



Fault Injection in C#

Patrick Siegler
Seminar Fault Injection SS15

Agenda

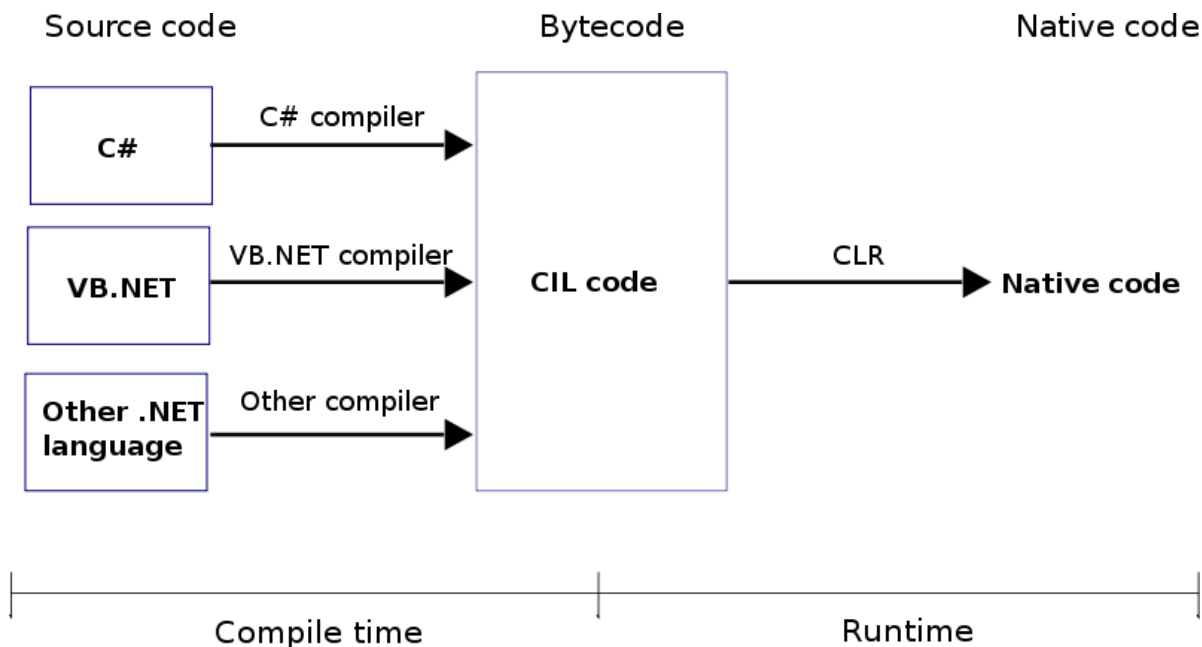
1. Motivation und Ansatz
2. Beispielanwendung
3. Mocks / Stubs
4. Shims / Moles
5. Vergleich

C# Fault Injection

Patrick Siegler
09.07.2015

Chart 2

- High-Level-Sprache mit Intermediate-Code
- Ermöglicht Reflection, z.B. Erstellung von Klassen zur Laufzeit



C# Fault Injection

Patrick Siegler
09.07.2015

Chart 3

Motivation

- Tests testen den erwarteten Normalfall
 - Evtl. einzelne Sonderfälle (ungültige Eingaben)
 - Beweisen Existenz von Fehlern, nicht Abwesenheit
- Coverage: wie Fehlerbehandlung testen?
- Test Suite beurteilen: Fehler, die nicht von Tests entdeckt werden?
- Robustness: Umgang mit unerwarteten Fehlern?

