

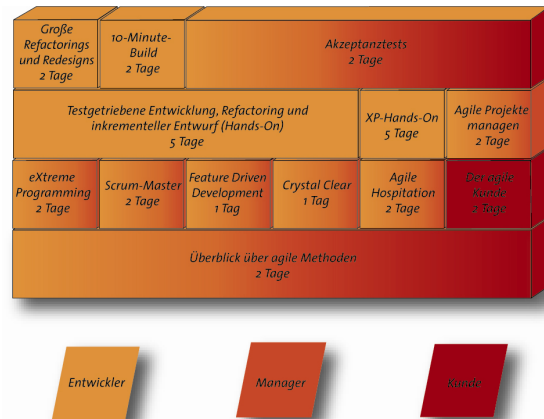
## Agil mit FDD (Feature Driven Development)



## Whitepaper



Unter der Marke *it-agile* bieten wir Beratung und Schulungen zu agilen Themen an. Weitere Informationen unter [www.it-agile.de](http://www.it-agile.de)



### Motivation

Agile Vorgehensweisen sind in aller Munde:

- Der regelmäßig erscheinende und vielzitierte CHAOS-Report nennt agile Vorgehensweisen explizit als wichtigen Erfolgsfaktor für Softwareprojekte.
- Das Vorgehensmodell des Bundes V-Modell XT unterstützt agile Vorgehensweisen.
- Microsoft und SAP setzen die agile Methode *Scrum* ein.
- Google geht nach einem eigenen agilen Vorgehensmodell vor.
- IBM nutzt für die Entwicklung der Eclipse-IDE ein eigenes agiles Vorgehen.

Dabei konzentriert sich die Diskussion heute auf eXtreme Programming (XP) und Scrum. Feature Driven Development (FDD) ist in Deutschland weniger bekannt, bietet aber sehr interessante Ansätze insbesondere für eher klassisch organisierte Unternehmen. Dieser Newsletter gibt eine Einführung in FDD und zeigt die wesentlichen Unterschiede zu XP und Scrum auf.

### FDD-Überblick

FDD wurde als schlanke Methode von Jeff De Luca im Jahre 1997 definiert, um ein großes zeitkritisches Projekt (15 Monate, 50 Entwickler) durchzuführen. Seitdem wurde FDD kontinuierlich weiterentwickelt. FDD stellt den Feature-Begriff in den Mittelpunkt der Entwicklung. Jedes Feature stellt einen Mehrwert für den Kunden dar.

Die Entwicklung wird anhand eines Feature-Plans organisiert. Eine wichtige Rolle spielt der Chefarchitekt (engl. Chief Architect), der ständig den Überblick über die Gesamtarchitektur und die fachlichen Kernmodelle behält. Bei größeren Teams werden einzelne Entwicklerteams von Chefprogrammierern (engl. Chief Programmer) geführt.

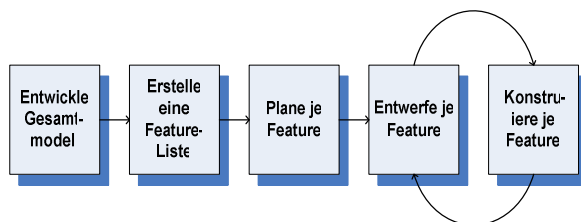
FDD definiert ein Prozess- und ein Rollenmodell, die gut mit existierenden klassischen Projektstrukturen harmonieren. Daher fällt es vielen Unternehmen leichter, FDD einzuführen als XP oder Scrum. Außerdem ist FDD ganz im Sinne der agilen Methoden sehr kompakt. Es lässt sich auf 10 Seiten komplett beschreiben.



## FDD-Prozessmodell

FDD-Projekte durchlaufen fünf Prozesse.

- Prozess #1: Entwickle ein Gesamtmodell  
Rollen: alle Projektbeteiligte
- Prozess #2: Erstelle eine Feature-Liste  
Rollen: i.d.R. nur die Chefprogrammierer
- Prozess #3: Plane je Feature  
Rollen: Projektleiter, Entwicklungsleiter, Chefprogrammierer
- Prozess #4: Entwerfe je Feature  
Rollen: Chefprogrammierer, Entwickler
- Prozess #5: Konstruiere je Feature  
Rollen: Entwickler



Diese Abfolge der Prozesse sieht auf den ersten Blick wasserfallartiger aus, als sie in der Praxis ist. Zum einen werden für die ersten drei Prozesse nur kurze Zeiträume investiert (häufig nur wenige Tage). Zum anderen werden die Prozesse 4 und 5 in ständigem Wechsel durchgeführt, weil jedes Feature in maximal zwei Wochen realisiert wird.

### Prozess #1: Entwickle ein Gesamtmodell

Im ersten Prozess definieren Fachexperten und Entwickler unter Leitung des Chefarchitekten Inhalt und Umfang des zu entwickelnden Systems. In Kleingruppen werden Fachmodelle für die einzelnen Bereiche des Systems erstellt, die in der Gruppe vorgestellt, ggf. überarbeitet und schließlich integriert werden. Das Ziel dieser ersten Phase ist ein Konsens

über Inhalt und Umfang des zu entwickelnden Systems sowie das fachliche Kernmodell.

