

Feature Driven Development (FDD)

Dieses Dokument ist die Übersetzung der Original-FDD-Definition von Jeff De Luca (<http://www.nebulon.com>), <http://www.nebulon.com/articles/fdd/latestprocesses.html>

Übersetzt hat Stefan Roock (stefan.roock@it-agile.de, <http://www.it-agile.de>).
Letzte Aktualisierung dieser Übersetzung: 14.11.2008

FDD-Prozess #1: Entwickle ein Gesamtmodell

Eine initiale projektweite Aktivität mit Fachexperten und Entwicklern unter der Leitung eines erfahrenen Objektmodellierers in der Rolle des Chefarchitekten.

Systemumfang und –kontext werden auf grober Ebene dargestellt. Es folgen detaillierte anwendungsfachliche Darstellungen für jeden Bereich, der modelliert werden soll. Nach jeder Darstellung werden kleine Teams aus Fachexperten und Entwicklern gebildet, die ihr eigenes Modell zur Unterstützung des vorgestellten Bereiches erstellen. Jedes Team stellt seine Modelle zum Review und zur Diskussion in der Gruppe vor. Eines der vorgestellten Modelle, oder eine Mischung aus mehreren Modellen, wird per Konsens ausgewählt und zum gemeinsamen Modell für das Fachgebiet. Das Modell des Fachgebietes wird in das Gesamtmodell integriert. Das Objektmodell wird dann iterativ aktualisiert mit den Inhalten des Prozesses #4 (*Entwirf je Feature*).

Eingangskriterien

- Fachexperten, Chefprogrammierer und der Chefarchitekt wurden ausgewählt.

Aufgaben

Stelle das Modellierungsteam zusammen	Projektmanager	Notwendig
--	-----------------------	------------------

Das Modellierungsteam besteht aus permanenten Mitgliedern aus dem Fachbereich und der Entwicklung, speziell Fachexperten und Chefprogrammierer. Weitere Projektmitglieder werden durch die Modellierungworkshops rotiert, so dass jeder die Chance erhält, zu partizipieren und den Fortschritt zu erleben.

Stelle Anwendungsbereich dar	Modellierungsteam	Notwendig
-------------------------------------	--------------------------	------------------

Ein Fachexperte gibt einen Überblick über das Fachgebiet, das modelliert werden soll. Der Überblick sollte auch Informationen enthalten, die in Beziehung zum Fachgebiet stehen, aber nicht notwendigerweise Teil der implementierten Funktionalität werden.

Studiere Dokumente	Modellierungsteam	Optional
---------------------------	--------------------------	-----------------

Das Team studiert verfügbare Referenz- oder Anforderungsdokumente wie Objektmodelle, funktionale Anforderungen (traditionell oder als Use-Case), Datenmodelle und Richtlinien.

Entwickle das Modell	Modellierungsteam in Kleingruppen	Notwendig
-----------------------------	--	------------------

Kleingruppen von maximal drei Personen erstellen jeweils ein Modell des Fachgebietes. Der Chefarchitekt wird möglicherweise ein erstes Modell skizzieren, um die Modellierung anzuregen. Ein Mitglied jeder Kleingruppe präsentiert das Modell der Kleingruppe im Plenum. Der Chefarchitekt schlägt ggf. weitere Modellierungsalternativen vor. Das Modellierungsteam wählt ein vorgeschlagenes Modell aus oder erstellt auf Basis der verschiedenen vorgeschlagenen Modelle ein neues.

Verfeinere das Gesamt-Objektmodell	Chefarchitekt, Modellierungsteam	Notwendig
---	---	------------------

Das Gesamt-Objektmodell wird gelegentlich aktualisiert mit den neuen Modellformen, die aus den Iterationen der oben genannten *Entwickle das Modell*-Aufgabe entstanden sind.

Schreibe Modellnotizen	Chefarchitekt, Chefprogrammierer	Notwendig
-------------------------------	---	------------------

Notizen zu detaillierten oder komplexen Modellformen und zu signifikanten Modellierungsalternativen werden erstellt, damit man später im Projekt darauf Bezug nehmen kann.

Überprüfung

Interne und externe Bewertung	Modellierungsteam, Fachbereich	Notwendig
--------------------------------------	---	------------------

Interne Bewertung oder Selbstkontrolle wird durch die aktive Teilnahme der Fachexperten erreicht. Bei Bedarf findet externe Bewertung statt, indem der Fachbereich (Anwender) kontaktiert wird, um Bestätigung oder Klarstellung zu bestimmten Angelegenheiten zu bekommen, die Auswirkungen auf das Modell haben.

