TOCICO CONFERENCE 2004

Feature-Driven Development

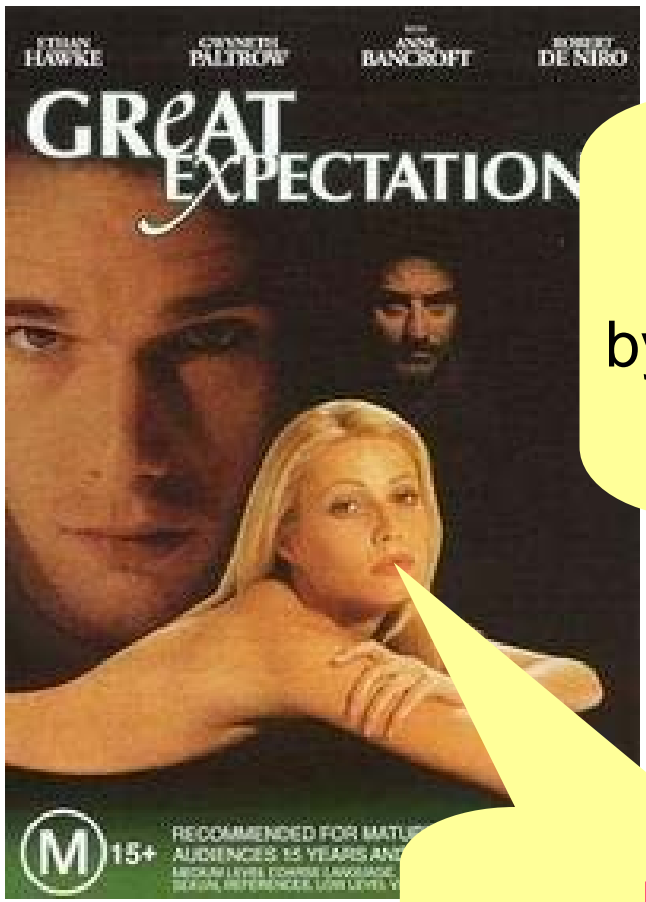
Towards a TOC, Lean & Six Sigma Solution for Software Engineering

Presented By: David J. Anderson, Microsoft Corp.

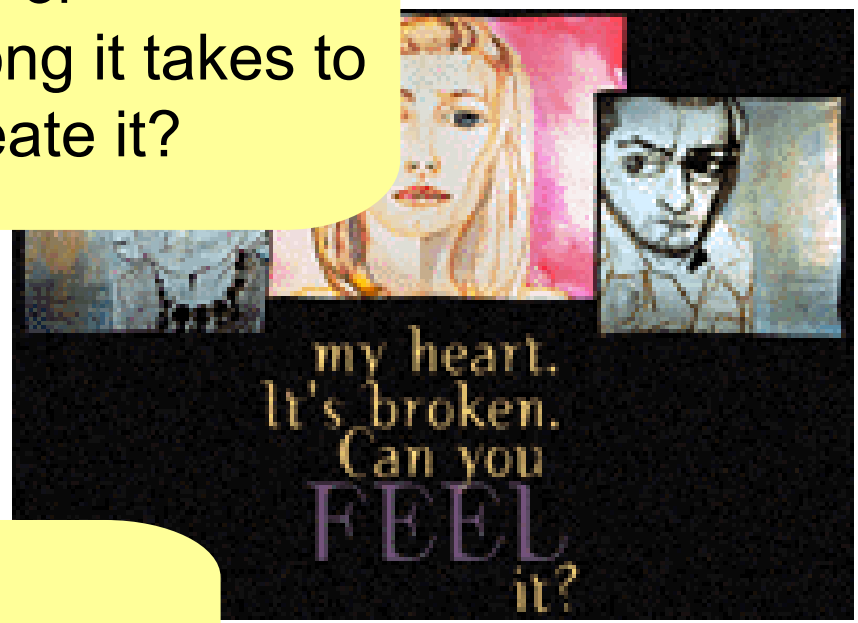
Date: 25th October 2004

Track: Expert E 2.4

How Do You Charge For Your Art?

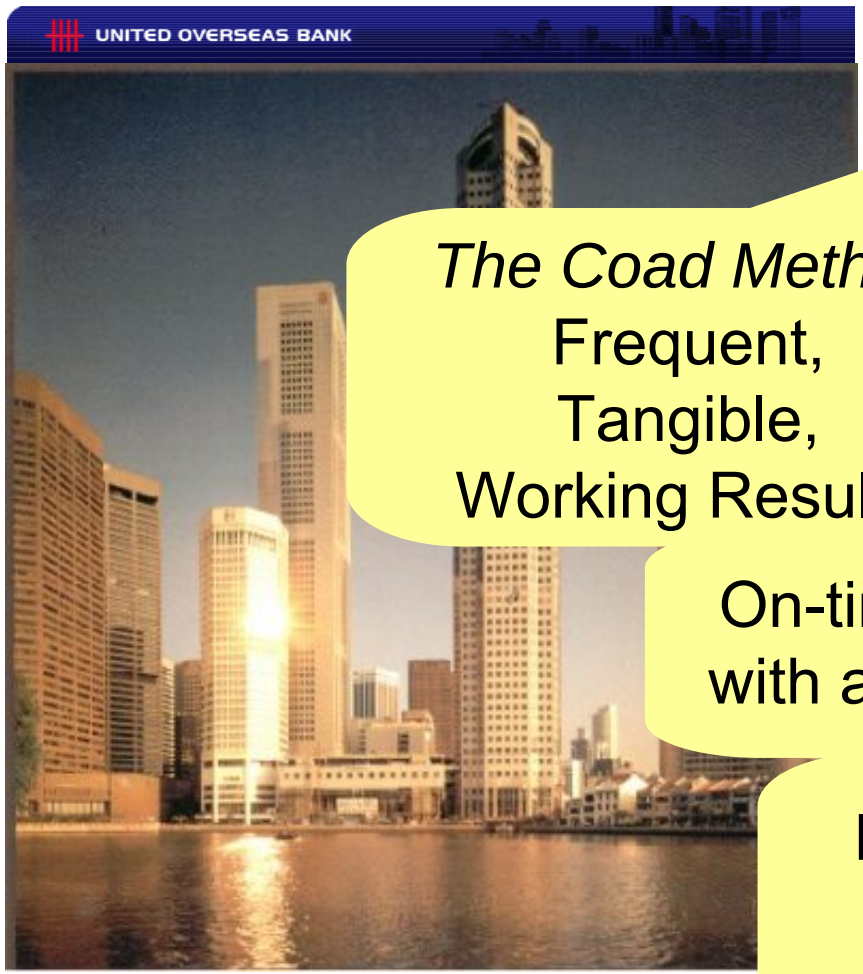


By the square inch
or
by how long it takes to create it?



By how beautiful it is!

Singapore Story



Peter Coad

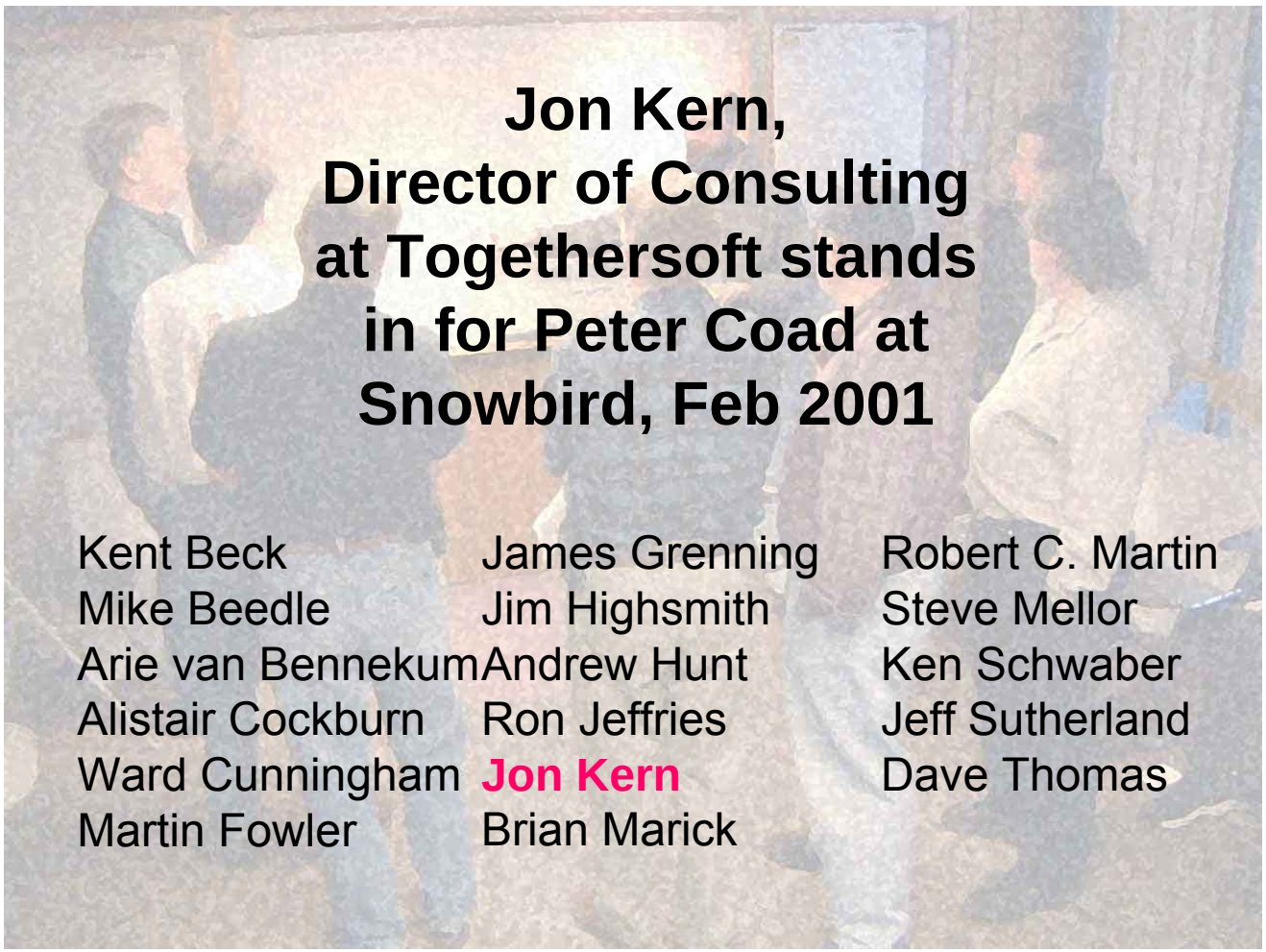
The Coad Method
Frequent,
Tangible,
Working Results

