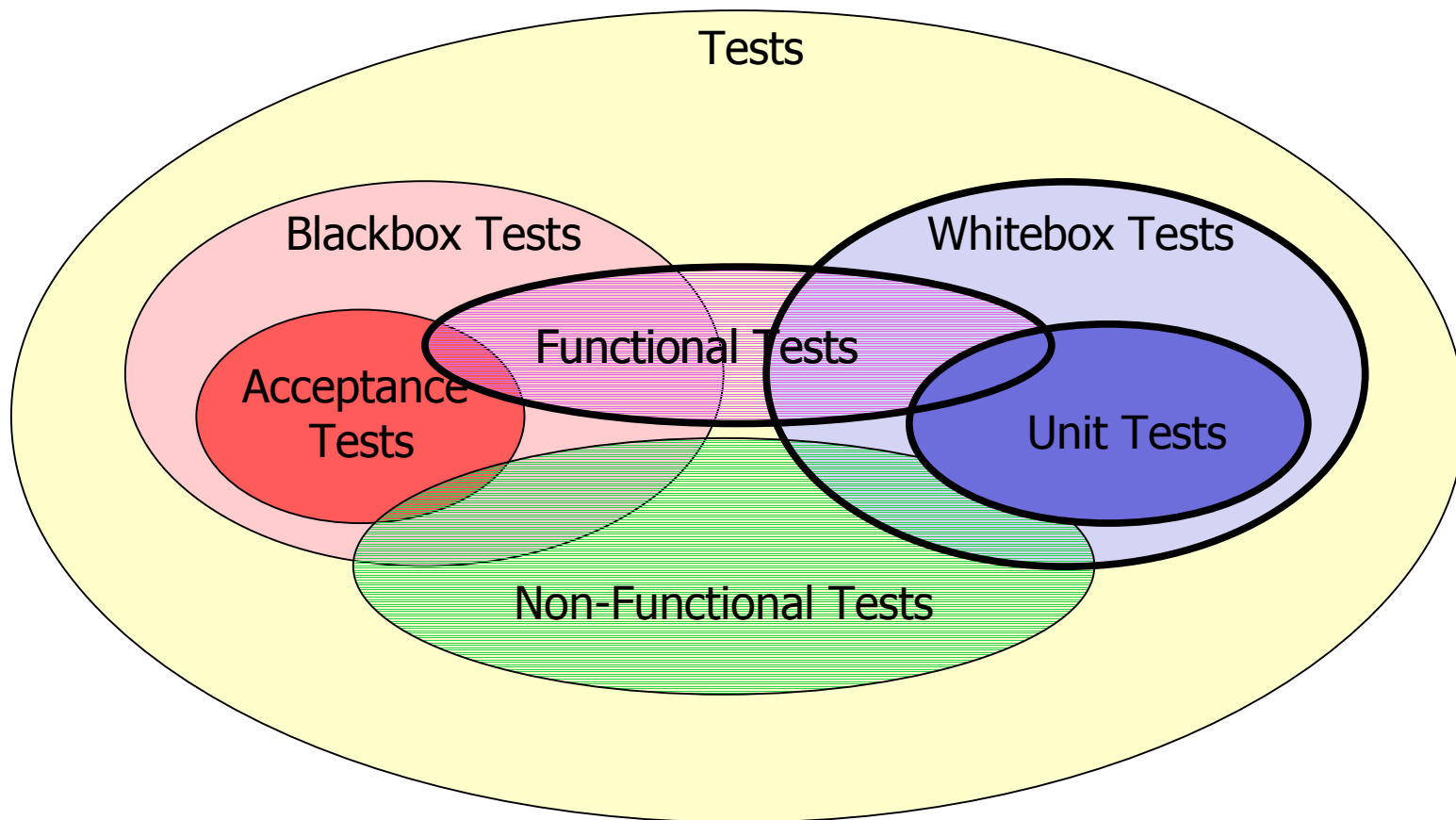
Franz-Josef Elmer

# Gliederung

- Begriffe und Konzepte
- JUnit: Unit Tests in Java
- JUnit Demo
- HttpUnit: Funktionale Tests von Web Sites
- HttpUnit Demo am Beispiel Chronos



