

Themenccluster: Agile und schlanke Methoden

Thema: Feature Driven Development

Dr. Walter Rafeiner-Magor

09.09.2013

Feature Driven Development (FDD)

● Definition:

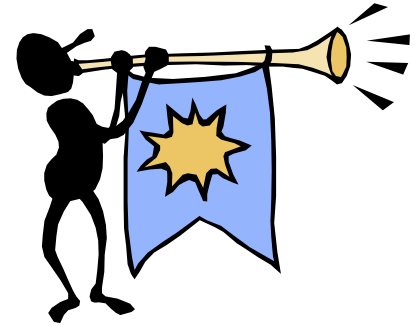
- FDD stellt den Begriff des „Features“ in den Mittelpunkt. Jedes Feature stellt einen Mehrwert für den Kunden dar.
- FDD ist eine schlanke und agile Methode der SE.

● Ausgangspunkt:

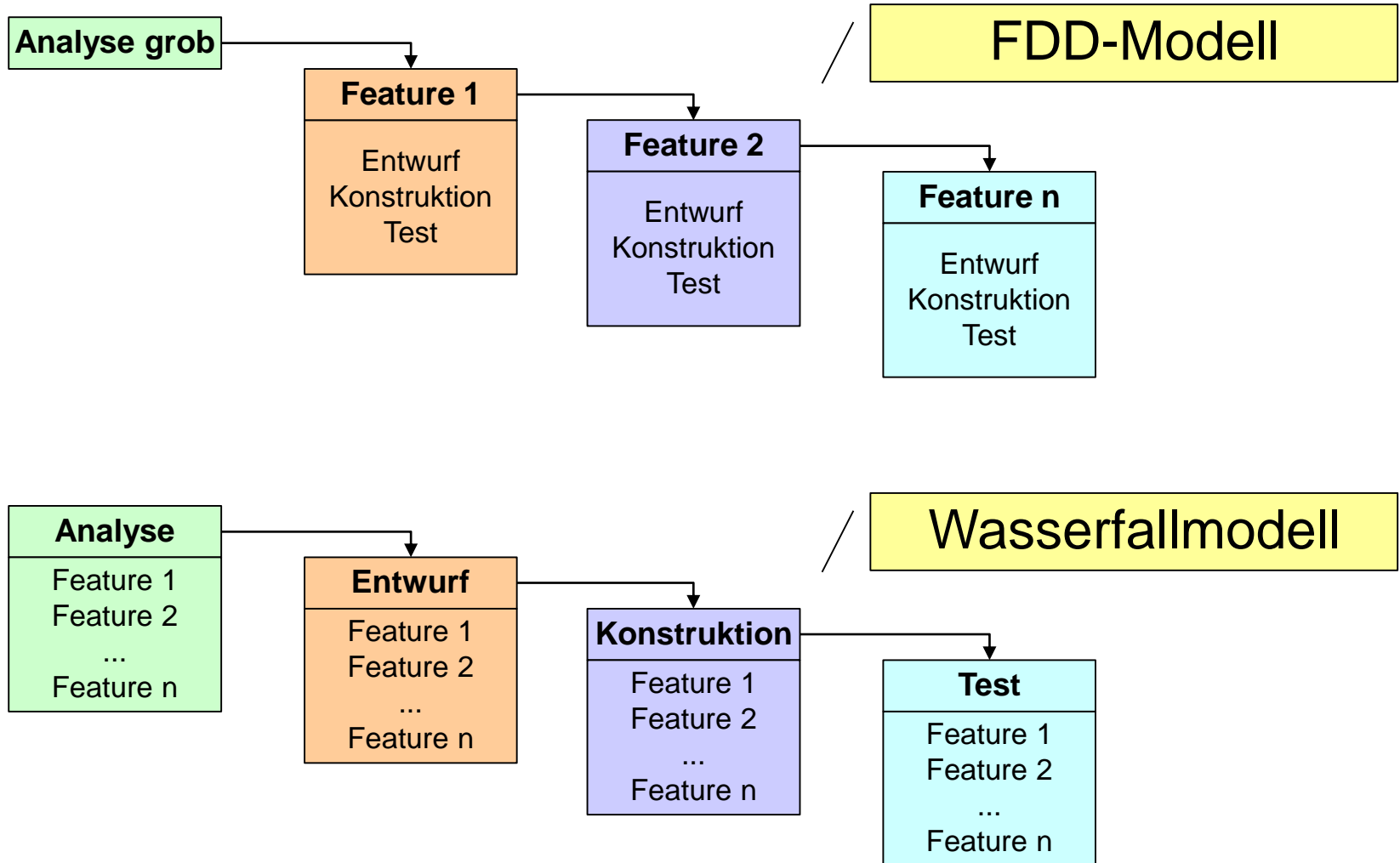
- Jeff de Luca, 1997
- Bankenprojekt: 50 Entwickler, 15 Monate

