

Mit Ordnung zum Ziel

In dieser Artikelserie wird mit Feature Driven Development (FDD) eine agile Methode beschrieben, die explizit eine vorgelagerte Analyse- und Modellierungsphase vorsieht und für große sowie für Festpreisprojekte gut geeignet ist.

von Stefan Rook und Henning Wolf

Im ersten Teil der vierteiligen Artikelserie zu Feature Driven Development (FDD) soll ein Überblick über die Prozesse im FDD gegeben werden. Diese Prozesse werden in den folgenden Artikeln der Serie im Detail vorgestellt. Im Gegensatz zu Scrum oder eXtreme Programming (XP) ist FDD jedoch weit weniger bekannt und zumindest in Europa auch in der Praxis wenig verbreitet. Das ist insofern ein änderungswürdiger Zustand, als das FDD interessante andere Schwerpunkte setzt als Scrum oder XP.

Gerade dadurch bietet FDD Antworten auf viele Fragen, die bei Scrum oder XP offen bleiben. Auch wenn Ihr Projekt nicht nur FDD wird, können Sie nützliche Anregungen und Techniken aus FDD mitnehmen und für Ihre Projektsituation verwenden.

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. Features sind sehr klein, für den Kunden verständlich und stellen jeweils einen Mehrwert für den Kunden dar. Die Entwicklung wird anhand des 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, das gut mit existierenden klassischen Projektstrukturen harmonisiert. Daher fällt es vielen Unternehmen leichter, FDD einzuführen als XP oder Scrum. Außerdem ist FDD ganz im Sinne der agilen Methoden sehr kompakt und es lässt sich auf 10 Seiten komplett beschreiben [1].

FDD-Prozessmodell

FDD-Projekte durchlaufen fünf Prozesse (Abb. 1):

- **Prozess #1:** Entwickle ein Gesamtmodell, Rollen: alle Projektbeteiligte
- **Prozess #2:** Erstelle eine Feature-Liste, Rollen: in der Regel nur die Chefprogrammierer

Navigator FDD-Artikelserie

Teil 1: Mit Ordnung zum Ziel

Teil 2: Planung mit Featurelisten

Teil 3: Entwurf und Konstruktion der Features

Teil 4: FDD-Projektmanagement

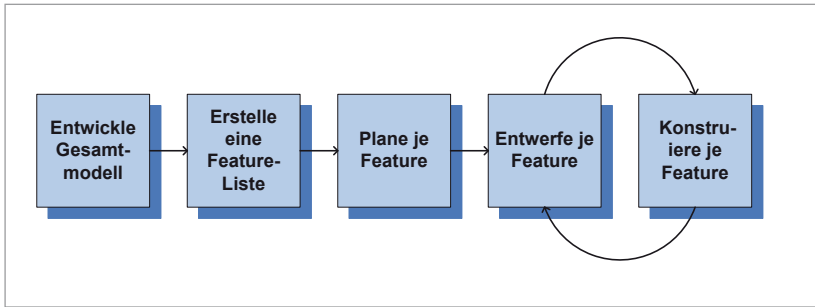


Abb. 1: Die fünf FDD-Prozesse

- **Prozess #3:** Plane je Feature, Rollen: Projektleiter, Entwicklungsleiter, Chefprogrammierer
- **Prozess #4:** Entwurf 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 ein bis zwei Wochen investiert. Zum anderen werden die Prozesse 4 und 5 in ständigem Wechsel durchgeführt, weil jedes Feature in maximal zwei Wochen realisiert wird. Ein Gesamtprojekt dauert maximal sechs Monate. Ist das Vorhaben größer, müssen mehrere Projekte initiiert werden.

Entwickle ein Gesamtmodell

Im ersten Prozess definieren Fachexperten und Entwickler unter Leitung des Chefarchitekten Inhalt und Umfang des zu entwickelnden Systems. Der Begriff Chefarchitekt ist etwas irreführend, weil er weniger eine vorgebende Rolle hat, sondern vielmehr eine moderierende. In zukünftigen Versionen des FDD wird die Rolle wahrscheinlich nach Chef-Modellierer umbenannt werden. 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.

Dabei wird in Kleingruppen modelliert, die jeweils aus einem Fachexperten und zwei bis drei Entwicklern bestehen. Die Ergebnisse der Kleingruppen werden immer wieder im Plenum vorgestellt und integriert. Die Modellierung erfolgt

parallel, sodass bei der Integration ein möglichst gutes Gesamtmodell entsteht. Wenn das Projekt sehr groß ist, können nicht mehr alle Programmierer komplett an den Modellierungswshops teilnehmen. In diesem Fall sind die Chefprogrammierer während der ganzen Modellierung dabei. Die anderen Programmierer sind jeweils zeitweise dabei. Jeder spätere Projektbeteiligte soll zumindest zeitweise an der Modellierung teilhaben.

