

Testgetriebene Entwicklung

Johannes Hackel und Falco Prescher

31. Oktober 2012

Gliederung

- 1 Allgemeines zur testgetriebenen Entwicklung
- 2 Vorgehensweise der testgetriebenen Entwicklung
- 3 Das Inversion Of Control im Zusammenhang mit Mocks
- 4 Testgetriebene Entwicklung in der Praxis

Allgemeines zur testgetriebenen Entwicklung

Entwicklungszyklus

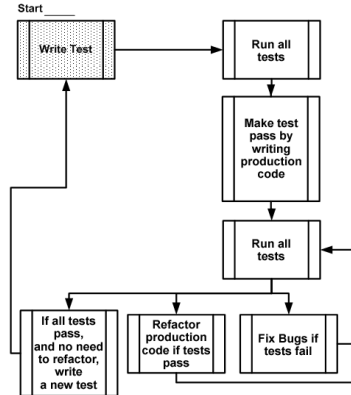


Abbildung: Der Entwicklungszyklus von Software mittels testgetriebener Entwicklung

Mocks

- Mocking - engl. für Vortäuschen
- Erstellung eines Versuchsobjektes durch eine vorgetäuschte Implementierung von Schnittstellen oder abstrakten Klassen
- Erzeugung des Objektes während der Laufzeit durch Mockframeworks
- Implementierung ohne Geschäftslogik
- Enthalten Konfigurations- und Testverifizierungsmöglichkeiten für Unit Tests
- Mockframeworks: Mockito¹ (Java) und Moq² (.NET)

¹<http://www.code.google.com/p/mockito/>

²<http://www.code.google.com/p/moq/>

Inversion Of Control

- kurz IoC - engl. für Steuerungsumkehrung
- Aufrufen von benutzerspezifische Methoden durch das Verhalten eines Softwareframeworks
- Dependency Injection (DI) = Spezialform des IoC
Möglichkeit zur Registrierung Schlüssel-Wertpaaren zu einem Container in einer Startsequenz (Bootstrap)
- Schlüssel = abstrakte Klasse oder Schnittstelle
Wert = Methode zur Rückgabe eines Objektes (Bsp. Konstruktoren oder Methoden)
- Inversion Of Control - Frameworks: PicoContainer³ (Java) und StructureMap⁴ (.NET)

³<http://www.picocontainer.codehaus.org/>

⁴<http://www.docs.structuremap.net/>

Inversion Of Control

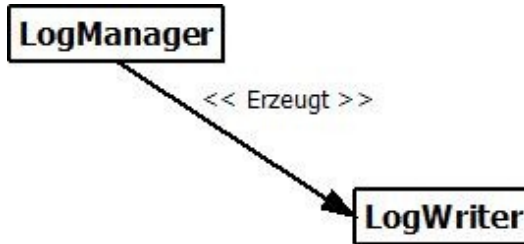


Abbildung: LogManager ist direkt abhängig von einer konkreten Implementierung von LogWriter

Inversion Of Control

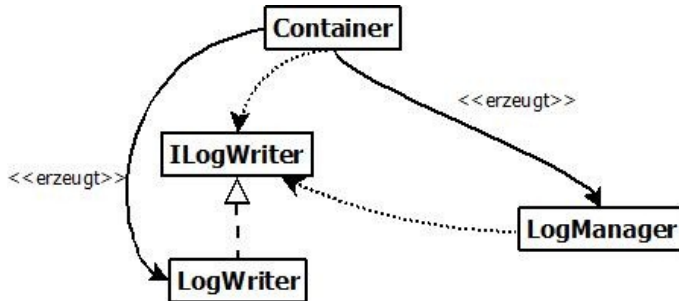


Abbildung: Der Container erzeugt Instanzen und Injiziert Abhängigkeiten

Kombination von Inversion Of Control mit Mocks

- IoC - Ermöglichen des Benutzens von Mocks zur Testzeit und Produktivcode zur Produktivzeit
- Mocks - Dadurch lose Kopplung von Programmkomponenten und Testen als einzelnes
- Ohne Inversion Of Control keine Mocks zum Testen nutzbar
- Ohne Mocks keine testgetriebene Entwicklung für große Programme möglich

siehe Beispiel

Quellen

- <http://www.codefest.at/post/2009/11/27/Design-Patterns-Teil-1-e28093-Inversion-of-Control-Dependency-Injection.aspx>
- <http://www.code.google.com/p/mockito/>
- <http://www.code.google.com/p/moq/>
- <http://www.junit.sourceforge.net/javadoc/overview-summary.html>
- <http://www.picocontainer.codehaus.org/>
- <http://www.docs.structuremap.net/>
- <http://www.webuser.hs-furtwangen.de/kaspar/seminar0607/TestDrivenDevelopment.pdf>
- OSHEROVE, Roy: the art of UNIT TESTING - with Examples in .NET. Manning Publications Co., 2009