

An Overview of
Feature Driven
Development
Presented By
Robert W. Starinsky
At DePaul University
02/26/2001

Presentation Agenda

The State of Development Today
What is *Feature Driven* Development
Why use *Feature Driven* Development
How does *Feature Driven* Development work
How *Feature Driven* Development tracks progress
Who is behind the *Feature Driven* Development movement

The State of Development Today

According to the Standish Group:

\$250 billion is spent annually for application development;
\$140 billion is lost due to canceled projects, late projects and projects that *NEVER* end;
\$110 billion is the value received -- less than half the total investment.

The State of Development Today

Why such poor results?

Poor or incomplete planning;
Lack of understanding of technical or business issues;
Failure to place customer or end user needs *FIRST*.

The State of Development Today

The Challenge:

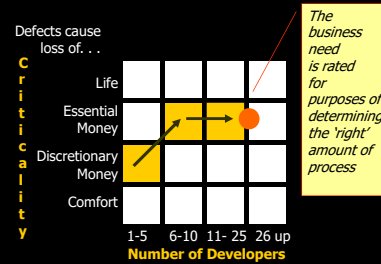
Traditionally, development methodologies have been process-rich and results-poor.

The Solution:

Leaner, scalable methodologies that are results-driven.

Just In Time and Just Right

*Methodology today is being optimized by
Criticality and Project Size:*



Development Methodologies

Leading experts in the field have concluded through field research that:

One size of methodology does not fit all.

Cookie cutter methodologies aren't effective.

Time is fleeting and change is the constant.

Results are what matters most.

Feature Driven Development

What is Feature Driven Development?

Feature Driven Development is a model driven, short iteration process.

Like any other development process, *Feature Driven Development* prescribes a series of steps to follow -- *from concept through design to implementation.*

Feature Driven Development

Why use Feature Driven Development?

Features illustrate *business value* in terms the end user can immediately understand.

Features can be used to *demonstrate incremental value* from the development process to the end user.

Features readily *support object-oriented programming* techniques.

Feature Driven Development

Why use Feature Driven Development?

Accommodates shorter business cycles.

Demonstrates frequent results.

Provides knowledge of exactly how far along the project is *at any point in time*.

Is scaleable to the size and criticality of the project.

Feature Driven Development

Feature Driven Development:

Features are very small blocks of *client-valued functionality*.

Features are organized into business-related groupings, referred to as *Feature Sets* and *Major Feature Sets*.

Feature Driven Development

Features are articulated by using *action-oriented* statements.

An example:

Major Feature Set:

Sales Order Management

Feature Set:

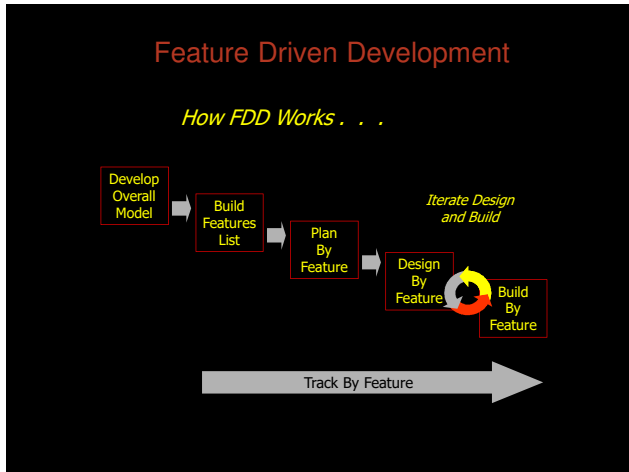
Process a Sales Order

Features:

Assign unique order number.

Calculate sales tax.

Calculate total of sale.



Feature Driven Development

Develop Overall Model:

Developers work closely with Business Domain Experts to establish the “business purpose” of the new system.

Once the business purpose has been established, Designers construct a “conceptual model” of the system.

Feature Driven Development

Build Features List:

Designers and Business Domain Experts use the conceptual model as a framework for developing a comprehensive “Features List”.

Features are grouped into “Feature Sets” that relate to setting up, conducting business and assessing results.

Feature Driven Development

Build Features List:

The “Features List” is prioritized based upon business value and business criticality to arrive at a “minimum whole product”.

This “minimum whole product” becomes the base, initial or “first” release of the software product.

Feature Driven Development

Plan by Feature:

A "launch date" is established for the "initial release".

The prioritized Features List, or "minimum whole product" is refined and finalized to insure Features represent small chunks of measurable development work.

Given, the launch date, the technical work is planned out and assigned.

Feature Driven Development

Design by Feature:

Role-players, roles and transactions are identified on a Feature-by-Feature basis.

User Interface "mock-ups" are sketched out.

An overall "architecture" for the system is designed before any "building" occurs.

Business Domain Experts are consulted on an iterative basis to uncover any additional specifics necessary to "design and build" each Feature.

Feature Driven Development

Build by Feature:

Features are "built" by implementing all necessary classes and methods.

Unit and "end-to-end" Feature testing occurs.

When the Feature is successfully working, all related classes to implement the Feature are "promoted" and the Feature is considered "complete".

Plan, Track and Report by Feature

Iterate Design and Build:

The Design and Build Steps are then repeated, for each additional Feature or Feature Set, leading up to a completed, deployable "initial release" of the system for the agreed upon "launch date".

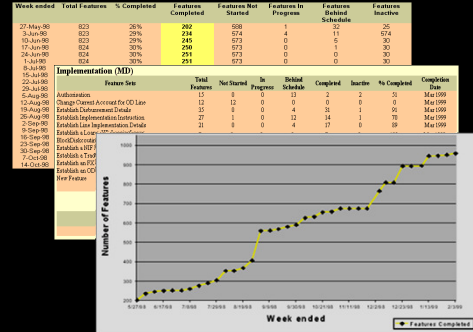
Plan, Track and Report *by Feature*

Track By Feature:

Each Feature is a unitary deliverable and a plannable unit of work.

The sum of the Features delivered equals the project status.

Plan, Track and Report *by Feature*



Feature Driven Development

In summary, Feature Driven Development:

Provides Clarity . . .

High level design is driven by Features needed to fulfill business needs

Elevates Control . . .

Features break down the work into smaller, more manageable deliverables

Facilitates Communication . . .

Overall project status is determined by the Features delivered

Who's Behind *Feature Driven Development*

This presentation presented feature driven development as an *alternative* approach to software development.

The feature driven methodology presented here is adapted from the works of:

Peter Coad, Mark Mayfield, Eric Lefebvre and Jeff De Luca

Materials regarding the use of "lighter" methodologies and "project-based" methodologies were adapted from the works of:

Edward Yourdon, Alan Davis and Alistair Cockburn