Bei der Modellierung wird in der Regel ein spezieller Ansatz namens Modeling in Color verwendet [2], der die fachlichen Klassen in vier Archetypen unterteilt: *PartyPlaceThing*, *Description*, *Role* und *Momentum-Intervall*. Modeling in Color legt auch fest, wie die Archetypen miteinander verknüpft werden dürfen. Auf den ersten Blick mag diese Art der Modellierung zu rigide erscheinen. In der Praxis führt Modeling in Color allerdings sehr schnell zu übersichtlichen und leicht verständlichen Entwürfen. Seiteneffekt der Standardisierung der Modellierung ist außerdem, dass sich Entwürfe verschiedener Teams sehr einfach integrieren lassen. Dies ist eine wichtige Voraussetzung für große Projekte. Die Modellierung bleibt dabei stets leichtgewichtig: Modelliert wird auf Flipcharts. Die Änderbarkeit wird durch den Einsatz von Post-Its sichergestellt. Am Ende der Modellierung werden die erstellten Modelle in der Regel in ein UML-Tool übertragen. Modeling in Color wird im Rahmen dieser Serie nicht mehr im Detail vorgestellt, weil es bereits in der Ausgabe 12.2007 des *Java Magazins* vorgestellt wurde [7].

Erstelle eine Feature-Liste

Im zweiten Prozess detaillieren die Chefprogrammierer die im ersten Prozess festgelegten Fachgebiete 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 Hierarchisierung zeigt Abbildung 2. Die Features werden sehr prägnant nach dem einfachen Schema $\langle \text{Aktion} \rangle \langle \text{Ergebnis} \rangle \langle \text{Objekt} \rangle$ aufgeschrieben, z.B. „Berechne Gesamtsumme der Verkäufe“. Ein Feature darf maximal zwei Wochen bis zu seiner Realisierung benötigen. Die meisten Features dauern jedoch nur wenige Stunden. Das Ergebnis dieses zweiten Prozesses ist eine kategorisierte Feature-Liste, deren Kategorien auf oberster Ebene die Fachgebiete von den Fachexperten aus Prozess #1 sind.

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*). Auf dieser Basis werden Feature-Pakete geschnürt und durch temporäre Feature-Teams realisiert.

Entwurf je Feature

Im vierten Prozess weisen die Chefprogrammierer die anstehenden Features Entwicklerteams auf Basis des Klassenbesitzums zu. Die Entwicklerteams erstellen gemeinsam ein oder mehrere