On-time, on-budget
with agreed function

Jeff De Luca

De Luca's 1st Law
80% Psychology,
20% Technology

Manifesto for Agile Development



Jon Kern,
Director of Consulting
at Togethersoft stands
in for Peter Coad at
Snowbird, Feb 2001

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	**Jon Kern**	Dave Thomas
Martin Fowler	Brian Marick	

Paradigm Shifting Dilemma

Feature Driven Development (FDD)

- **Highly Effective**
- **High Quality**
- **Faster To Market**
- **Team working**
- **No Overtime**
- **Productivity**
  - **2 to 10 fold improvement**
- **Quality improvement**
  - **3:1 to 2:100**

But…

- **No Time Tracking**
- **No Gantt Charts**
- **No Task Tracking**
- **No Time on Task Estimates**

How to communicate it
to a skeptical audience?

Engineering process

Marketing Requirements

Engineering Lead Time

Develop an Overall Model

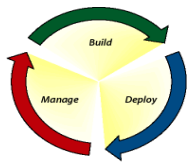
Build **Feature** List

Plan By Feature

Design By Feature

Build By Feature

Finished Code

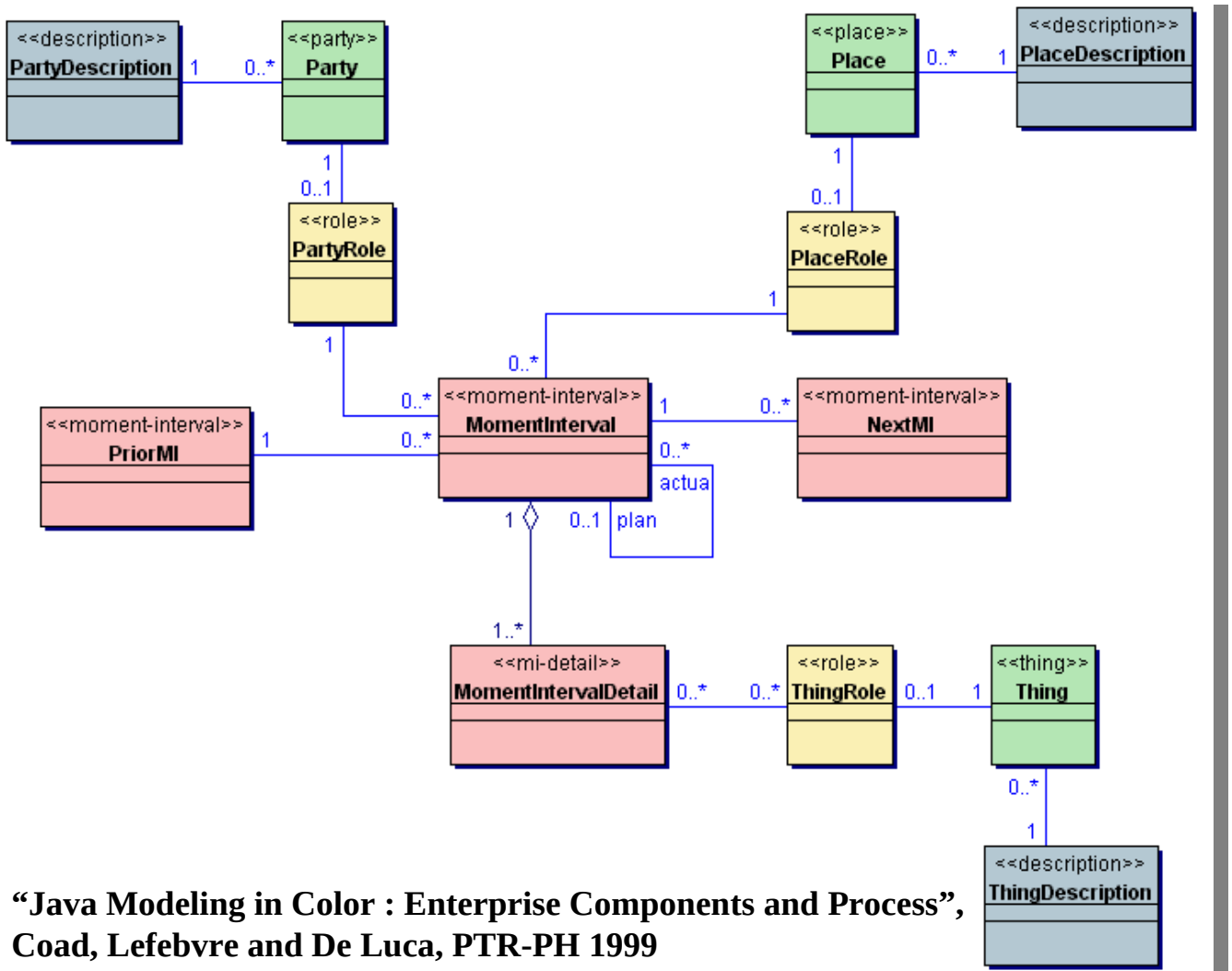


Weekly Build Cycle

Test By Feature