Ausgangskriterien

Das Ergebnis dieses Prozesses ist das Objektmodell.

- Klassendiagramme mit Fokus auf der Modellform, also: Welche Klassen gibt es im Fachgebiet und wie sind diese miteinander.
- Identifizierte Methoden und Attribute finden sich in den Klassen.
- Sequenzdiagramme (falls vorhanden).
- Modellnotizen, die festhalten, warum eine bestimmte Modellform gewählt wurde und/oder welche Alternativen betrachtet wurden.

FDD-Prozess #2: Erstelle eine Featureliste

Eine initiale projektweite Aktivität, um alle Features zu identifizieren, die zu den Anforderungen gehören.

Es wird ein Team gebildet, das normalerweise aus genau den Chefprogrammierern aus Prozess #1 besteht. Das Team zerlegt den Anwendungsbereich in Fachgebiete, in darin enthaltene Geschäftsaktivitäten und je Geschäftsaktivität in einzelne Schritte (Features). So entsteht eine kategorisierte Featureliste. Die Kategorisierung der obersten Ebene stammt von der Partitionierung des Anwendungsbereiches, welche die Fachexperten in Prozess #1 vorgenommen haben.

Eingangskriterien

- Fachexperten, Chefprogrammierer und der Chefarchitekt wurden ausgewählt.

Aufgaben

Stelle das Featurelisten-Team zusammen	Projektmanager, Entwicklungsmanager	Notwendig
---	--	------------------

Das Team besteht aus den Chefprogrammierern des Modellierungsteams in Prozess #1.

Erstelle Featureliste	Featurelisten-Team	Notwendig
------------------------------	---------------------------	------------------

Das Team soll die Features auf Basis des Wissens identifizieren, dass es in Prozess #1 gewonnen hat. Es findet eine einfache funktionale Dekomposition in Fachgebiete statt, die von der Partitionierung des Anwendungsbereiches stammt, welche die Fachexperten in Prozess #1 für die Darstellung des Anwendungsbereiches verwendet haben. Der Anwendungsbereich wird in Fachgebiete unterteilt, die aus Geschäftstätigkeiten bestehen, die aus Geschäftstätigkeitsschritten (Features) bestehen. Features sind Systemfunktionen, die in der Sprache des Kunden in folgendem Benennungsschema beschrieben werden:

<Aktion> <Ergebnis> <Objekt>

Beispiele: „Berechne Summe eines Verkaufs“
„Berechne Gesamtzahl Verkäufe eines Artikels für ein Ladengeschäft“

Features sind so klein, dass sie in maximal zwei Wochen umgesetzt werden können, aber nicht so klein, dass sie auf der Ebene von Getter- und Setter-Methoden angesiedelt sind. Zwei Wochen ist das obere Limit; die meisten Features benötigen weniger Zeit. Wenn ein Geschäftsaktivitätsschritt größer erscheint, wird er in kleinere Schritte zerteilt, welche dann die Features darstellen.

Überprüfung

Interne und externe Bewertung	Featurelisten-Team, Fachbereich	Notwendig
--------------------------------------	--	------------------

Interne Bewertung bzw. Selbstkontrolle wird durch die aktive Teilnahme der Mitglieder des Modellierungsteams erreicht. Bei Bedarf wird Rücksprache mit den Fachexperten aus dem Modellierungsteam oder dem Fachbereich (Anwender) gehalten, um Bestätigung oder Klärung von speziellen Angelegenheiten zu erreichen, die die Featureliste beeinflussen.

Ausgangskriterien

Das Ergebnis dieses Prozesses ist die Featureliste.

- Eine Liste von Fachgebieten.
- Für jedes Fachgebiet eine Liste der Geschäftsaktivitäten des Fachgebiets.
- Für jeden Schritt der Geschäftsaktivität ein Feature, das diesen Schritt umsetzt.

FDD-Prozess #3: Plane je Feature

Eine initiale projektweite Aktivität, um den Entwicklungsplan zu erstellen.

Der Projektmanager, der Entwicklungsmanager und die Chefprogrammierer planen die Reihenfolge, in der die Features implementiert werden sollen, basierend auf Feature-Abhängigkeiten, der Arbeitsbelastung im Entwicklungsteam und der Komplexität der Features. Die Hauptaufgaben in diesem Prozess verlaufen nicht strikt-sequenziell. Wie viele Planungsaktivitäten werden sie gemeinsam betrachtet, so dass Verfeinerungen aus einer oder mehreren Aufgaben entstehen und dann die anderen Aufgaben erneut betrachtet werden. In einem typischen Szenario betrachtet man zunächst die Entwicklungsreihenfolge und dann die Zuordnung der Geschäftsaktivitäten zu Chefprogrammierern. In diesem Rahmen überlegt man nur, welche der Schlüsselklassen welchen Entwicklern zugewiesen werden (Erinnerung: Chefprogrammierer sind auch Entwickler). Wenn eine Balance erreicht ist und die Entwicklungsreihenfolge und die Zuordnung der Geschäftsaktivitäten zu Chefprogrammierern im Wesentlichen vollständig ist, wird die Zuordnung des Klassenbesitzes vervollständigt (über die Schlüsselklassen hinaus, die ja bereits ihre Besitzer haben).