# Begriffe



# Unit Test

- Automatisches Test Programm für
  - eine Funktion
  - ein Modul
  - eine Klasse
- Test liefert
  - OK oder
  - Fehlermeldung
- Test Suite
  - Gruppe von Tests
  - Test Suite ist selber ein Test

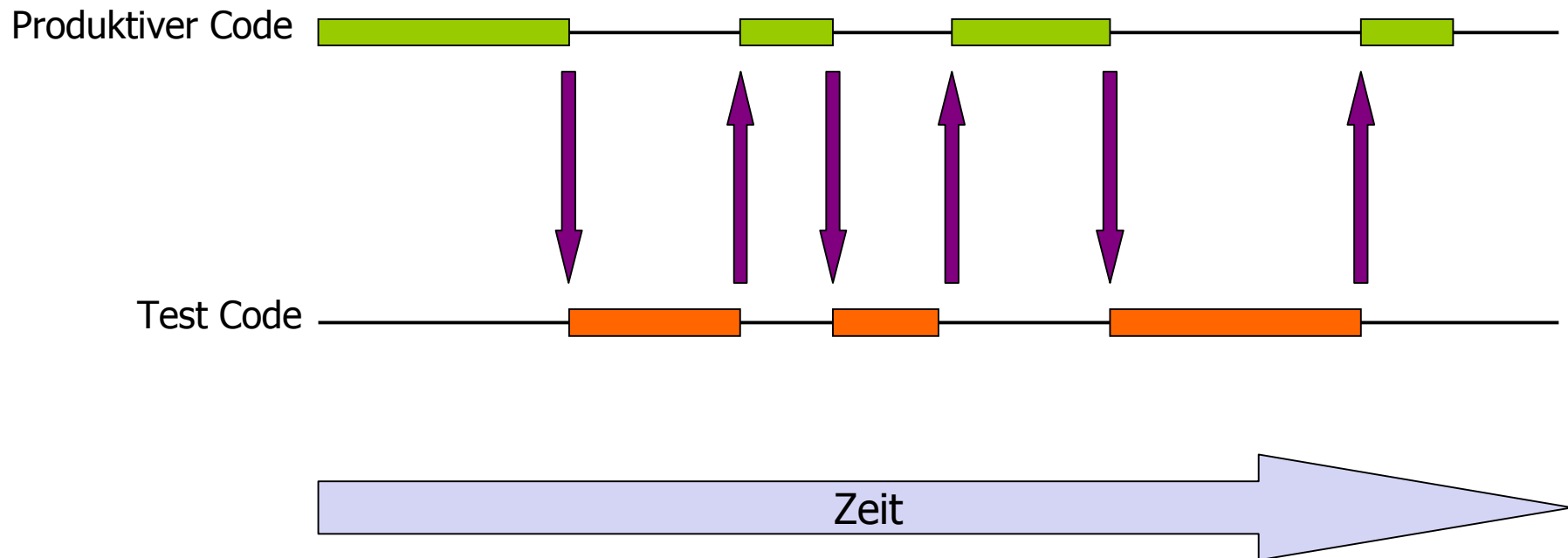
# Framework für Unit Tests

<b>Sprache</b>	<b>Framework</b>	<b>Homepage</b>
Java	JUnit	<a href="http://www.junit.org">www.junit.org</a>
C++	CppUnit	<a href="http://sourceforge.net/projects/cppunit">http://sourceforge.net/projects/cppunit</a>
C#	NUnit, csUnit	<a href="http://www.nunit.org">www.nunit.org</a> <a href="http://www.csunit.org">www.csunit.org</a>
PL/SQL	utPLSQL	<a href="http://utplsql.sourceforge.net/">http://utplsql.sourceforge.net/</a>