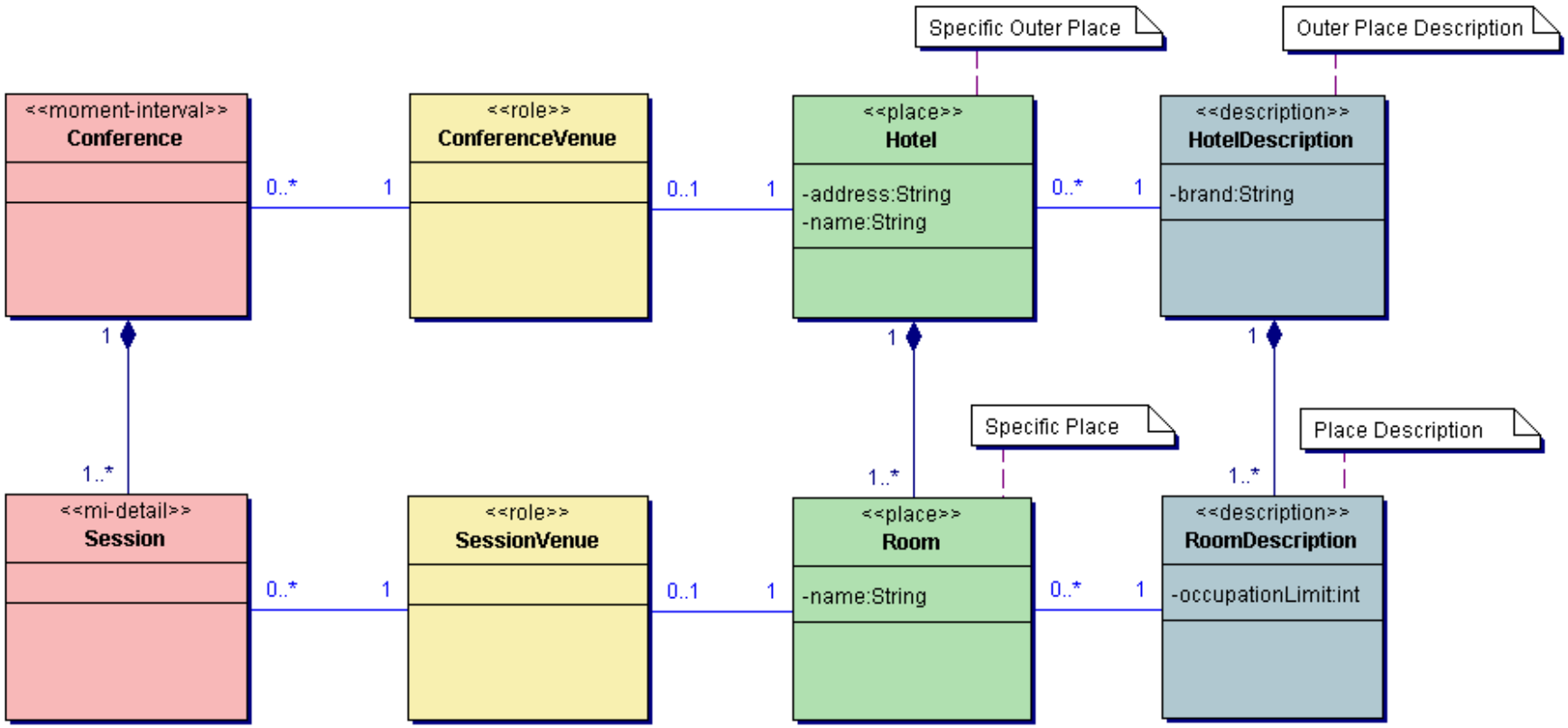
Bug Reports

Domain Model Meta-Pattern



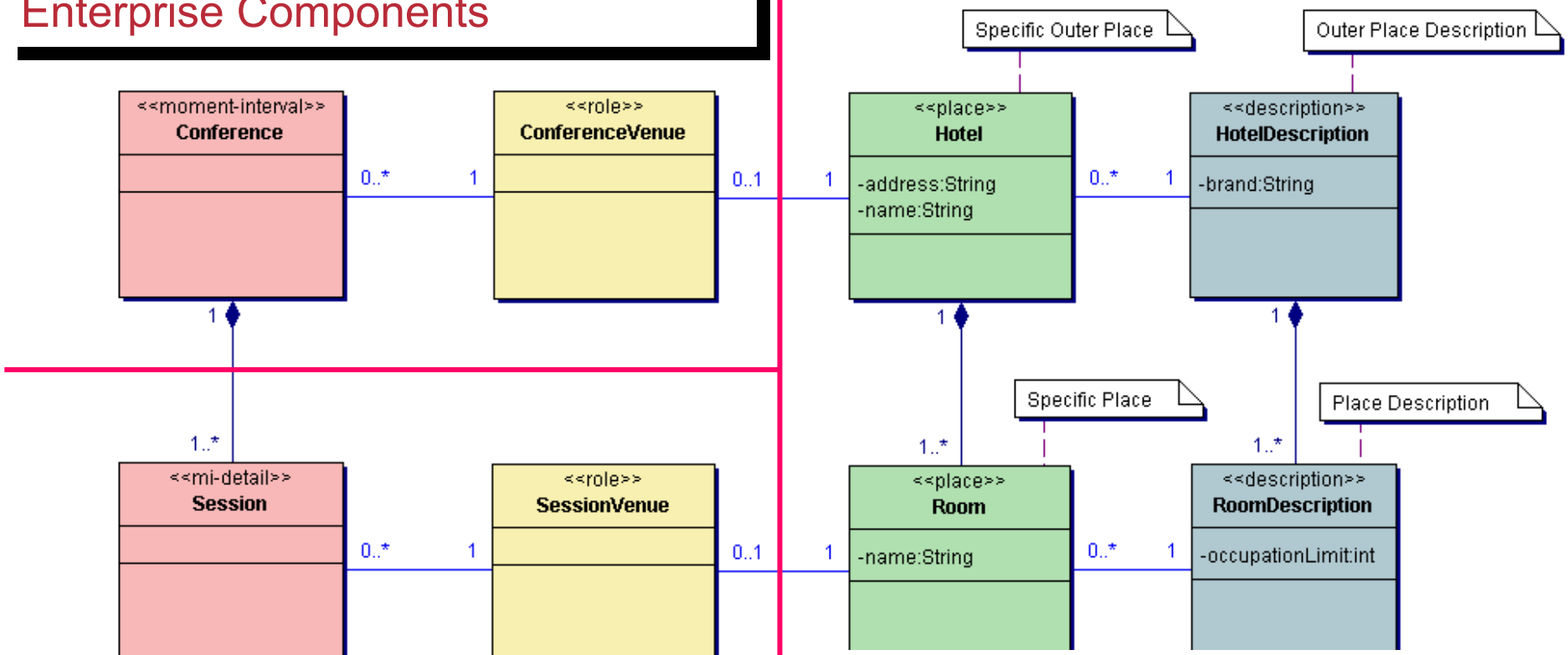
"Java Modeling in Color : Enterprise Components and Process", Coad, Lefebvre and De Luca, PTR-PH 1999

Conference Application Example



Postponement (of componet definition)

Pinks and yellows are re-usable across multiple greens – the core Enterprise Components



Greens and blues are re-usable across discrete Enterprise Applications modeled as sequences of pinks

Definition of a Feature

- **Tiny piece of client-valued functionality which can be delivered in less than 2 man weeks**

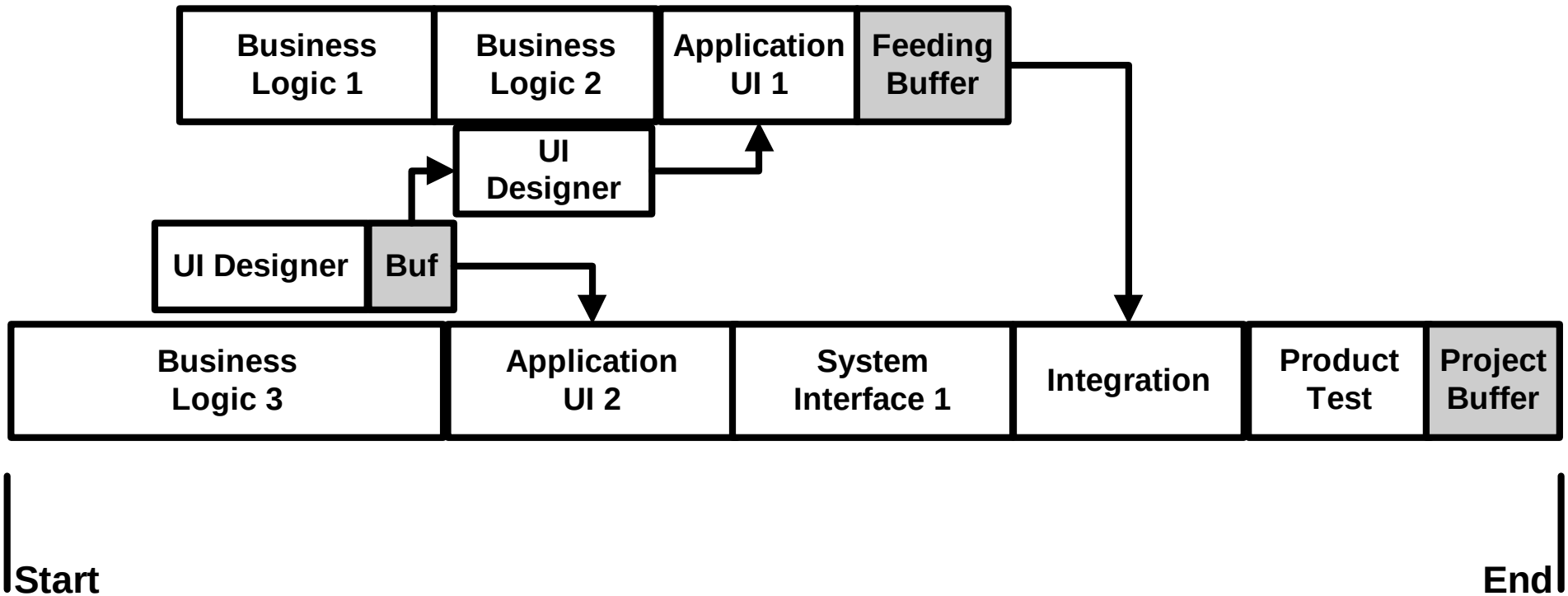
- **Business Logic Feature**
  - **<action> <result> [of|to|from|for] <object>**
  - **E.g. list availability of conference venues for given dates and attendee numbers**

Build a Feature List

- **Prioritize Features**
  - **1 to 5 how important to be "in" release**
  - **1 to 5 how problematic if left "out" of release**
- **Group Features into Sets and Sets into Subject Areas**
- **Groupings filtered by Release will be used for scheduling**