Eingangskriterien

- Der Prozess #2 (*Erstelle eine Featureliste*) wurde abgeschlossen.

Aufgaben

Stelle das Planungsteam zusammen	Projektmanager	Notwendig
---	-----------------------	------------------

Das Planungsteam besteht aus dem Entwicklungsmanager und den Chefprogrammierern.

Lege die Entwicklungsreihenfolge fest	Planungsteam	Notwendig
--	---------------------	------------------

Das Planungsteam weist jeder Geschäftsaktivität einen Fertigstellungstermin (nur Monat und Jahr) zu. Die Identifikation der Geschäftsaktivitäten und Fertigstellungstermine (und damit die Entwicklungsreihenfolge) basiert auf:

- Die Abhängigkeiten zwischen Features im Sinne der Abhängigkeiten zwischen den beteiligten Klassen.
- Die Arbeitsbelastung der Klassenbesitzer soll gleichmäßig sein.
- Die Komplexität der zu implementierenden Features.
- Risikobehaftete oder komplexe Geschäftstätigkeiten sollen früh realisiert werden.
- Externe (sichtbare) Meilensteige wie Beta-Versionen, Vorabversionen, Prüfpunkte und „komplette Produkte“, die diese Meilensteine erfüllen.

Weise Geschäftsaktivitäten an Chefprogrammierer zu	Planungsteam	Notwendig
---	---------------------	------------------

Das Planungsteam legt Chefprogrammierer als Besitzer der Geschäftsaktivitäten fest. Die Festlegung berücksichtigt:

- Die Entwicklungsreihenfolge.
- Die Abhängigkeiten zwischen Features im Sinne der Abhängigkeiten zwischen den beteiligten Klassen.
- Die Arbeitsbelastung der Klassenbesitzer soll gleichmäßig sein (Erinnerung: die Chefprogrammierer sind auch Klassenbesitzer).
- Die Komplexität der zu implementierenden Features.

Weise Klassen an Entwickler zu	Planungsteam	Notwendig
---------------------------------------	---------------------	------------------

Das Planungsteam weist Entwickler als Klassenbesitzer zu. Entwickler besitzen jeweils mehrere Klassen. Die Zuweisung von Klassen zu Entwicklern berücksichtigt:

- Die Arbeitsbelastung der Klassenbesitzer soll gleichmäßig sein (Erinnerung: die Chefprogrammierer sind auch Klassenbesitzer).
- Die Komplexität der zu implementierenden Features.
- Die Benutzung (z.B. „hochfrequent“) der Klassen.
- Die Entwicklungsreihenfolge.

Überprüfung

Selbstkontrolle	Planungsteam	Notwendig
------------------------	---------------------	------------------

Weil die Planung eine Teamaktivität ist, findet eine Selbstkontrolle mit aktiver Teilnahme von Chefentwicklern, Entwicklungsmanager und Projektmanager statt.

Ausgangskriterien

Das Ergebnis dieses Prozesses ist der Entwicklungsplan, bestehend aus

- Geschäftsaktivitäten mit Fertigstellungsdaten (Monat und Jahr).
- Verantwortlicher Chefprogrammierer je Geschäftsaktivität.
- Fachgebiete mit Fertigstellungsdaten (Monat und Jahr), abgeleitet vom letzten Fertigstellungsdatum der zugehörigen Geschäftsaktivitäten.
- Die Klassenbesitz-Liste, in der jeder Klasse ein Entwickler als Besitzer zugeordnet ist.

FDD-Prozess #4: Entwurf je Feature

Eine Aktivität je Feature, um das Feature-Entwurfspaket zu erstellen.

Eine Menge von Features wird für die Entwicklung eingeplant, indem sie einem Chefprogrammierer zugewiesen werden. Der Chefprogrammierer wählt Features für Entwicklung aus seiner Feature-„Eingangsbox“. Er wählt mitunter mehrere Features, die alle dieselben Klassen (und damit Entwickler) benötigen. In der Praxis wählt der Chefprogrammierer häufig Mengen von Features aus, die das gleiche Fertigstellungsdatum haben. Das Ergebnis nennt man *Chefprogrammierer-Arbeitspaket*.