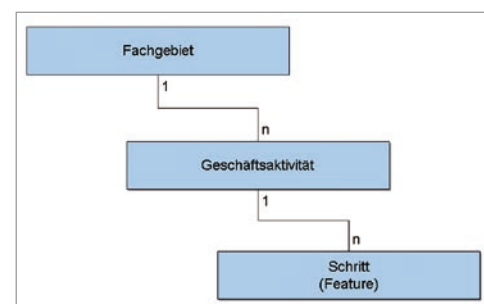


Abb. 2: Anforderungsorganisation in FDD

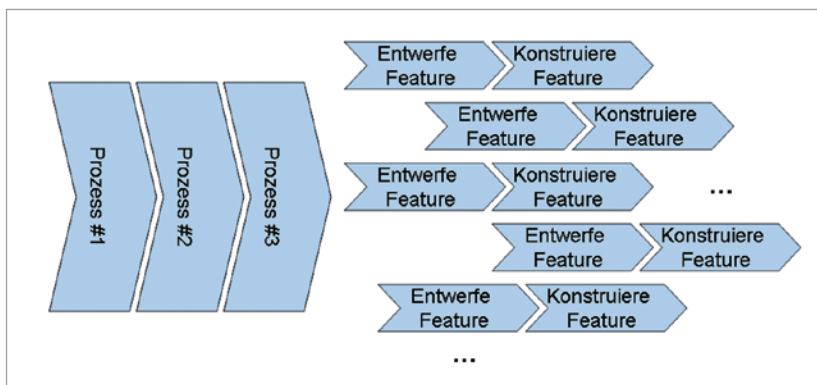


Abb. 3: Features werden parallel entwickelt

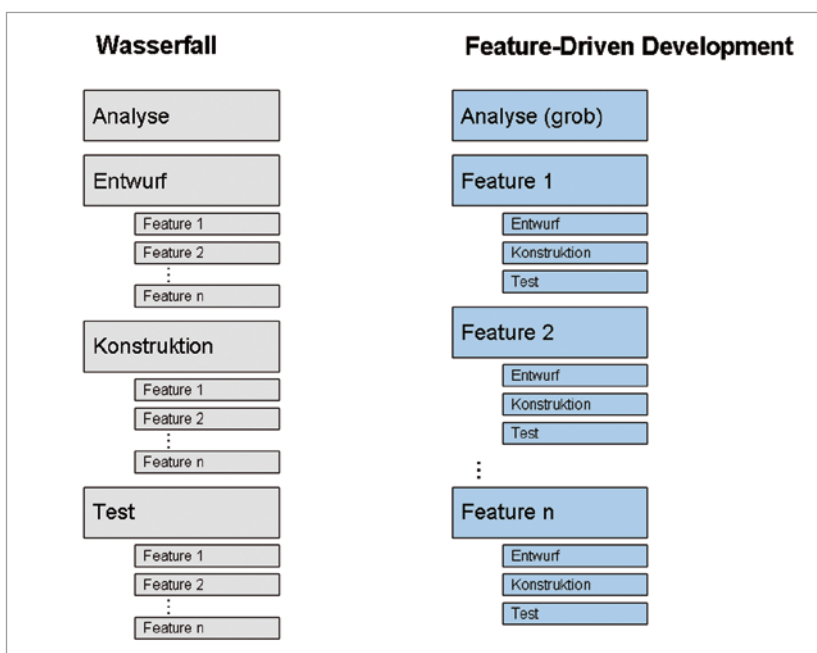


Abb. 4: Schematischer Vergleich Wasserfall und FDD

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.

Konstruiere je Feature

Im fünften Prozess programmieren die Entwickler die im vierten Prozess vorbereiteten Features. Bei der Programmierung werden Code-Inspektionen und Komponententests zur Qualitätssicherung eingesetzt. Der Fokus liegt allerdings eindeutig bei den Code-Inspektionen. FDD bezieht sich dabei auf wissenschaft-

liche Untersuchungen, dass Code-Inspektionen effektiver bei der Fehlerbeseitigung sind als Tests. Die Komponententests sind ein zusätzliches Sicherheitsnetz. Testgetriebene Entwicklung ist nicht vorgesehen. Die Prozesse 4 und 5 werden je Feature nacheinander durchlaufen, sodass sie in der Praxis ineinander übergehen. Damit laufen in einem Projekt in der Regel parallel viele dieser Prozesse, weil Features parallel abgearbeitet werden.

Im Gegensatz zu Scrum oder XP fällt auf, dass es für Entwurf und Konstruktion der Features kein Iterationskonzept gibt. Das ist der vorgelagerten Modellierung geschuldet. Eine kundengetriebene Umplanung der Inhalte würde wieder bei Prozess #1 beginnen. Eine inkrementelle Auslieferung ist bei passender zeitlicher

Planung der Einzelfeatures jedoch problemlos möglich und auch erwünscht.

Festpreise

FDD wird sehr häufig für Festpreisprojekte eingesetzt. In der vorgelagerten Modellierung (Prozesse #1 und #2) definieren Auftraggeber und Auftragnehmer gemeinsam den Umfang des Projekts. Damit entsteht zum einen ein gemeinsames Verständnis des Projekthinhalts, zum anderen bekommt der Auftragnehmer ausreichend detaillierte Informationen, um eine belastbare Aufwandsschätzung zu erstellen. Daher werden die ersten beiden Prozesse in der Regel zuerst beauftragt und das eigentliche Projekt erst danach angeboten und beauftragt. Dann wird der Projekthinhalt über die Prozesse #3 bis #5 realisiert. Sollen während der Realisierung größere Änderungen an den Anforderungen durchgeführt werden, ist dies mit einigem Aufwand und Overhead verbunden. Das ist für Festpreiskonstellationen aber nicht besonders schädlich. Der Vertrag würde ohnehin hinfällig und neue Verträge oder Change-Requests müssen initiiert werden. Festpreisprojekte gehen genauso wie FDD davon aus, dass am vereinbarten Leistungsumfang nur wenig gerüttelt wird. Für die Aufwandsschätzung gilt als Daumenregel, dass eine Modellierungsdauer von einer Woche ein Dreimonatsprojekt nach sich zieht und eine Modellierungsdauer von zwei Wochen ein Sechsmonatsprojekt.