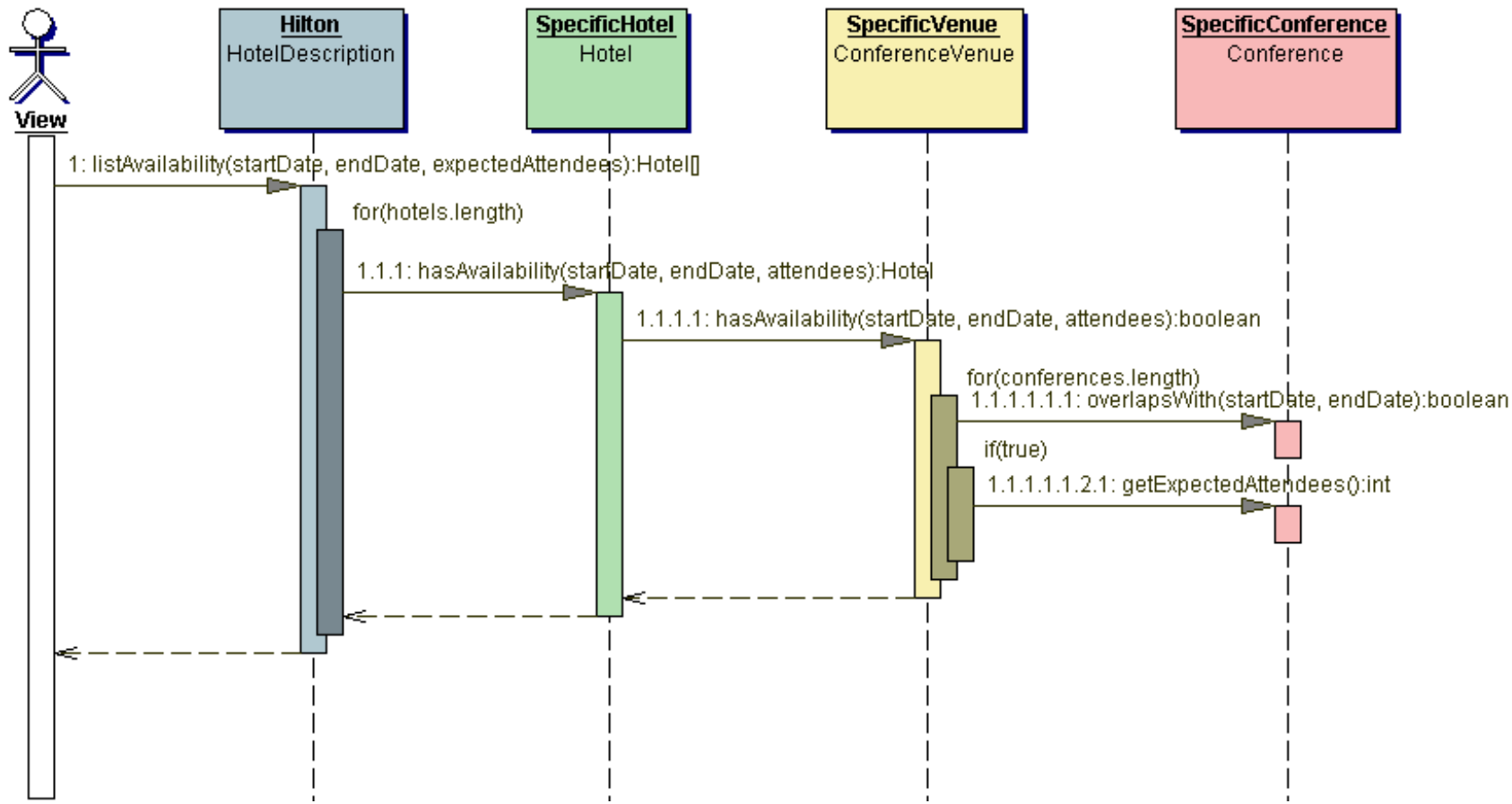
Feature / Feature	Release 1			Release 2			Release 3		
	IN	OUT	TOTAL	IN	OUT	TOTAL	IN	OUT	TOTAL
Set the Milestones for a Feature	5	5	10	5	3	8	5	5	10
List the Features for the Project/Release	5	5	10	5	5	10	5	5	10
Set the CPW for the Feature	4	3	7	5	5	10	5	5	10
Set the Subject Area for the Feature Set	4	3	7	5	5	10	5	5	10
Set the Feature Set for the Feature	3	3	6	5	5	10	5	5	10
List the Virtual Team members for the CPW	4	2	6	4	4	8	5	5	10
List the Feature Completion Dates for the Release	3	1	4	4	4	8	5	4	9
List the Feature Completion Dates for the CPW	3	3	6	4	4	8	5	5	10
Total the Features for a Product	3	1	4	4	1	5	5	5	10
Total the number of open issues in the Issue Log for a given	1	1	2	3	1	4	5	5	10
List Change Requests for the Release	1	1	2	2	2	4	4	5	9
List the Subject Areas for the Product	2	1	3	3	2	5	4	4	8
List all Feature Sets for the Subject Area	2	1	3	3	2	5	4	4	8

Critical Chain Schedule

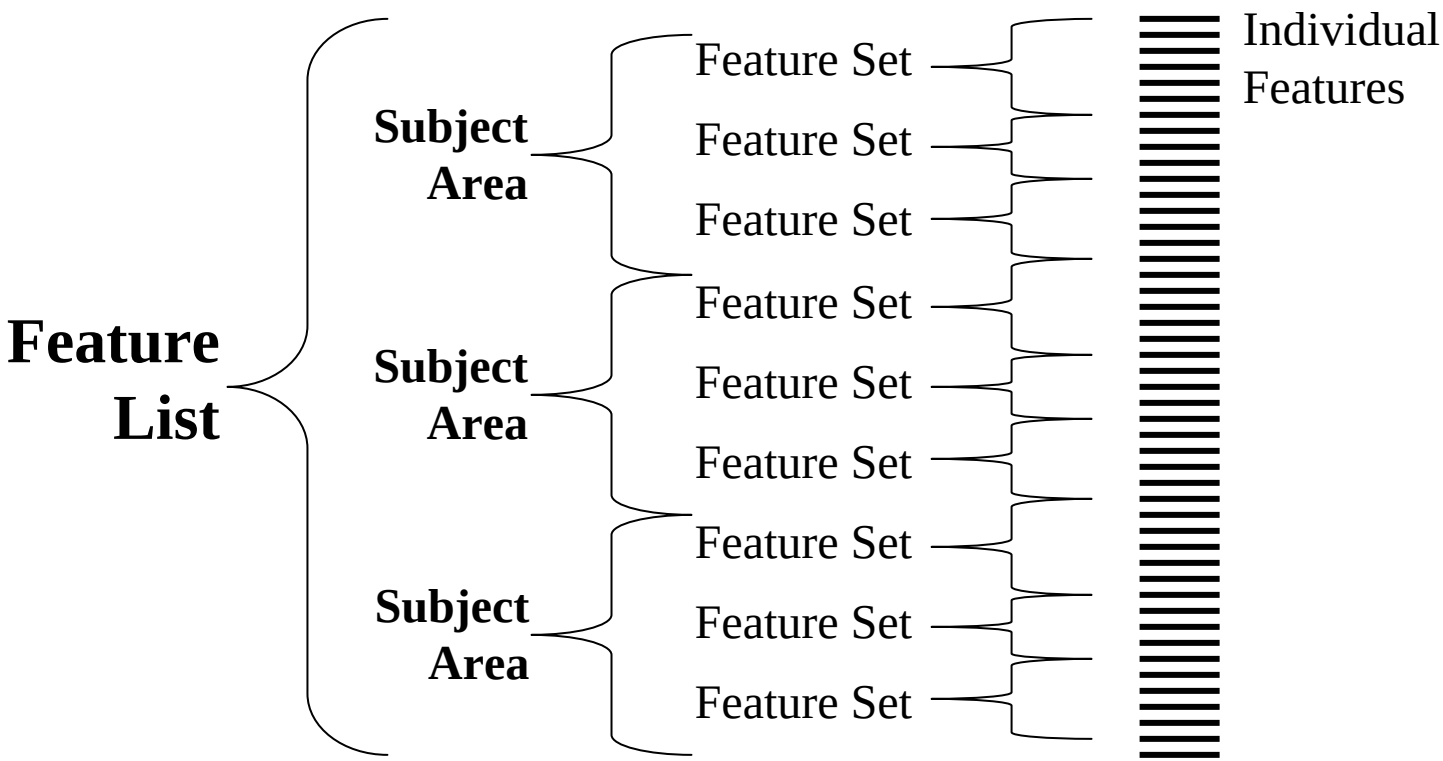


- Schedule based on Feature Set groupings
- Buffers aggregated across many Features
- This example has UI Designer as system constraint

