Feature Driven Development

Eric Nickell

Overview

- History
- What is Feature Driven Development?
- What is a Feature?
- Feature Driven Development Roles
  - Class Ownership
- Feature Driven Development Process
  - Mandated Code Inspections
- Reporting
- Summary
- References

History

- Original Creator: Jeff De Luca
  - Singapore in late 1997
- FDD evolved from an actual project
  - Bank Loan Automation
  - Luca was Project manager
  - 50 member developer team
  - Peter Coad : Chief Architect
    - 1990’s object-oriented analysis and design expert

What is Feature Driven Development?

- FDD is an agile software development process
- FDD uses a short-iteration model
- FDD combines key advantages of other popular agile approaches along with other industry-recognized best practices
- FDD was created to easily scale to much larger projects and teams

What is a Feature?

- Definition: small function expressed in client-valued terms
- FDD’s form of a customer requirement

What is a Feature?

- Feature naming template:

<action> the <result> <by|for|of|to> a(n) <object>

- Examples:
  - Calculate the total of a sale
  - Validate the password of a user
  - Authorize the sales transaction of a customer

What is a Feature?

- Features are to be “small” in the sense they will take no more than two weeks to complete
- Features that appear to take longer are to be broken up into a set of smaller features
- Note: Two weeks is the maximum, most features take far less time (1 - 5 days)

FDD Primary Roles

- Project Manager
- Chief Architect
- Development Manager
- Domain Experts
- Class Owners
- Chief Programmers

Class Ownership

- Class assigned to specific developer
- Class owner responsible for all changes in implementing new features

- Collective Ownership
  - Any developer can modify any artifact at any time
  - addresses problem of deadlock
- Class Ownership does not imply exclusivity but only responsibility

Class Ownership

- Advantages
  - Someone responsible for integrity of each class
  - Each class will have an expert available
  - Class owners can make changes much quicker
  - Easily lends to notion of code ownership
  - Assists in FDD scaling to larger teams

FDD Primary Roles

- Project Manager
- Chief Architect
- Development Manager
- Domain Experts
- Class Owners
- Chief Programmers

FDD Supporting Roles

- Domain Manager
- Release Manager
- Language Guru
- Build Engineer
- Toolsmith
- System Administrator
- Tester
- Deployer
- Technical Writer

Feature Driven Development Process

- Process #1: Develop an Overall Model
- Process #2: Build a Features List
- Process #3: Plan By Feature
- Process #4: Design By Feature
- Process #5: Build By Feature