Bewertung

FDD wirkt in seiner Beschreibung zunächst wasserfallartig: einzelne Phasen folgen sequenziell hintereinander und ein freier Wechsel zwischen den Phasen ist nicht vorgesehen. Abbildung 4 zeigt, dass es dennoch erhebliche Unterschiede zwischen Wasserfall und FDD gibt.

Außerdem sind die einzelnen FDD-Prozesse sehr kurz und die erstellten Artefakte leichtgewichtig, sodass FDD dennoch Reaktionen auf veränderte Randbedingungen und Anforderungen erlaubt. Aufgrund der vorgelagerten Modellierung kann FDD dabei nicht ganz so schnell reagieren wie z.B. XP oder Scrum. Potenziell muss man bei jeder Änderung zurück in den ersten Prozess und die Mo-

dellierung wiederholen. Faktisch ist das nur bei Änderungen notwendig, die den Kern der bisherigen Modellierung infrage stellen. Der höhere initiale Modellierungs- und Planungsaufwand macht FDD also etwas schwerfälliger, allerdings auch weniger anfällig für bestimmte Störungen im Entwicklungsprozess (z.B. „Kunde entscheidet sich ständig um“). Wer besonders schnell auf Veränderungen reagieren können will, ist mit Scrum oder XP meist besser bedient als mit FDD. Wer eine moderate Veränderungsfrequenz hat (z.B. bei Projekten mit klassischen Festpreisverträgen oder bei gut planbarer Produktentwicklung), sollte sich FDD im Detail ansehen. FDD macht konkrete Vorgaben dazu, wie Anforderungen aufzuschreiben und zu strukturieren sind und gibt eine deutliche Empfehlung, die Modellierung mit Modeling in Color durchzuführen. Es gibt damit in Bereichen Anleitung, in denen die meisten anderen agilen Methoden sich ausschweigen. FDD setzt weniger stark auf Selbststeuerung des Entwicklungsteams als dies bei XP und insbesondere Scrum der Fall ist. FDD strukturiert stärker vor, wer in welcher Rolle wann welche Aufgaben hat. Damit kann FDD die Potenziale selbstgesteuerter Teams nicht vollständig ausschöpfen, geht aber auch nicht deren Risiken ein.

Übertragung

Auch wenn FDD nicht die geeignete Methode für das eigene Projekt ist, dürften viele Projekte dennoch von FDD-Techniken profitieren können. Das Konzept der vorgelagerten kurzen Modellierung wird natürlich auch von Scrum- und XP-Projekten verwendet. In der Scrum- und XP-Literatur wird dieses Thema aber leider bestenfalls am Rande erwähnt. Modeling in Color für die initiale Modellierung lässt sich problemlos auch in anderen Methoden verwenden. Wir haben vielversprechende Erfahrungen mit dem Ansatz in XP-Projekten gemacht, haben die Modelle dann aber zunächst nur sehr grob entwickelt und über die Zeit inkrementell detailliert. Die Art und Weise der Anforderungsdefinition funktioniert ebenfalls unabhängig von FDD. Insbesondere das starre Beschreibungsschema für Features führt zu Klarheit, weniger Missverständnissen und sehr kleinen Features. Weil die

Features so klein sind, ist allen Beteiligten in der Regel auch klar, was Bestandteil des Features ist und was nicht mehr. Außerdem müssen die Features wegen ihrer Größe häufig gar nicht mehr explizit geschätzt werden. Es reicht aus, sie zu zählen. Dieses Thema werden wir im vierten Teil dieser Artikelserie noch einmal aufgreifen. Ebenfalls im vierten Teil der Artikelserie werden wir das *Parking-Lot-Diagramm* für das Tracking kennenlernen, das auch in Nicht-FDD-Projekten nützlich ist.

Ausblick

In Teil 2 dieser Artikelserie werden die Featurelisten im Mittelpunkt stehen. Es wird beschrieben, wie die Features organisiert und auf Feature-Teams zugewiesen werden. Teil 3 nimmt sich der letzten beiden FDD-Prozesse an und beschreibt, wie die Features entworfen und programmiert werden. Der vierte und letzte Teil der Artikelserie zeigt, wie das Projektmanagement im Feature Driven Development abläuft und welche Steuerungs- und Controllinginstrumente verwendet werden. ■



Stefan Rook und Henning Wolf sind Berater bei der akquinet it-agile GmbH. Sie arbeiten als Entwickler, Projektleiter und agile Methodenberater. Sie haben seit 1998 Erfahrungen mit agilen Methoden. Über XP haben sie ein Erfahrungsbuch geschrieben, in Scrum und FDD sind sie zertifiziert.