Features Design UML Sequence Diagram



FDD – How it works



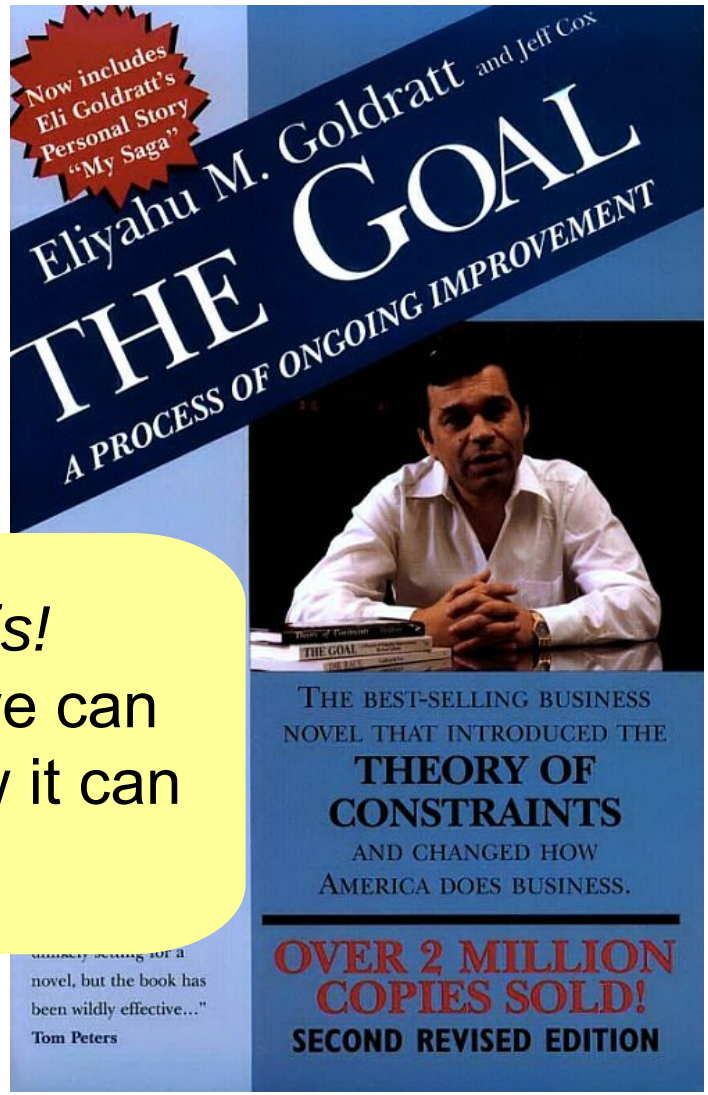
Epiphany!

Sprint.

Features As Inventory!

Read This!

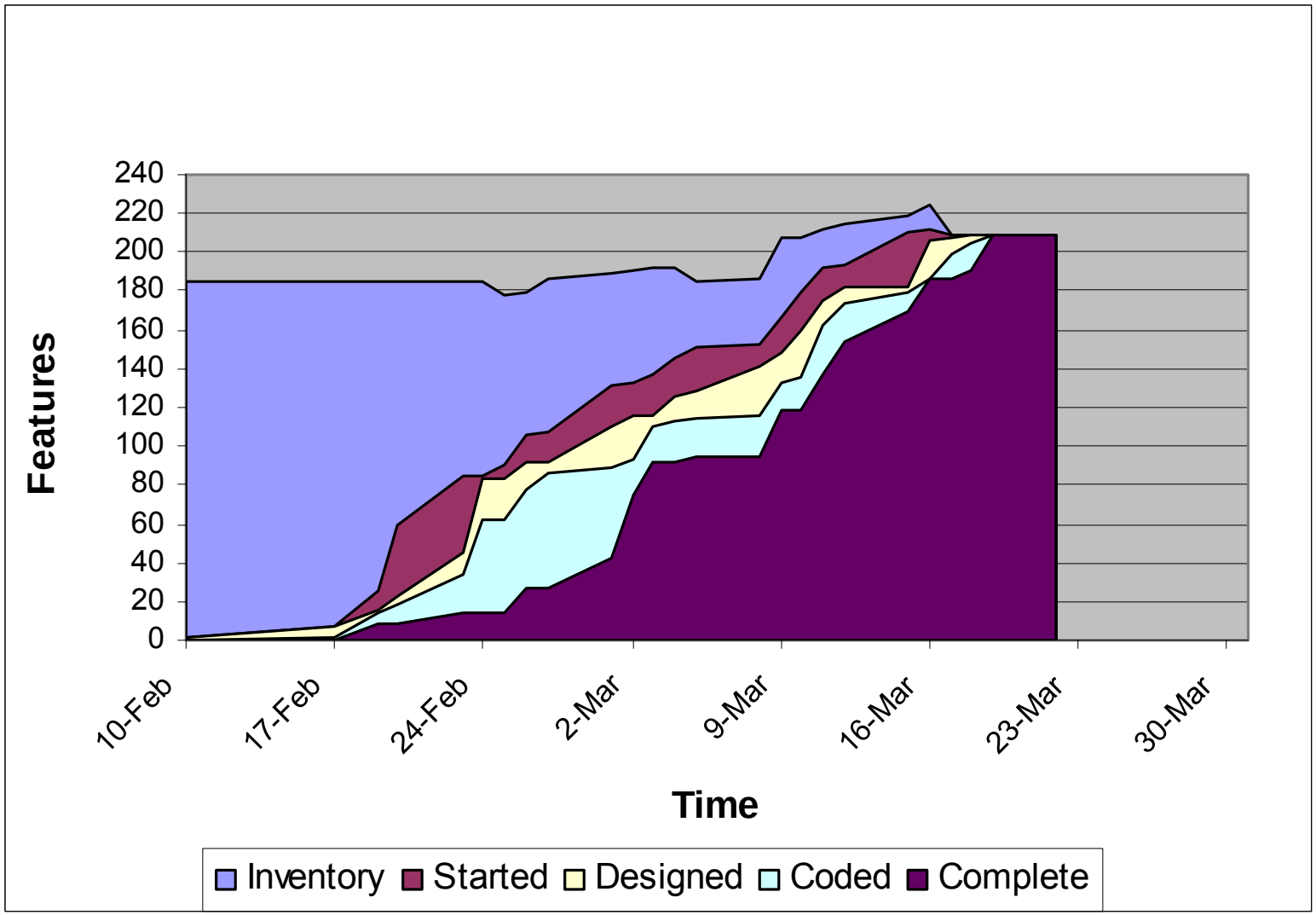
Let's see if we can figure out how it can help!



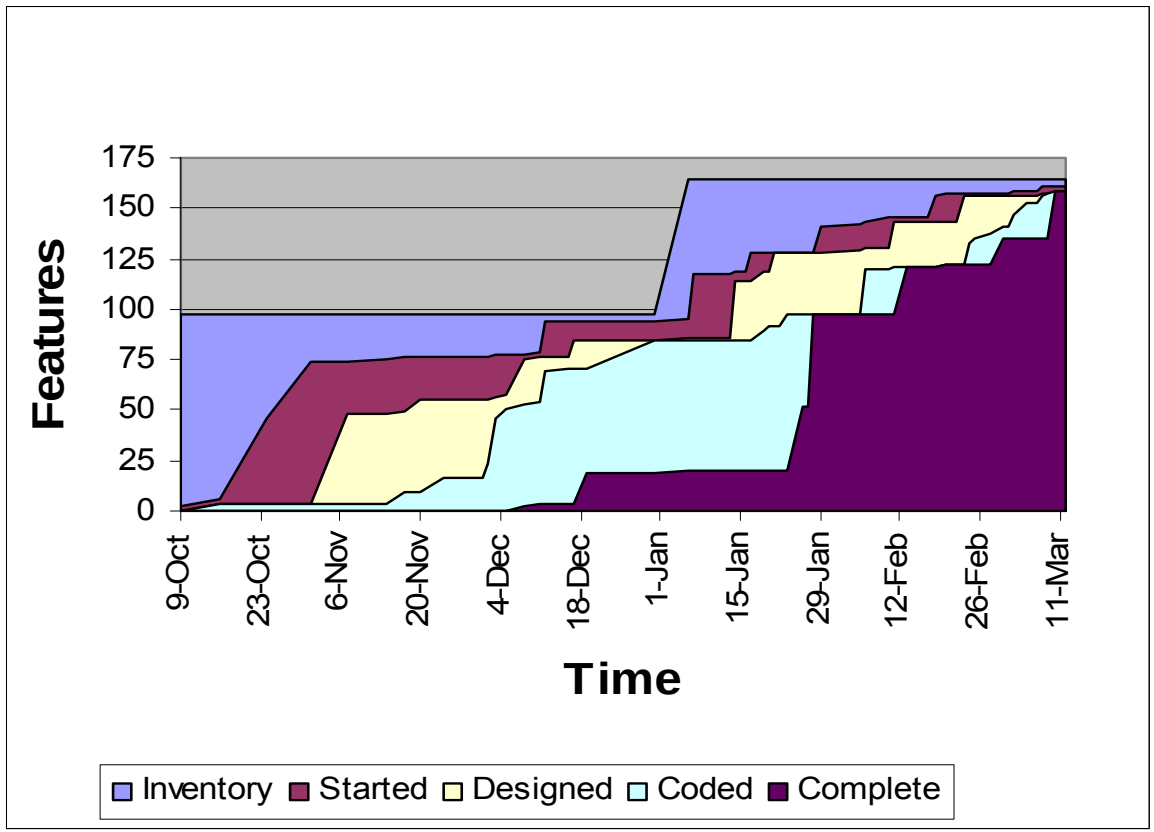
Managing The Design Factory

- Marvin Patterson **(1993)**
  - **design is a process of discovery of information**
  - **Flow of value is achieved by increasing the certainty of information being discovered**
- Donald Reinertsen **(1997)**
  - **design processes can track the flow of information discovery, design in process (DIP) is analogous to "inventory" in production processes**
  - **Track with Cumulative Flow Diagram from Lean Production**
  - **Design is perishable – inventory depreciates over time**

