

Lohnt sich TDD?

TDD steht für Test-Driven Development und besteht aus

(1) Test First

Zuerst den automatisierten Test schreiben, dann den Produktivcode.

(2) Refactoring

Die Veränderung der inneren Struktur einer Software, ohne das äußere Verhalten zu verändern. Erfordert eine gute Testabdeckung mit automatisierten Tests.

(3) Simplizität

Zustand, zu dessen Bestehen nur wenige Faktoren beitragen und deren Zusammenspiel durch wenige Regeln beschrieben sind. Kann erreicht werden durch Refactoring.

Lohnt sich Test First?

Ja, denn...

... automatisiertes Testen lohnt sich. Googles Investition in automatisierte Tests beschert eine Rendite von 60 %.

... ohne Test First werden Sie's schwer haben, automatisierte Tests aufzubauen; TDD bedeutet auch Test-Driven *Design*. Es drohen andernfalls

- ▶ Technische Schuld
- ▶ Big Ball of Mud



Lohnt sich Refactoring?

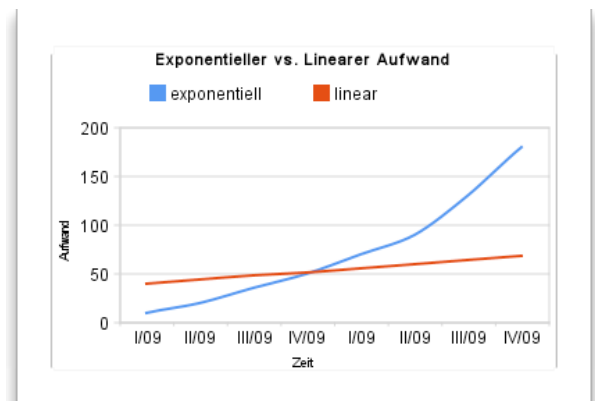
Ja, denn...

... ihre Software ist damit flexibel und anpassbar. Was sie sein muss, wenn sie iterativ und inkrementell vorgehen (beispielsweise bei Scrum).

... Refactoring ermöglicht → Simplizität.

Lohnt sich Simplizität?

Ja, denn damit haben Sie linearen statt exponentiellen Aufwand und damit einen viel längeren Atem.



Fazit: Lohnt sich's?

Ja, denn...

... automatisierte Tests lohnen sich, werden unterstützt von Test First, was wiederum Refactoring ermöglicht, ohne dass Simplizität nur äußerst schwer herstellbar ist.

Quellen:

- Gojko Adzic: "Improving testing practices at Google"
- Michael Hill: "How TDD and Pairing increase production"
- Martin Fowler: "TechnicalDebt"
- Martin Fowler: "FlaccidScrum"
- Wikipedia: Einfachheit

Noch Fragen?

- Gerne an info@it-agile.de

Vortragende:

- Johannes Link (@johanneslink), Freiberufler
- Bernd Schiffer (@berndschiffer), it-agile GmbH