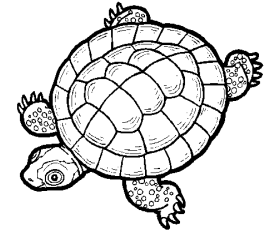
# Konzepte im Unit Testing

- "Code a little, test a little"
- Mock Objects
- Refactoring
- Test Driven Development

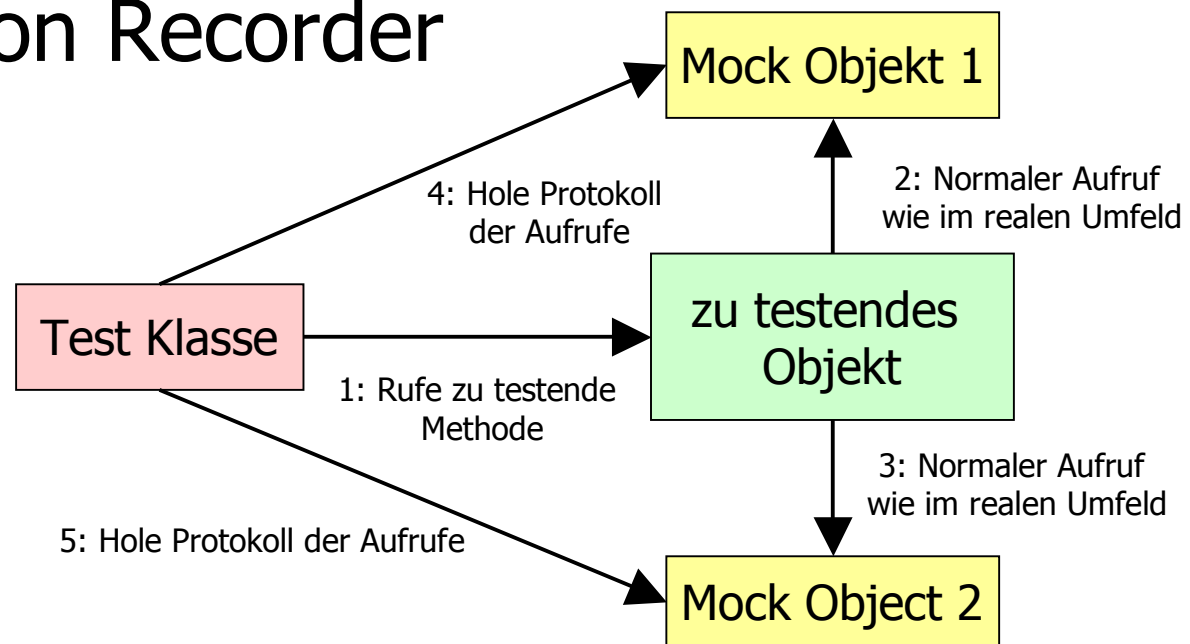
# Code a little, test a little



# Mock Objects

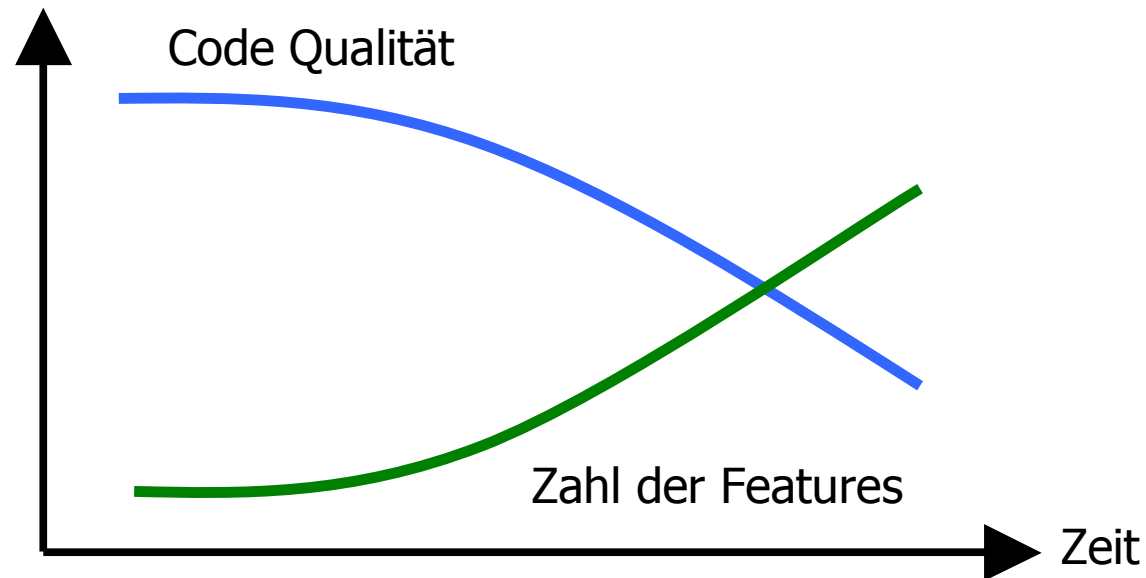


- Mock Turtle aus "Alice in Wonderland"
- Täuschen ein Umgebung vor
- Interaction Recorder