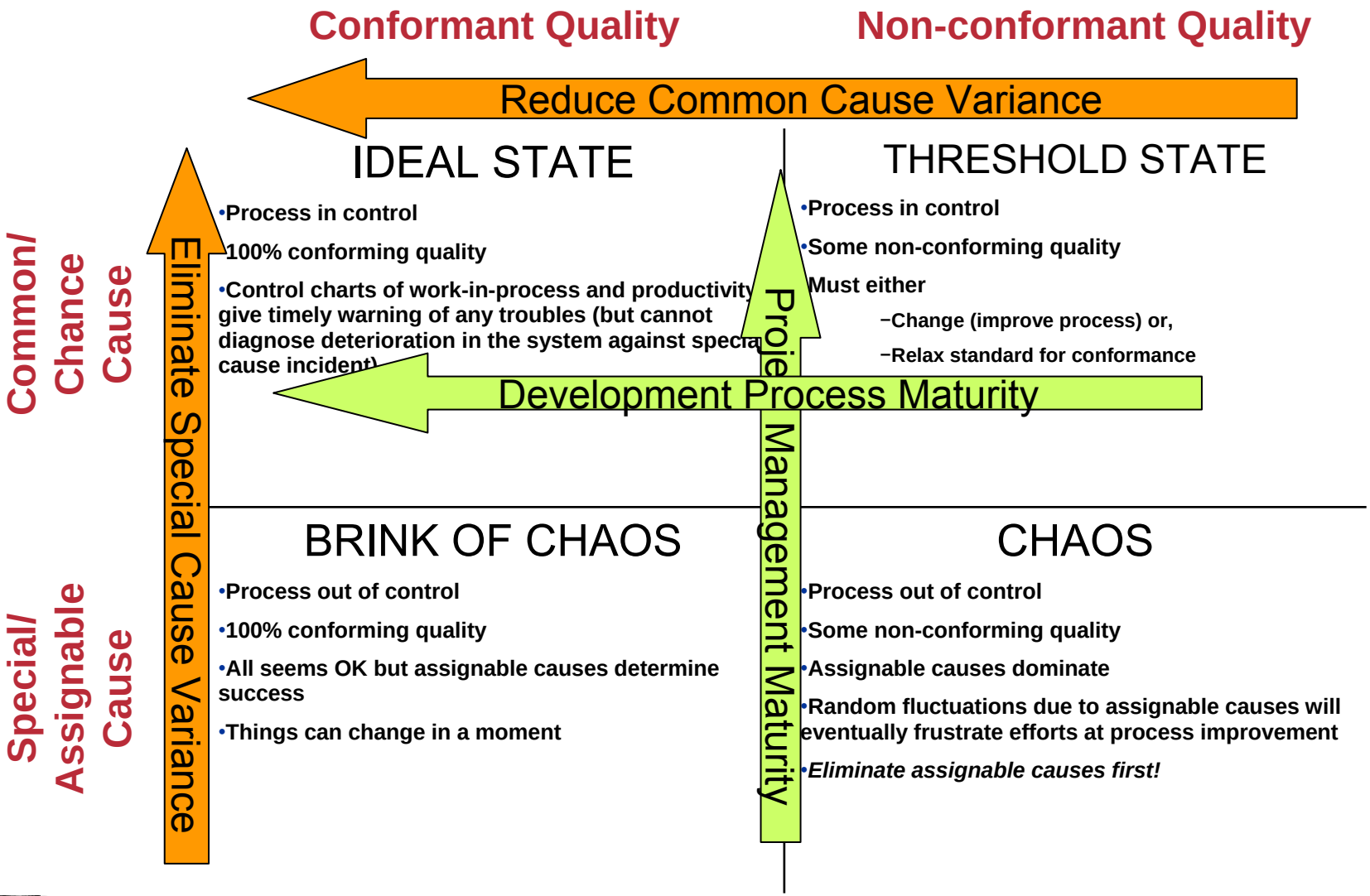
Achieving Smooth Flow



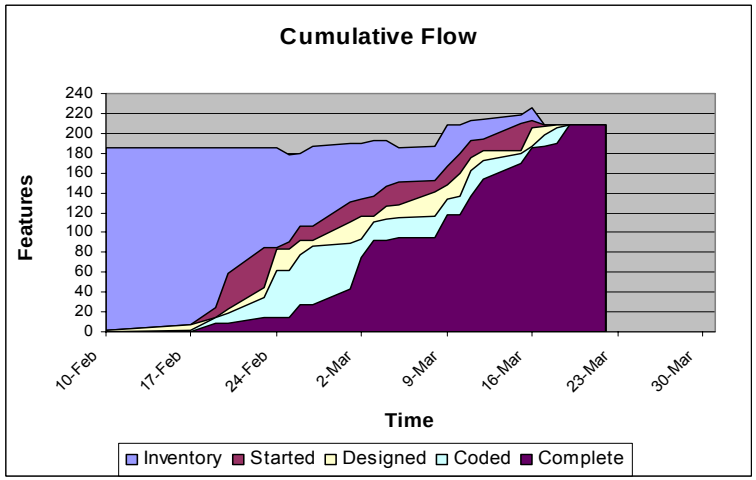
Ragged Flow

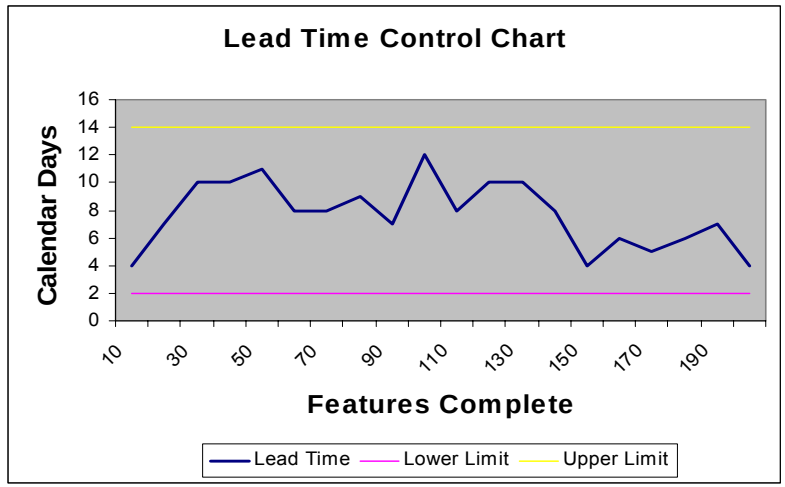


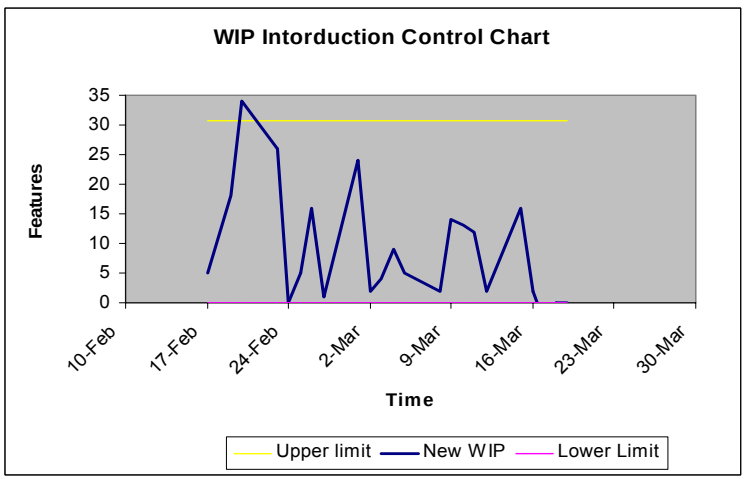
Variation and Six Sigma