● Einsatz:

- Für größere Teams geeignet
- Für Projekte mit OO-Techniken
- Teams, welche mit phasenorientierten Modellen vertraut sind, aber auch Interesse an iterativen Methoden haben.



FDD versus Wasserfallmodell



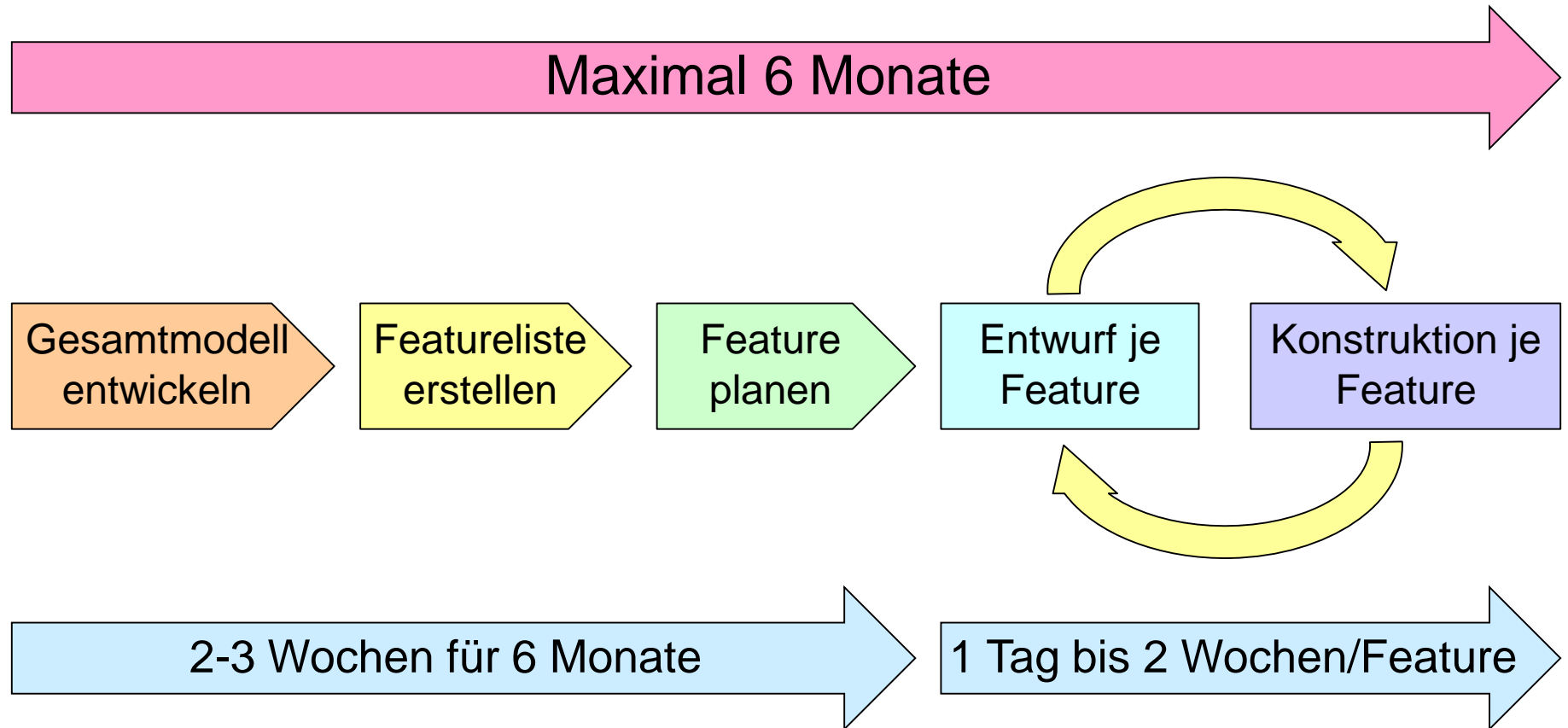
FDD: Rollenmodell

● Rollen:

- Projektleiter
- Entwicklungsleiter
- Chefarchitekt (Chefmodellierer)
- Chef-Programmierer
- Entwickler
- Fachexperten



FDD: Teilprozesse



FDD: Features

● Beispiele:

Features sammeln

- Add a student to a seminar waiting list.
- Calculate fee for a parking pass.
- Calculate the average mark on a transcript.
- Display the name and address of a student on a transcript.
- Drop a student from a seminar.
- Enroll a student in a seminar.
- List the prerequisites for a seminar.
- List the seminars of a student on a transcript.
- Track number of parking passes.

FDD: Featuresets