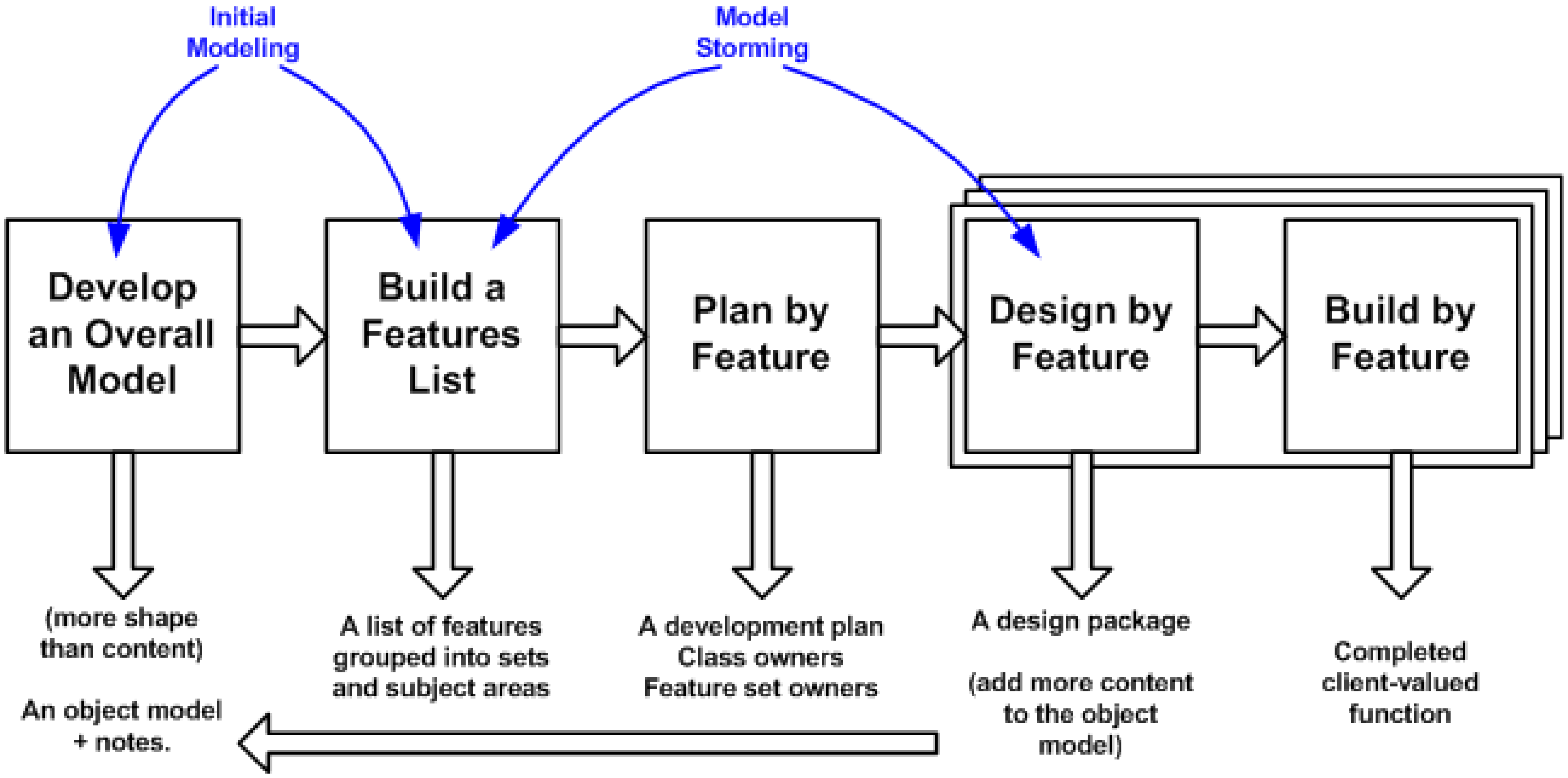
Feature Driven Development Process

- Project wide upfront design activities:
  - Process #1: Develop an Overall Model
  - Process #2: Build a Features List
  - Process #3: Plan By Feature
  - Goal: not to design the system in its entirety but instead is to do just enough initial design that you are able to build on

Feature Driven Development Process

- Deliver the system feature by feature:
  - Process #4: Design By Feature
  - Process #5: Build By Feature
  - Goal: Deliver real, completed, client-valued function as often as possible

Feature Driven Development Process



Process #1: Develop an Overall Model

- Form a modeling team
- Domain walk-through
- Build High-level object model
- Record Notes
- Goal - for team members to gain a good, shared understanding of the problem domain and build a foundation

Process #2: Build a Features List

- All Features are organized in a three level hierarchy :
    - Domain Subject Area
        - Business Activity
            - Features

Process #3: Plan By Feature

- Construct initial schedule
  - Formed on level of individual features
    - Prioritize by business value
    - Also consider dependencies, difficulty, and risks
- Assign responsibilities to team members
  - Determine Class Owners
  - Assign feature sets to chief programmers

Process #4: Design By Feature

- Form Feature Teams
- Team members collaborate on the full low level analysis and design
- Certain features may require teams to bring in domain experts
- Teams need to update the model artifact to support their changes

Feature Teams

- Chief Programmers pick teams based on the current feature in development
- Chief Programmers lead picked team
- Usually 3 to 5 people
- Upon completion of the current feature the team disbands
- Each team will concurrently work on their own independent iteration
- Possible to be on multiple teams at once