Dann stellt der Chefprogrammierer ein Featureteam zusammen, indem er die Besitzer der Klassen identifiziert, die voraussichtlich in die Entwicklung der oder des ausgewählten Features involviert sind. Dieses Team erstellt dann die Sequenzdiagramme für die zugewiesenen Features. Der Chefprogrammierer verfeinert danach das Objektmodell auf Basis der Sequenzdiagramme. Daraufhin schreiben die Entwickler Klassen- und Methodenrumpfe. Eine Entwurfsinspektion findet statt.

Eingangskriterien

- Der Planungsprozess wurde abgeschlossen.

Aufgaben

Stelle Feature-Team zusammen	Chefprogrammierer	Notwendig
-------------------------------------	--------------------------	------------------

Der Chefprogrammierer identifiziert die Klassen, die voraussichtlich relevant für den Entwurf der aktuellen Features sind und aktualisiert die Feature-Informationen entsprechend. Mit Hilfe der Klassenbesitz-Liste identifiziert der Chefprogrammierer die Entwickler, die für das Featureteam notwendig sind. Als Teil dieser Aufgabe erstellt der Chefprogrammierer ein neues Entwurfspaket für die Features als Teil des Arbeitspaketes.

Stelle Anwendungsbereich dar	Fachexperte	Optional
-------------------------------------	--------------------	-----------------

Der Fachexperte gibt für das zu entwerfende Feature einen Überblick über den Anwendungsbereich. Dieser sollte auch fachliche Informationen enthalten, die mit dem Feature zusammenhängen, aber nicht notwendigerweise Teil der Implementation werden.

Diese Aufgabe ist optional, abhängig von der Komplexität des Features und/oder seiner Interaktionen.

Studiere referenzierte Dokumente	Feature-Team	Optional
---	---------------------	-----------------

Das Feature-Team studiert die referenzierten Dokumente für das zu entwerfende Feature sowie alle Memos, Oberflächendesigns, Schnittstellenspezifikationen externer Systeme und jegliche sonstige unterstützende Dokumentation.

Dies ist eine optionale Aufgabe, abhängig von der Komplexität des Features und/oder seiner Interaktionen.

Entwickle Sequenz-Diagramm(e)	Feature-Team	Notwendig
--------------------------------------	---------------------	------------------

Entwickle die Sequenzdiagramme, die für das zu entwerfende Feature notwendig sind. Die Diagramm-Dateien sollten ins Versionskontrollsystem eingecheckt werden. Alle Alternativentwürfe, Entwurfsentscheidungen, Anforderungsdetails und Notizen werden ebenfalls erfasst und im Abschnitt *Entwurfsalternativen* des Entwurfspaketes dokumentiert.

Verfeinere das Objektmodell	Chefprogrammierer	Notwendig
------------------------------------	--------------------------	------------------

Der Chefprogrammierer erstellt einen Feature-Team-Bereich für die Features. Dieser Bereich ist entweder ein Verzeichnis auf einem Dateiserver oder einem PC, das vom Chefprogrammierer bei Bedarf gesichert wird. Alternativ kann man die Unterstützung des Versionskontrollsystems in Anspruch nehmen. Der Feature-Team-Bereich hat die Aufgabe, den jeweils aktuellen Arbeitszustand des Features im Feature-Team sichtbar zu machen und vor dem Rest des Projektes zu verbergen. Der Chefprogrammierer verfeinert das Modell, indem er auf Basis der Sequenzdiagramme Klassen, Methoden und Attribute hinzufügt oder ändert. Dies führt dazu, dass die Quelltexte im Feature-Team-Bereich aktualisiert werden.

Der Chefprogrammierer erstellt Modelldiagramme in einem Format, das für die Veröffentlichung im Projekt geeignet ist. Diese Dateien sollten in das Versionskontrollsystem eingecheckt und im Projekt-Intranet veröffentlicht werden.

Die Klassenbesitzer schreiben die Klassen- und Methodenrumpfe für die Sequenzdiagramme des Features auf Basis der aktualisierten Quelltexte im Feature-Team-Bereich aus der „Verfeinere das Objektmodell“-Aufgabe. Dazu gehören Parameter- und Rückgabetypen, Ausnahmen und Nachrichten. Sobald ein Entwickler diese Aufgabe erledigt hat, generiert der Chefprogrammierer die API-Dokumentation mit *<Ihr Werkzeug>* und veröffentlicht es im Projekt-Intranet.