Features strukturieren

● **Transcript:**

- Calculate the average mark on a transcript.
- List the seminars of a student on a transcript.
- Display the name and address of a student on a transcript.

● **Enrollment:**

- List the prerequisites for a seminar.
- Enroll a student in a seminar.
- Drop a student from a seminar.
- Add at student to a seminar waiting list.

● **Parking Passes:**

- Calculate fee for a parking pass.
- Track number of parking passes.

FDD: Gesamtmodell entwickeln

- **Ziele:**

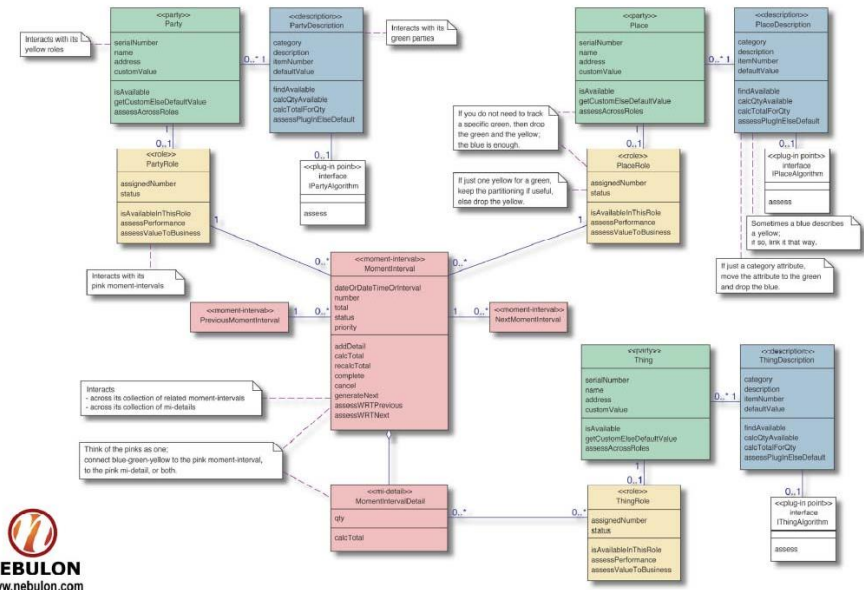
- Fachgebiet kennenlernen und verstehen
- Fachliches Klassenmodell
- Ggf. auch Sequenz- und Zustandsdiagramme

● Beteiligte:

- Chefarchitekt = Moderator
- (Chef-)Programmierer
- Fachexperten

● Umsetzung:

- Meist Modeling In Color



Exkurs: Modeling in Color

- **Grundlagen:**

- Fachliche Konzepte werden
- in 4 Archetypen klassifiziert.