C# Fault Injection

Patrick Siegler
09.07.2015

Chart 4

Ansatz

- Fehlerinjektion ist Test des Umgangs mit Fehlern
 - “legitime” Fehler: Rückgabewerte von Bibliotheken, ungültige Eingabe
 - Fehler in der Programmlogik (Bugs)

- Vorbedingungen:
 - Fehlermodell: “Stürzt nicht ab, sondern gibt Fehler aus, wenn...”
 - Testfall: Normaler, legaler Referenz-Fall, der modifiziert wird

- I.A. 2 Ansätze
 - Quellcode verändern
 - Zustand verändern

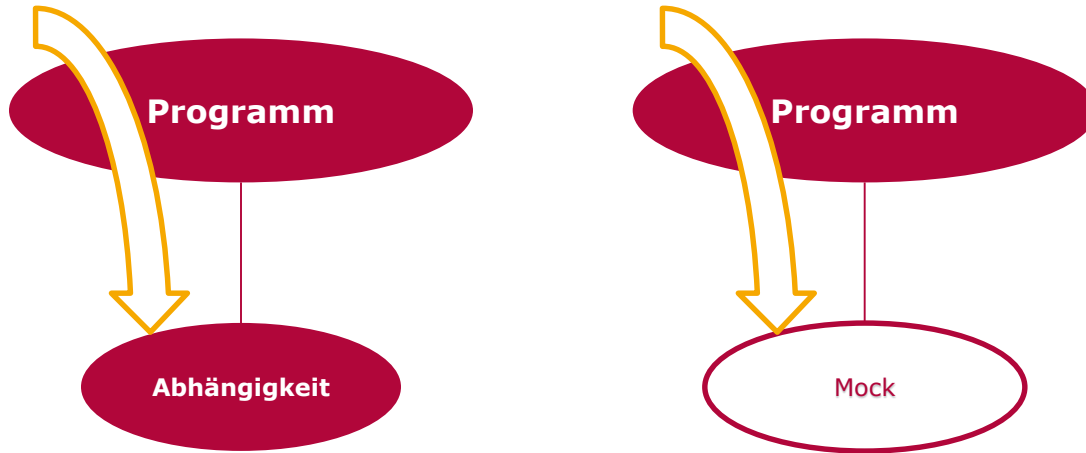
C# Fault Injection

Patrick Siegler
09.07.2015

Chart **5**

Mocks / Stubs

- Voraussetzung: Dependency Injection
- Anstatt Abhängigkeit ein Mock-Objekt übergeben
- Mocks zur Laufzeit durch Vererbung und Reflection erstellen



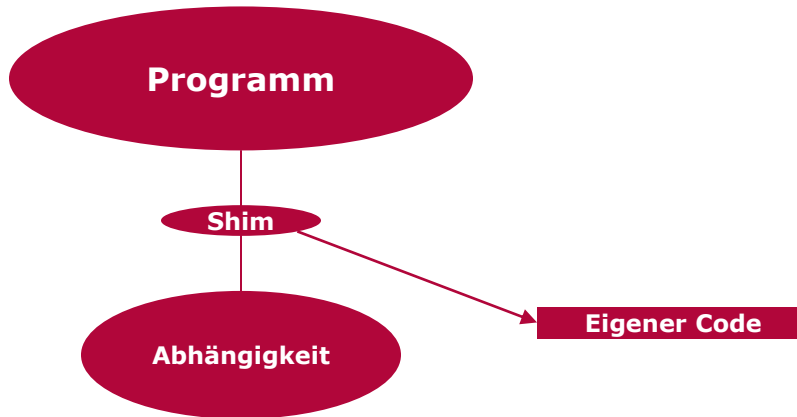
C# Fault Injection

Patrick Siegler
09.07.2015

Chart 6

Shims / Moles – TestApi

- Objekt normal verwenden, aber einzelne Aufrufe umleiten
- Neuer Prozess mit speziellem Profiler ermöglicht Eingriffe