Links & Literatur

- [1] FDD-Beschreibung auf 10 Seiten: www.nebulon.com/articles/fdd/download/fddprocessesA4.pdf
- [2] Peter Coad, Eric Lefebvre, Jeff De Luca: Java Modeling in Color with UML, Prentice-Hall, 1999
- [3] Henning Wolf, Stefan Rook, Martin Lippert: eXtreme Programming, 2. Aufl., dpunkt, 2005
- [4] Webseite zu FDD: www.featuredrivendevelopment.com
- [5] Webseite von Jeff De Luca: www.nebulon.com
- [6] Schulung und Coaching zu FDD in Deutschland: fdd.it-agile.de
- [7] Thilo Fotscher: Mit Color Modeling farbige Domänenmodelle erstellen, in: *Java Magazin* 12.2007, S. 37–42

Verlag:
Software & Support Verlag GmbH

Anschrift der Redaktion:
Java Magazin
Software & Support Verlag GmbH
Geleitsstraße 14
D-60599 Frankfurt am Main
Tel. +49 (0) 69 6300890
Fax. +49 (0) 69 63008989
redaktion@javamagazin.de
www.javamagazin.de

Chefredakteur: Sebastian Meyen
Redaktion: Claudia Schaumlöffel
Chefin vom Dienst: Nicole Bechtel
Schlussredaktion: Nicole Bechtel, Katharina Klassen, Frauke Pesch
Leitung Grafik & Produktion: Jens Mainz
Layout, Titel: Kristin Brockmann, Jessica Demirkaya, Melanie Hahn, David Heinze, Dominique Kalbassi, Jens Mainz, Michel Michiels-Corsten, Katharina Ochsenhirt, Maria Rudi
CD/DVD-Erstellung: Daniel Zuzek

Autoren dieser Ausgabe:
Manuel Aldana, Jens Bendisposto, Dr. Axel Benz, Adam Bien, Jens Bühring, Thomas Cotic, Oliver Dohmen, Sven Effttinge, Thomas Eichberger, Michael Frandsen, Thilo Fotscher, Kay Glahn, Michael Jastram, Wolfgang Martin, Harald Negrin, Carsten Pflug, Wolfgang Pleus, Florian Potschka, Matthias Rohr, Stefan Rook, Chris Rupp, Stefan Saring, Eric Schmieders, Daniel Seiler, Heiko Sippel, Axel Terfloth, Matthias Weißendorf, Henning Wolf, Eberhard Wolff, Lars Wunderlich

Anzeigenverkauf:
Software & Support Verlag GmbH
Patrik Baumann
Tel. +49 (0) 69 6300890
Fax. +49 (0) 69 63008989
pbaumann@javamagazin.de

Es gilt die Anzeigenpreisliste Nummer 11

Pressevertrieb:
DPV Network
Tel.+49 (0) 40 378456261,
www.dpv-network.de

Druck: PVA Landau
ISSN: 1619-795X

Abo-Service:
Software & Support Verlag GmbH
Tel. +49 (0) 69 6300890
Fax +49 (0) 69 63008989
www.javamagazin.de/service/

Abonnementpreise der Zeitschrift:		
Inland:	12 Ausgaben	€ 79,-
Europ. Ausland:	12 Ausgaben	€ 89,-
Studentenpreis (Inland):	12 Ausgaben	€ 69,-
Studentenpreis (Ausland):	12 Ausgaben	€ 79,-

Einzelverkaufspreis:	
Deutschland:	€ 7,50
Österreich:	€ 8,60
Schweiz:	sFr 15,80

Erscheinungsweise: monatlich

© Software & Support Verlag GmbH

Alle Rechte, auch für Übersetzungen, sind vorbehalten. Reproduktionen jeglicher Art (Fotokopie, Nachdruck, Mikrofilm oder Erfassung auf elektronischen Datenträgern) nur mit schriftlicher Genehmigung des Verlages. Jegliche Software auf der Begleit-CD zum Java Magazin unterliegt den Bestimmungen des jeweiligen Herstellers. Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Honorierte Artikel gehen in das Verfügungsrecht des Verlags über. Mit der Übergabe der Manuskripte und Abbildungen an den Verlag erteilt der Verfasser dem Herausgeber das Exklusivitätsrecht zur Veröffentlichung. Für unverlangt eingeschickte Manuskripte, Fotos und Abbildungen keine Gewähr. Java™ ist ein eingetragenes Warenzeichen der Sun Microsystems Inc.