Process #4: Design By Feature

- Form Feature Teams
- Team members collaborate on the full low level analysis and design
- Certain features may require teams to bring in domain experts
- Teams need to update the model artifact to support their changes

Process #5: Build By Feature

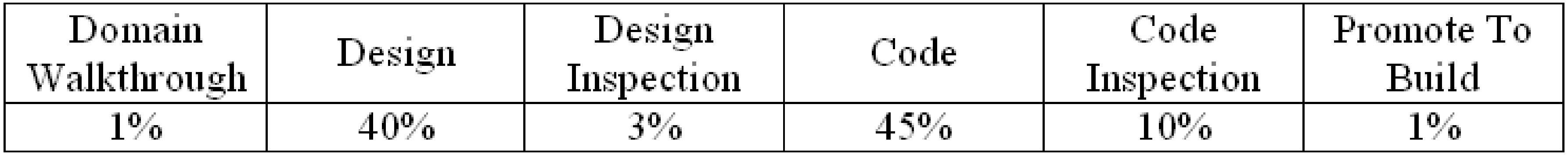
- Implement designed feature
- Test feature
  - Unit-level
  - Feature-level
- Mandated Code Inspections
- Integrate with regular build

Mandated Code Inspections

- Two Main Reasons
  - Research has shown that when done properly, inspections find more bugs as well as different types of bugs than any other form of testing
  - Great learning experience

Reporting

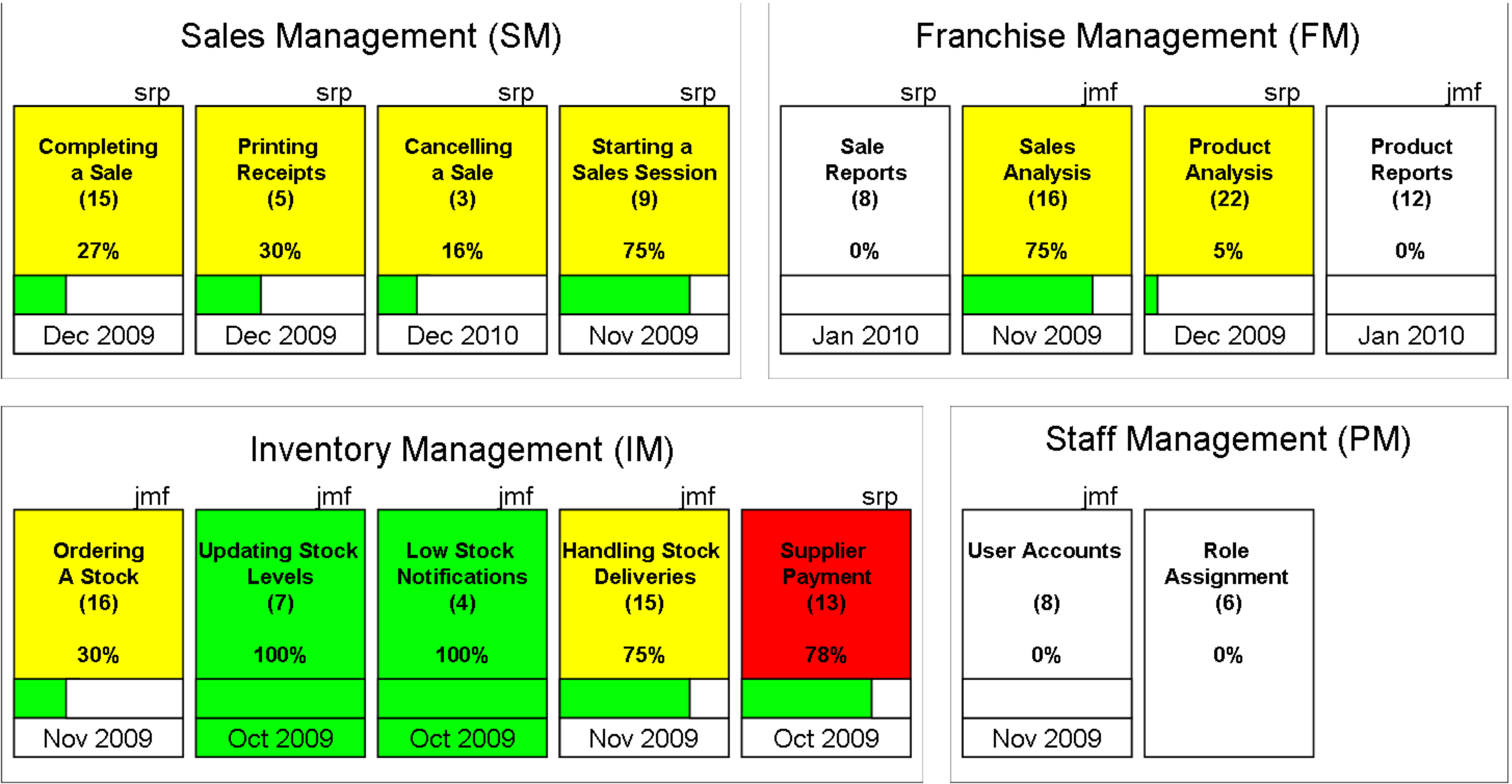
- FDD emphasizes the ability to provide accurate, meaningful, and timely progress information to all stakeholders within and outside the project
- Feature Milestones