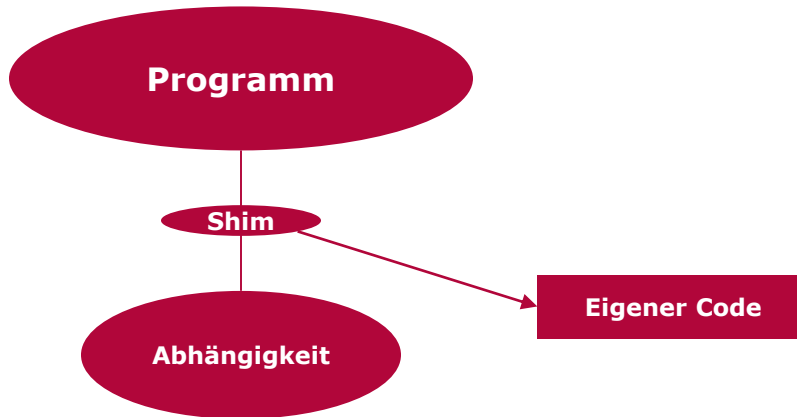
C# Fault Injection

Patrick Siegler
09.07.2015

Chart 7

Shims / Moles – MS Fakes

- Objekt normal verwenden, aber einzelne Aufrufe umleiten
- Integriert in Visual Studio, TestRunner hat selbst Profiler



C# Fault Injection

Patrick Siegler
09.07.2015

Chart 8

Shims vs Stubs

	TestApi	MS Fakes	Mocks
Quelltext notwendig	Nein	Nein	Eventuell
Veränderungen notwendig	Nein	Nein	Dependency Injection
Overhead	Hoch	Niedrig	Niedrig
Beobachtbare Systemgrenze	Prozess (I/O, User Interface)	Jede Klasse / Objekt	Jede Klasse / Objekt
Umsetzung	Shim: Profiler API	Shim: Profiler API Unterstützt auch Stubs	Stub: dynamische Vererbung
Grenzen	Startet neuen Prozess	Benötigt Visual Studio / TeamFoundation Server	Nur virtuelle Methoden von injizierten Objekten einer vererbten Klasse

C# Fault Injection

Patrick Siegler
09.07.2015

Chart 9

Fazit

- Fault Injection in VM-Sprachen durch Reflection und native APIs relative einfach
- Mocking
 - Bei passendem Quelltext am einfachsten
 - Keine nativen Abhängigkeiten
- TestApi
 - Stillstand seit 2011
 - Startet neuen Prozess mit Profiler
- Microsoft Fakes
 - Integriert in VS / TFS
 - Benötigt keinen neuen Prozess

C# Fault Injection

Patrick Siegler
09.07.2015

Chart **10**

Quellen

- Voas, Jeffrey. **"Software fault injection: Growing "safer" systems."** *IEEE AEROSPACE CONFERENCE PROCEEDINGS*. Vol. 2. 1997.
- Voas, Jeffrey. **"A Tutorial on Software Fault Injection."** *Spectrum* (2000).
- Tillmann, Nikolai, Jonathan de Halleux, and Tao Xie. **"Transferring an automated test generation tool to practice: From Pex to Fakes and Code Digger."** *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering*. ACM, 2014.
- Bieman, James M., Daniel Dreilinger, and Lijun Lin. **"Using fault injection to increase software test coverage."** *Software Reliability Engineering, 1996. Proceedings., Seventh International Symposium on*. IEEE, 1996.
- **Mock framework vs MS Fakes frameworks** <https://stackoverflow.com/questions/9677445/mock-framework-vs-ms-fakes-frameworks>
- Manolov, Ivo. **Introduction to TestApi – Part 5: Managed Code Fault Injection APIs** http://blogs.msdn.com/b/ivo_manolov/archive/2009/11/25/9928447.aspx
- Verissimo, Hamilton. **Castle's DynamicProxy for .NET** <http://www.codeproject.com/Articles/9055/Castle-s-DynamicProxy-for-NET>
- **Isolating Code Under Test with Microsoft Fakes** <https://msdn.microsoft.com/en-us/library/hh549175.aspx>

- Rhino Mocks: <http://www.hibernatingrhinos.com/oss/rhino-mocks>
- Castle Project: <http://www.castleproject.org/>
- TestApi Home: <https://testapi.codeplex.com/>

C# Fault Injection

Patrick Siegler
09.07.2015

Chart **11**

Thank you
for your attention!

Speaker
Job Description
Institute