- **Party Place Thing:**

- Jemand oder etwas, das unterschiedliche Rollen annehmen kann (Person, Organisation; Büro, Shop; Flugzeug, DVD,...)

- **Description:**

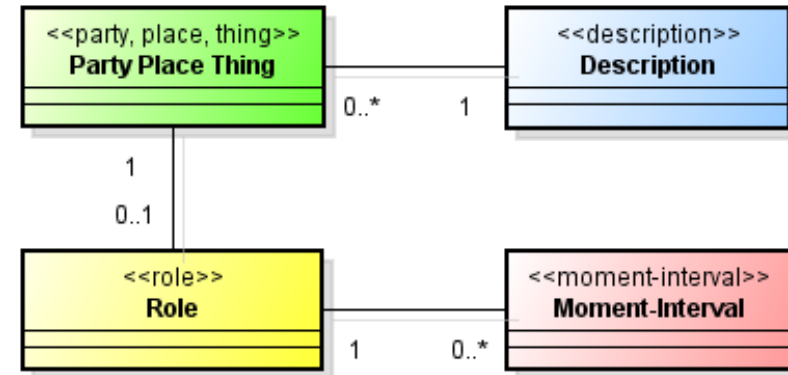
- Auflistung von Werten und deren Verhalten (alle Arten von Produktbeschreibungen; Der Unterschied zwischen blau und grün ist der Unterschied zwischen dem Film und der DVD auf welcher dieser Film gebrannt wurde)

- **Role:**

- Mitwirkung einer Person, Ort,... an einem Geschäftsfall

- **Moment-Interval:**

- Ist es ein Zeitpunkt oder Zeitraum im Hinblick auf einen Geschäftsfall (Order, Meeting, Buchung,...)



FDD: Featureliste erstellen

- **Ziele:**

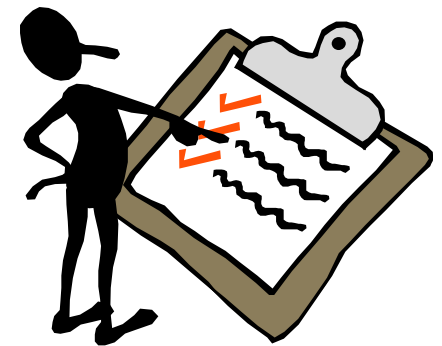
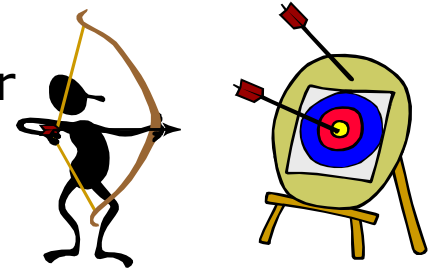
- Alle Anforderungen in Form von Features liegen vor
- Basis für weiteres Vorgehen und Schätzung

- **Beteiligte:**

- Chef-Programmierer
- Anschließend Abstimmung mit Fachexperten

- **Umsetzung:**

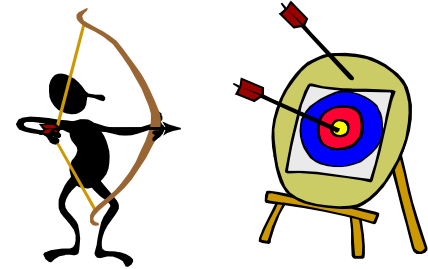
- Feature-Schema: **<Aktion> <Ergebnis> <Objekt>**
Beispiel: "Berechne Gesamtzahl Verkäufe eines Artikels für ein Ladengeschäft"
- Hierarchie der Features:
 - Major Feature Set (Geschäftsbereich)
 - Feature Set (Geschäftstätigkeit)
 - Feature (Systemfunktion)