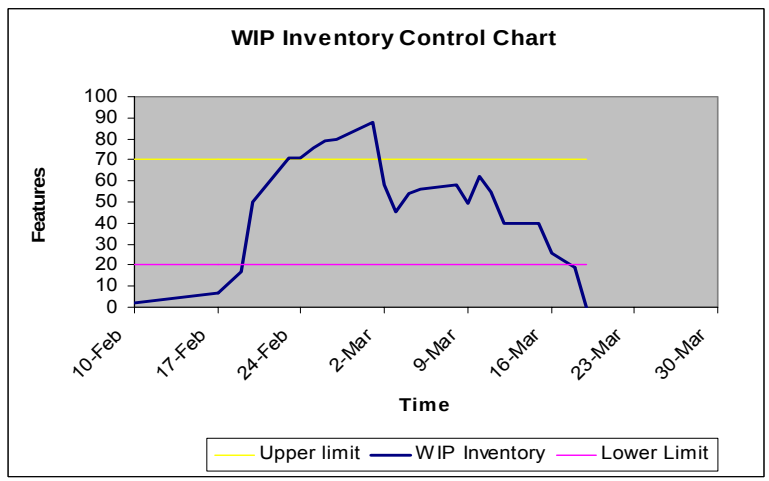


Using Control Charts with CFDs

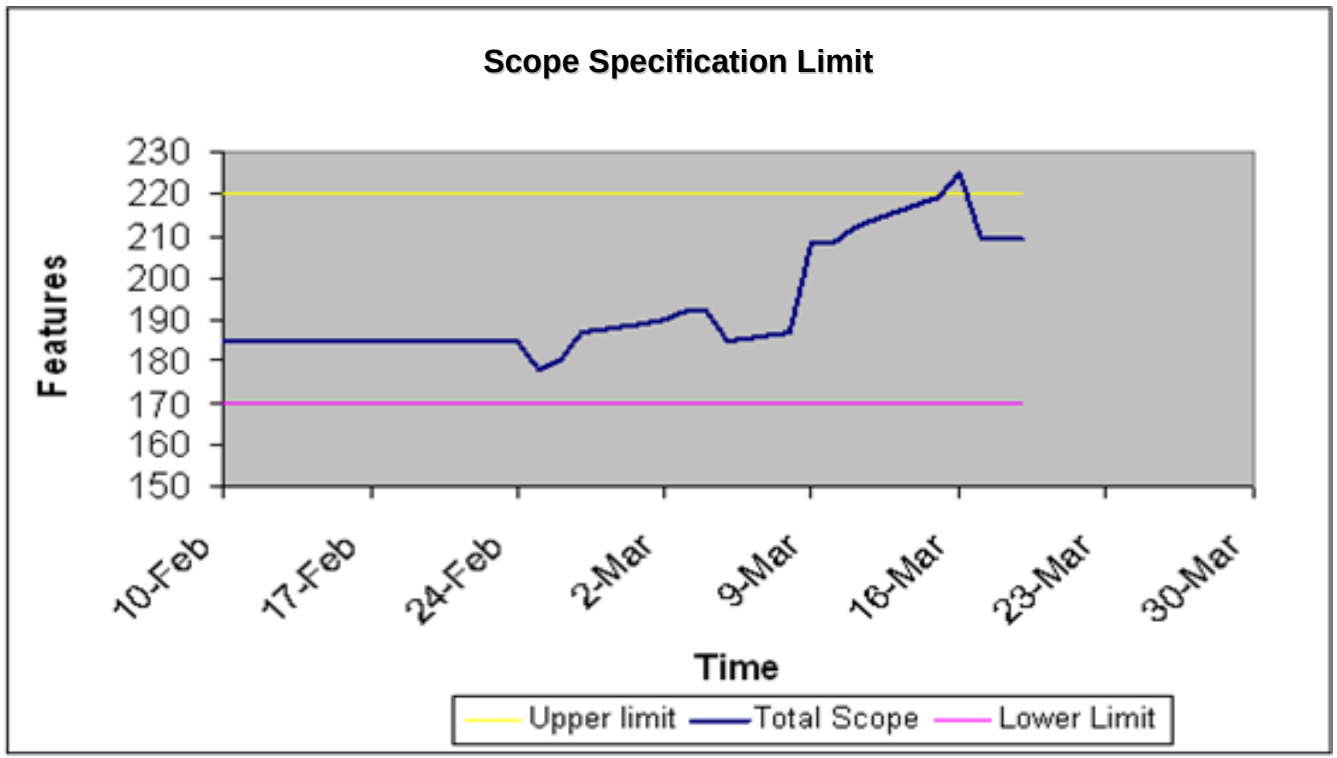




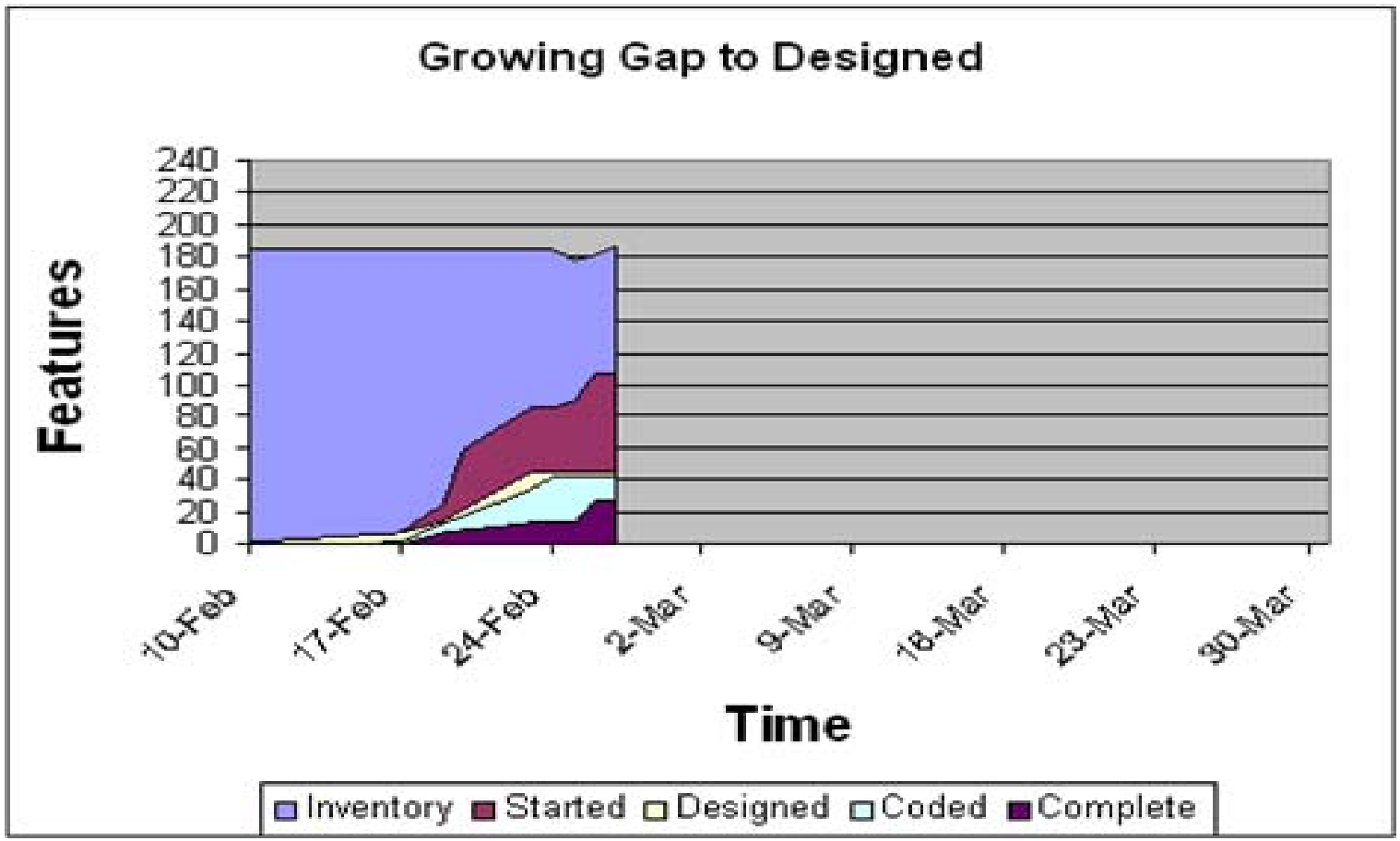




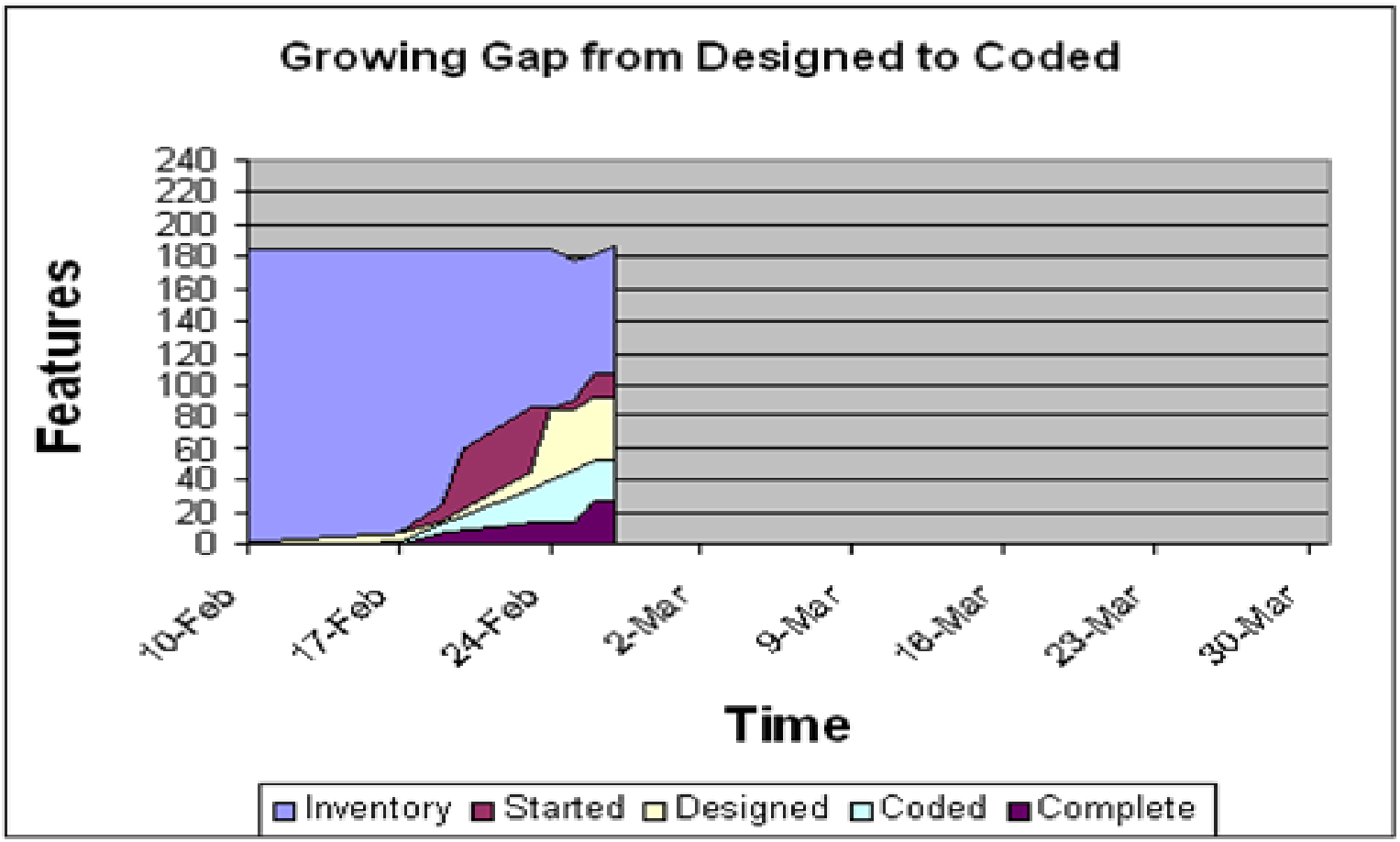
Buffering Scope Uncertainty



Requirements Uncertainty