### Prozess #2: Erstelle eine Feature-Liste

Im zweiten Prozess detaillieren die Chefprogrammierer die im ersten Prozess festgelegten Systembereiche in Features. Dazu wird ein dreistufiges Schema verwendet: Fachgebiete (engl. Subject Areas) bestehen aus Geschäftstätigkeiten (engl. Business Activities), die durch Schritte (engl. Steps) ausgeführt werden. Die Schritte entsprechen den Features. Die Features werden sehr prägnant nach dem einfachen Schema *<Aktion> <Ergebnis> <Objekt>* aufgeschrieben, z.B. „Berechne Gesamtsumme der Verkäufe“. Ein Feature darf maximal zwei Wochen zu seiner Realisierung benötigen.

Das Ergebnis dieser zweiten Phase ist eine kategorisierte Feature-Liste, deren Kategorien auf oberster Ebene von den Fachexperten aus Phase 1 stammen.

### Prozess #3: Plane je Feature

Im dritten Prozess planen der Projektleiter, der Entwicklungsleiter und die Chefprogrammierer die Reihenfolge, in der Features realisiert werden sollen. Dabei richten sie sich nach den Abhängigkeiten zwischen den Features, der Auslastung der Programmiererteams sowie der Komplexität der Features. Auf Basis des Plans werden die Fertigstellungstermine je Geschäftsaktivität festgelegt. Jede Geschäftsaktivität bekommt einen Chefprogrammierer als Besitzer zugeordnet. Außerdem werden für die bekannten Kernklassen Entwickler als Besitzer festgelegt (engl. Class Owner List).

### Prozess #4: Entwerfe je Feature

Im vierten Prozess weisen die Chefprogrammierer die anstehenden Features Entwicklerteams auf Basis des



Klassenbesitzums zu. Die Entwicklerteams erstellen ein oder mehrere Sequenzdiagramme für die Features und die Chefprogrammierer verfeinern die Klassenmodelle auf Basis der Sequenzdiagramme. Die Entwickler schreiben dann erste Klassen- und Methodenrumpfe. Schließlich werden die erstellten Ergebnisse inspiziert. Bei fachlichen Unklarheiten können die Fachexperten hinzugezogen werden.

### Prozess #5: Konstruiere je Feature

Im fünften Prozess programmieren die Entwickler die im vierten Prozess vorbereiteten Features. Bei der Programmierung werden Komponententests und Code-Inspektionen zur Qualitätssicherung eingesetzt.

Außerdem bietet FDD elegante Strukturierungsmöglichkeiten für Anforderungen. Durch die im Gegensatz zu XP oder Scrum detailliertere Planung kann FDD tendenziell besser mit Teams umgehen, die sehr heterogen qualifiziert sind.

### Wann FDD?

Die Voraussetzung für FDD ist ein moderates Tempo für Anforderungsänderungen. Das ist i.d.R. der Fall bei Produktentwicklung und bei Projekten, die relativ lange fachliche Analysephasen durchlaufen (z.B. zur Erstellung von Lastenheften oder Ausschreibungen). Ist das Projektteam groß und heterogen, wird eine Selbstorganisation mit XP oder Scrum mitunter schwierig. FDD bietet in diesen Konstellationen durch sein Rollenmodell eine gesunde Basis für ein diszipliniertes Vorgehen.

### Bewertung

FDD wirkt in seiner Beschreibung wasserfallartig: einzelne Phasen folgen sequenziell hintereinander und ein freier Wechsel zwischen den Phasen ist nicht vorgesehen.

Allerdings sind die einzelnen Phasen sehr kurz, so dass FDD dennoch Reaktionen auf veränderte Randbedingungen und Anforderungen erlaubt. Allerdings kann FDD nicht ganz so schnell reagieren, wie z.B. XP oder Scrum. Der höhere Planungsaufwand macht FDD etwas schwerfälliger, allerdings auch weniger anfällig für bestimmte Störungen im Entwicklungsprozess (z.B. „Kunde entscheidet sich ständig um“).

### Referenzen

- Scrum und eXtreme Programming: siehe <http://www.wikipedia.de>
- Wolf, Rook, Lippert: „eXtreme Programming“, dpunkt, 2. Auflage. Das Buch beschreibt auch FDD und setzt es in Beziehung zu anderen agilen Methoden.
- FDD-Beschreibung auf 10 Seiten: <http://www.nebulon.com/articles/fdd/download/fddprocessesA4.pdf>
- <http://www.featuredrivendevelopment.com/>
- Schulung und Coaching zu FDD in Deutschland: <http://fdd.it-agile.de>
- Web-Site von Jeff De Luca: <http://www.nebulon.com>



Agil mit FDD  
**Whitepaper**

## Unser Angebot

Unter unserer Beratungsmarke für agile Methoden **it-agile** bieten wir Schulungen und Coaching zu FDD an:

- <http://www.it-agile.de>



Feature Driven Development

Am 8. und 9. Mai 2007 bieten wir eine vom FDD-Vater Jeff De Luca durchgeführte FDD-Schulung mit Zertifizierung in Hamburg an.



Nähere Informationen finden sie unter

- <http://fdd.it-agile.de>

## FDD auf einen Blick

- Agil und strukturiert.
- Definierte Prozesse zum Anforderungsmanagement.
- Features als zentrales Mittel der Projektplanung und Entwicklung.
- Geeignet für heterogene Teams.
- Auch für große Projekte einsetzbar.

Weitere Informationen erhalten Sie gerne unter [agile@akquinet.de](mailto:agile@akquinet.de)