FDD: Feature planen

- **Ziele:**

- Projektplan erstellt, Aufwände und Termine geklärt
- Ablauf für alle Entwickler (ausbalanciert) festlegt



- **Beteiligte:**

- Projektleiter
- Entwicklungsleiter
- Chef-Programmierer

- **Umsetzung:**

- Feature-Klassen-Beziehungen ermitteln
- Class-Owner festlegen
- Arbeitsbelastung ausbalancieren
- Es ergeben sich Feature-Teams:
 - Alle Class-Owner der beteiligten Klassen.



FDD: Entwurf je Feature

- **Ziele:**

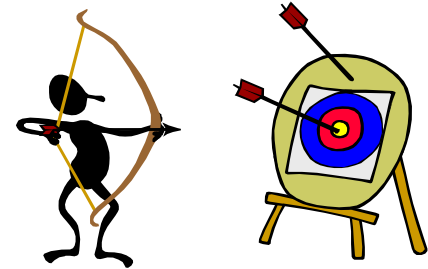
- Gemeinsamen Entwurf erstellen
- Aus gemeinsamen Entwürfen lernen

- **Beteiligte:**

- Feature-Team
- ggf. Chefarchitekt
- ggf. Chef-Programmierer

- **Umsetzung:**

- Klassen- und Sequenzdiagramme erstellen
- Design-Inspektionen
- Tests erstellen
- Arbeitspakete (Bearbeitung mehrerer Features parallel)
- Anschließend folgt sofort: Konstruktion je Feature



FDD: Konstruktion je Feature

● Ziele:

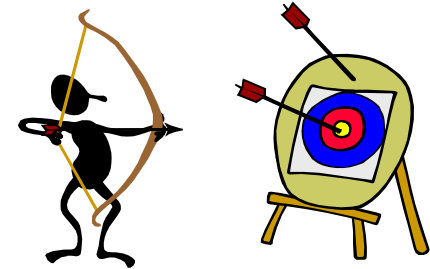
- Produktivklassen fertiggestellt
- Tests erfolgreich
- Programmieren verbessert

● Beteiligte:

- Feature-Team
- ggf. Chefarchitekt
- ggf. Chef-Programmierer

● Umsetzung:

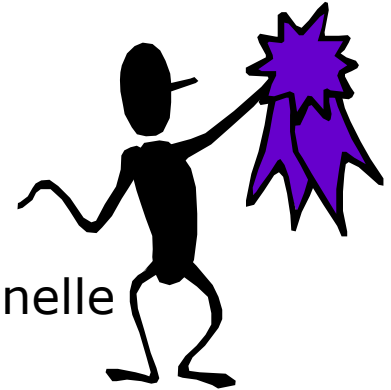
- Class-Owner programmieren Produktivklassen
- Class-Owner setzen Tests um
- Im Team erfolgen Code-Inspektionen
- Anschließend folgt sofort wieder: Entwurf je Feature



FDD: Zusammenfassung

● Vorteile:

- Für größere Teams geeignet
- Für Projekte mit OO-Techniken
- Auch für Organisationen geeignet, welche konventionelle Softwareentwicklung gewöhnt sind



● FDD ist agil?:

- Rückkopplung/Feedback: an vielen Stellen
- Einfachheit: ja

● Agiles Manifest:

- Menschen wichtiger als Prozesse: ja, aber anders
- Laufende Software wichtiger als Doku: ja
- Zusammenarbeit mit dem Kunden wichtiger als Vertragsverhandlungen: ja
- Veränderungen begrüßen statt Planverfolgung: ja, aber anderer Takt



Vielen Dank!