COURSE NAME

SOFTWARE ENGINEERING

CSC 3114

(UNDERGRADUATE)

# CHAPTER 7

# FEATURE DRIVEN DEVELOPMENT (FDD)

# HISTORY OF FDD

- ❑ 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

# WHAT IS FDD?

- Feature Driven Development (FDD)
- 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?

- FDD delivers the system feature by feature
- Feature is a small function expressed in client-valued terms which presents the customer requirements to be developed in software using small iteration
- 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. Two weeks is the maximum, most features take less time (1 - 5 days)
- 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

# CLASS OWNERSHIP

- ❑ Class (feature) 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
- ❑ Advantages of Class Ownership are:
  - 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 (XP)

# FDD ROLES

□ FDD Primary Roles

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

□ FDD Supporting Roles

| | |
|---|---|
| Language Guru (shared vocabulary) | |
| Toolsmith (making tools for application) | |
| Tester | |
| Technical Writer (documentation) | |

# FDD 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

# FDD PROCESS

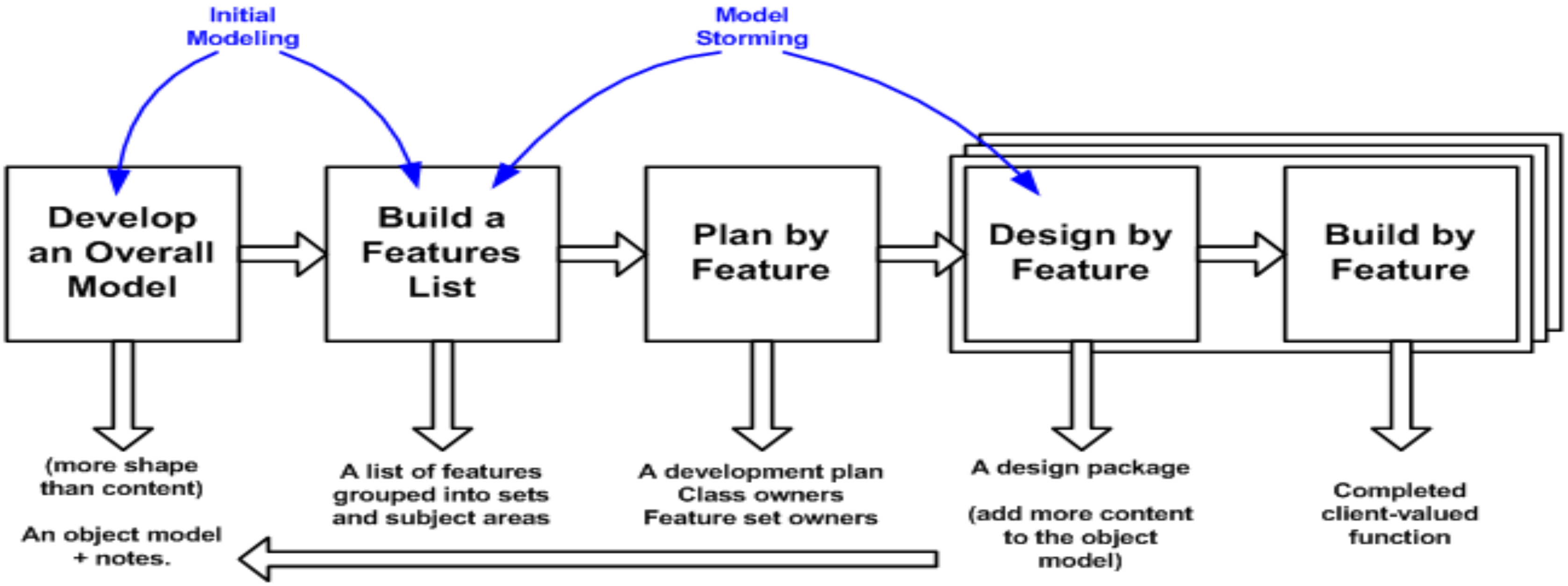
❑ Project wide upfront design activities:

- 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

❑ 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

# FDD PROCESS


Initial Modeling
Model Storming
Develop an Overall Model
Build a Features List
Plan by Feature
Design by Feature
Build by Feature
(more shape than content)
An object model + notes.
A list of features grouped into sets and subject areas
A development plan Class owners Feature set owners
A design package
(add more content to the object model)
Completed client-valued function

Copyright 2002-2005 Scott W. Ambler
Original Copyright S. R. Palmer & J.M. Felsing

# FDD 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

# FDD PROCESS

❑ **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

# FDD PROCESS

❑ **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 Team**

- 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

# FDD PROCESS

❑ **Process #5: Build by Feature**

- Implement designed feature
- Test feature
  - Unit-level
  - Feature-level
- Mandated Code Inspections (formal review with checklist)
- Integrate with regular build

# FDD PROCESS

- **Mandated Code Inspections** for Two Main Reasons
  - Research has shown that when it is done properly, inspections find more bugs as well as different types of bugs than any other form of testing.
  - It is also a 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

# REFERENCES

- R.S. Pressman & Associates, Inc. (2010). *Software Engineering: A Practitioner's Approach.*
- Kelly, J. C., Sherif, J. S., & Hops, J. (1992). An analysis of defect densities found during software inspections. *Journal of Systems and Software*, *17*(2), 111-117.
- Bhandari, I., Halliday, M. J., Chaar, J., Chillarege, R., Jones, K., Atkinson, J. S., & Yonezawa, M. (1994). In-process improvement through defect data interpretation. *IBM Systems Journal*, *33*(1), 182-214.