Überprüfung

Entwurfsinspektion	Feature-Team	Notwendig
---------------------------	---------------------	------------------

Eine Entwurfsinspektion mit den Mitgliedern des Feature-Teams oder mit anderen Projektmitgliedern findet statt. Der Chefprogrammierer entscheidet, ob die Entwurfsinspektion innerhalb des Feature-Teams stattfindet oder zusammen mit anderen Projektmitgliedern.

Bei Akzeptanz des Entwurfes wird eine ToDo-Liste je betroffener Klasse erstellt und jedes Teammitglied plant Termine für seine Aufgaben.

Der Chefprogrammierer muss außerdem die Änderungen des Feature-Team-Bereiches ins Versionskontrollsystem integrieren.

Ausgangskriterien

Das Ergebnis dieses Prozesses ist ein erfolgreich inspiziertes Entwurfspaket. Das Entwurfspaket besteht aus:

- Memo, das das Entwurfspaket integriert beschreibt, so dass es eigenständig einem Review unterzogen werden kann.
- Die referenzierten Anforderungen (falls vorhanden) in der Form von Dokumenten sowie alle verwandten Memos und unterstützende Dokumentationen.
- Die Sequenzdiagramme.
- Entwurfsalternativen (falls vorhanden).
- Das Objektmodell mit neuen/aktualisierten Klassen, Methoden und Attributen.
- Die von *<Ihr Werkzeug>* generierten Ausgaben für Klassen- und Methodenrumpfe, die durch diesen Entwurfsprozess erzeugt oder verändert wurden.
- Termine/To-Do-Punkte für Arbeiten an betroffenen Klassen für jedes Teammitglied.

FDD-Prozess #5: Konstruiere je Feature

Eine Aktivität je Feature, um eine komplette kundennützliche Funktion (Feature) zu erstellen. Beginnend mit dem Entwurfspaket implementieren die Klassenbesitzer die Anteile ihrer Klassen, die für das Feature notwendig sind. Für den entwickelten Code werden dann Unittests geschrieben und ausgeführt und der Code wird inspiziert – die Reihenfolge dieser Schritte wird vom Chefprogrammierer festgelegt. Nach erfolgreicher Code-Inspektion wird der Code für den Build freigegeben.

Eingangskriterien

- Der Prozess #4 (*Entwurf je Feature*) wurde abgeschlossen. Das bedeutet, dass das Entwurfspaket erfolgreich inspiziert wurde.

Aufgaben

Implementiere Klassen und Methoden	Feature-Team	Notwendig
---	---------------------	------------------

Die Klassenbesitzer implementieren die Anteile, die notwendig sind, damit ihre Klassen die Anforderungen des Features erfüllen.

Code-Inspektion	Feature-Team	Notwendig
------------------------	---------------------	------------------

Es findet eine Code-Inspektion mit den Feature-Team-Mitgliedern oder anderen Projektmitgliedern vor oder nach der Unittest-Aufgabe statt. Der Chefprogrammierer entscheidet, ob die Code-Inspektion innerhalb des Feature-Teams oder mit anderen Projektmitgliedern stattfindet. Die Entscheidung, ob die Code-Inspektion vor oder nach dem Unittest stattfindet, fällt der Chefprogrammierer.

Unittest	Feature-Team	Notwendig
-----------------	---------------------	------------------

Die Klassenbesitzer testen ihren Code, um sicherzustellen, dass ihre Klassen alle Anforderungen erfüllen. Der Chefprogrammierer legt fest, ob und wo übergreifende Tests notwendig sind, also Tests, die über einzelne Klassen des Features hinausgehen.

Gebe für den Build frei	Chefprogrammierer, Feature-Team	Notwendig
--------------------------------	--	------------------

Klassen können erst für den Build freigegeben werden, nachdem eine Code-Inspektion erfolgreich durchgeführt wurde. Der Chefprogrammierer hält nach, welche Klassen freigegeben wurden und ist der Integrationspunkt für das komplette Feature.

Überprüfung

Code-Inspektion und Unittest	Chefprogrammierer, Feature-Team	Notwendig
-------------------------------------	--	------------------

Eine erfolgreiche Code-Inspektion plus die erfolgreiche Ausführung der Unittests prüft das Ergebnis dieses Prozesses. Code-Inspektion und die Unittest-Aufgabe sind oben beschrieben.

Ausgangskriterien

Die Ergebnisse dieses Prozesses sind:

- Klassen und/oder Methoden, bei denen die Code-Inspektion erfolgreich war.
- Klassen, die für den Build freigegeben wurden.
- Die Fertigstellung einer für den Kunden nützlichen Funktion (Feature).