Domain Walkthrough	Design	Design Inspection	Code	Code Inspection	Promote To Build
1%	40%	3%	45%	10%	1%

Reporting

- Parking Lot Chart



Sales Management (SM)

srp
Completing a Sale (15)
27%
Dec 2009

srp
Printing Receipts (5)
30%
Dec 2009

srp
Cancelling a Sale (3)
16%
Dec 2010

srp
Starting a Sales Session (9)
75%
Nov 2009

Franchise Management (FM)

srp
Sale Reports (8)
0%
Jan 2010

jmf
Sales Analysis (16)
75%
Nov 2009

srp
Product Analysis (22)
5%
Dec 2009

jmf
Product Reports (12)
0%
Jan 2010

Inventory Management (IM)

jmf
Ordering A Stock (16)
30%
Nov 2009

jmf
Updating Stock Levels (7)
100%
Oct 2009

jmf
Low Stock Notifications (4)
100%
Oct 2009

jmf
Handling Stock Deliveries (15)
75%
Nov 2009

srp
Supplier Payment (13)
78%
Oct 2009

Staff Management (PM)

jmf
User Accounts (8)
0%
Nov 2009

Role Assignment (6)
0%

Summary

- FDD combines many of the best practices of other agile models
- FDD was initially created for and is more geared towards large project teams
- FDD puts less focus on initial design and quickly gets to the point where the team can deliver new functionality to the project feature by feature

References

- [1] Palmer, Stephen. "FDD History." N.p., 2010. Web. 27 Mar 2010. <http://www.step-10.com/SoftwareProcess/FeatureDrivenDevelopment/FDDHistory.html>.
- [2] Ambler, Scott. "Feature Driven Development (FDD) and Agile Modeling." *Agile Modeling*. N.p., 2005-2009. Web. 27 Mar 2010. <http://www.agilemodeling.com/essays/fdd.htm>.
- [3] Palmer, Stephen. "An Introduction to Feature-Driven Development." *DZone*. N.p., 11/20/2009. Web. 27 Mar 2010. <http://agile.dzone.com/articles/introduction-feature-driven>.
- [4] Palmer, Stephen. "An Introduction to Feature-Driven Development – Part 2." *DZone*. N.p., 11/20/2009. Web. 27 Mar 2010. <http://java.dzone.com/articles/introduction-feature-driven-2>.
- [5] De Luca, Jeff. "The Latest FDD Processes." *Nebulon*. N.p., n.d. Web. 27 Mar 2010. <http://www.nebulon.com/articles/fdd/download/fddprocessesUSLetter.pdf>.
- [6] Palmer, Stephen. "FDD: People." N.p., 2010. Web. 27 Mar 2010. <http://www.step-10.com/SoftwareProcess/FeatureDrivenDevelopment/FDDPeople.html>.