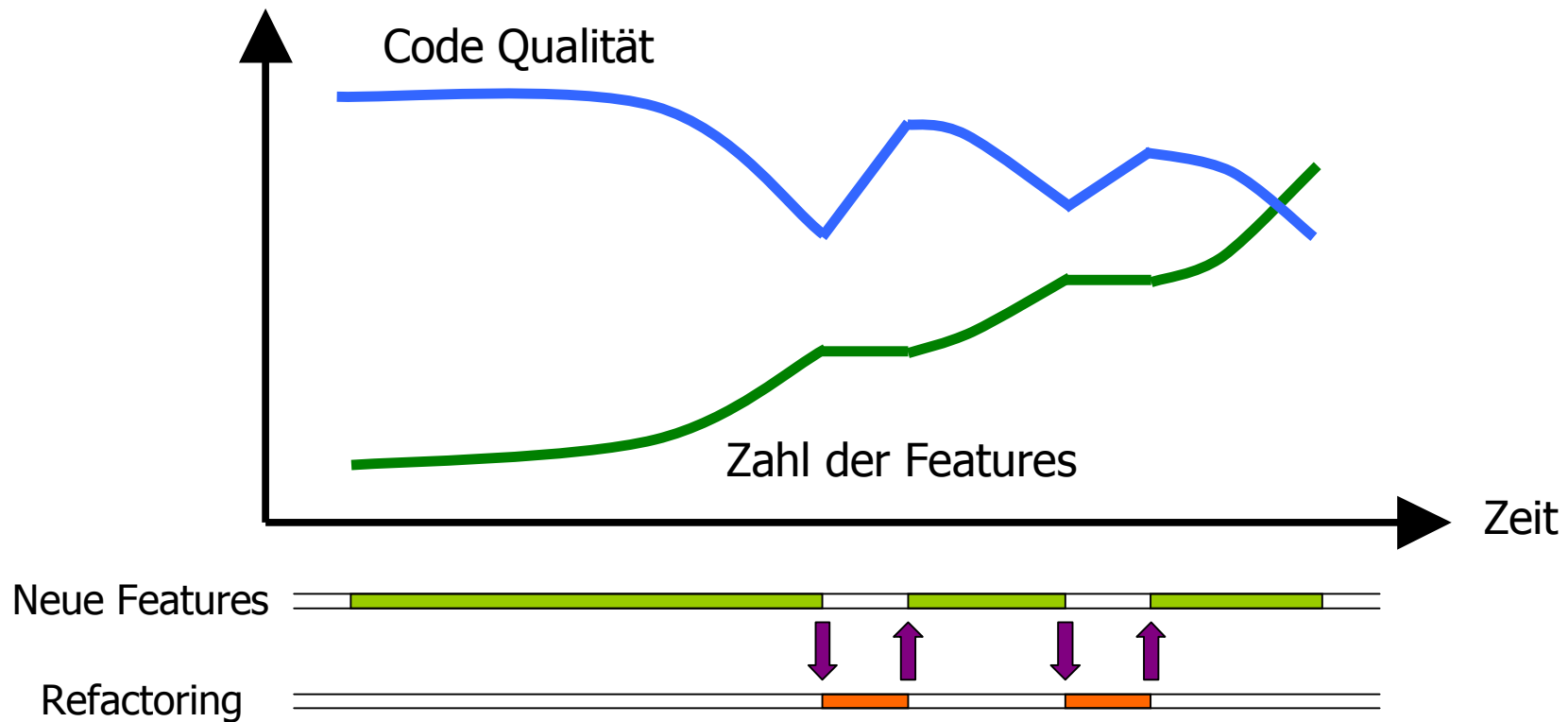
# Refactoring

- Allgemeine Erfahrung:



# Refactoring

- Code umbauen ohne neue Features zu implementieren

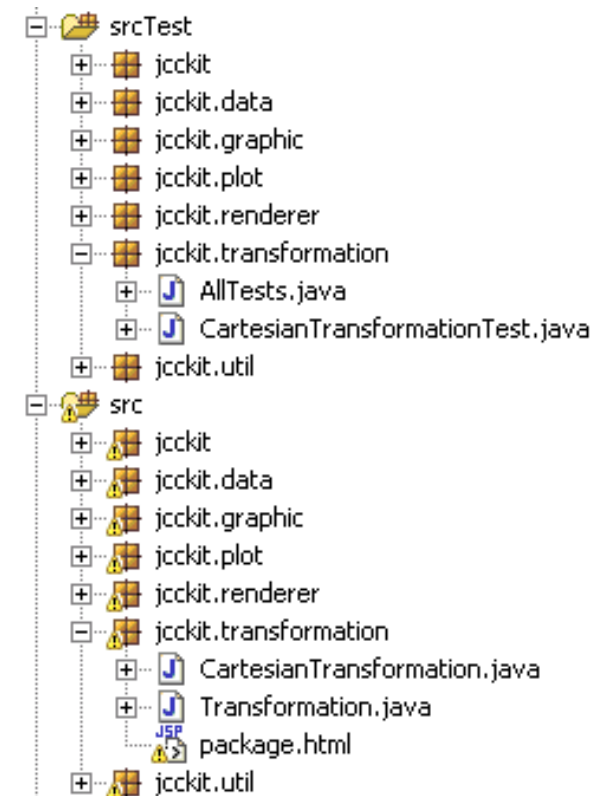


# Test Driven Development

- Testcode **vor** funktionalem Code schreiben
- Testcode ist funktionale Spezifikation
- Besseres Design des funktionalen Codes

# JUnit

- Basis Klasse: **TestCase**
- Testmethoden:  
`public void test<X>()`
- Namenskonvention:  
`<ClassToBeTested>Test`
- Test Container: **TestSuite**
- Namenskonvention:  
**AllTests**



# HttpUnit

- Java Bibliothek um JUnit Tests für Web Sites zu schreiben
- Klasse **WebConversation** simuliert einen Web Browser
- Klasse **WebResponse** repräsentiert eine HTML Page
- Beispiel:

```
WebResponse wr = new WebConversation().getResponse("http://www.google.com");
```

# HttpUnit

- Programmatischer Zugriff auf all HTML Elemente wie z.B.:
  - Links
  - Buttons
  - Textfelder
  - Auswahllisten
- Eingeschränkte JavaScript Unterstützung (keine DOM Manipulationen)