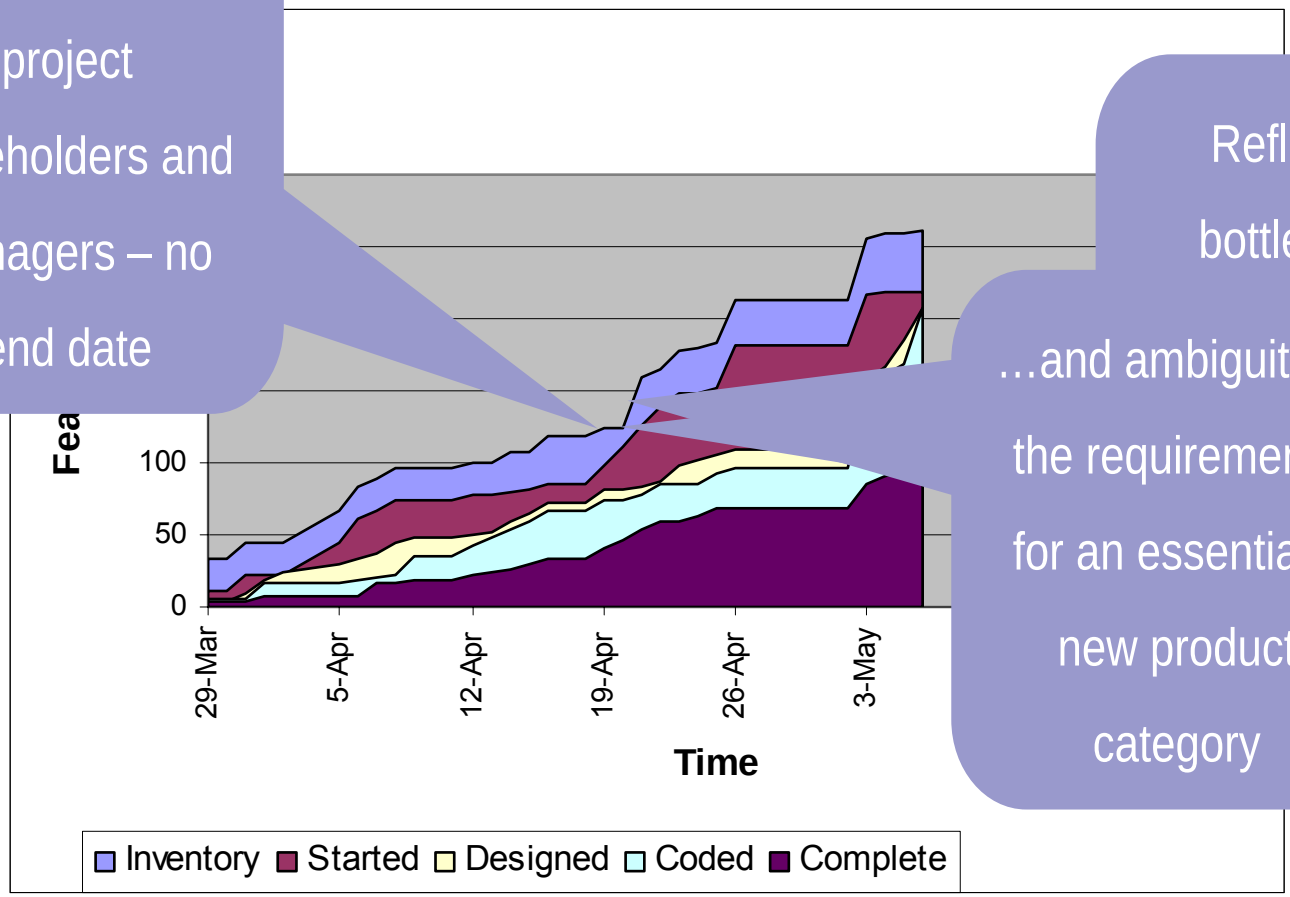


Architecture or Refactoring



Just-in-time Domain Analysis

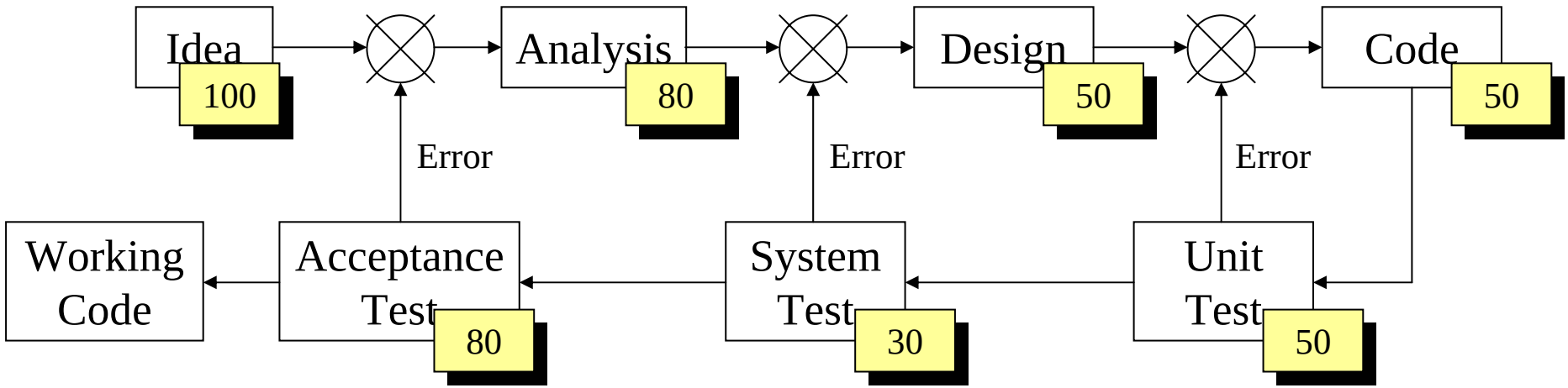
Problematic for project stakeholders and managers – no end date



Reflected a bottleneck in … for … ysis … ure…

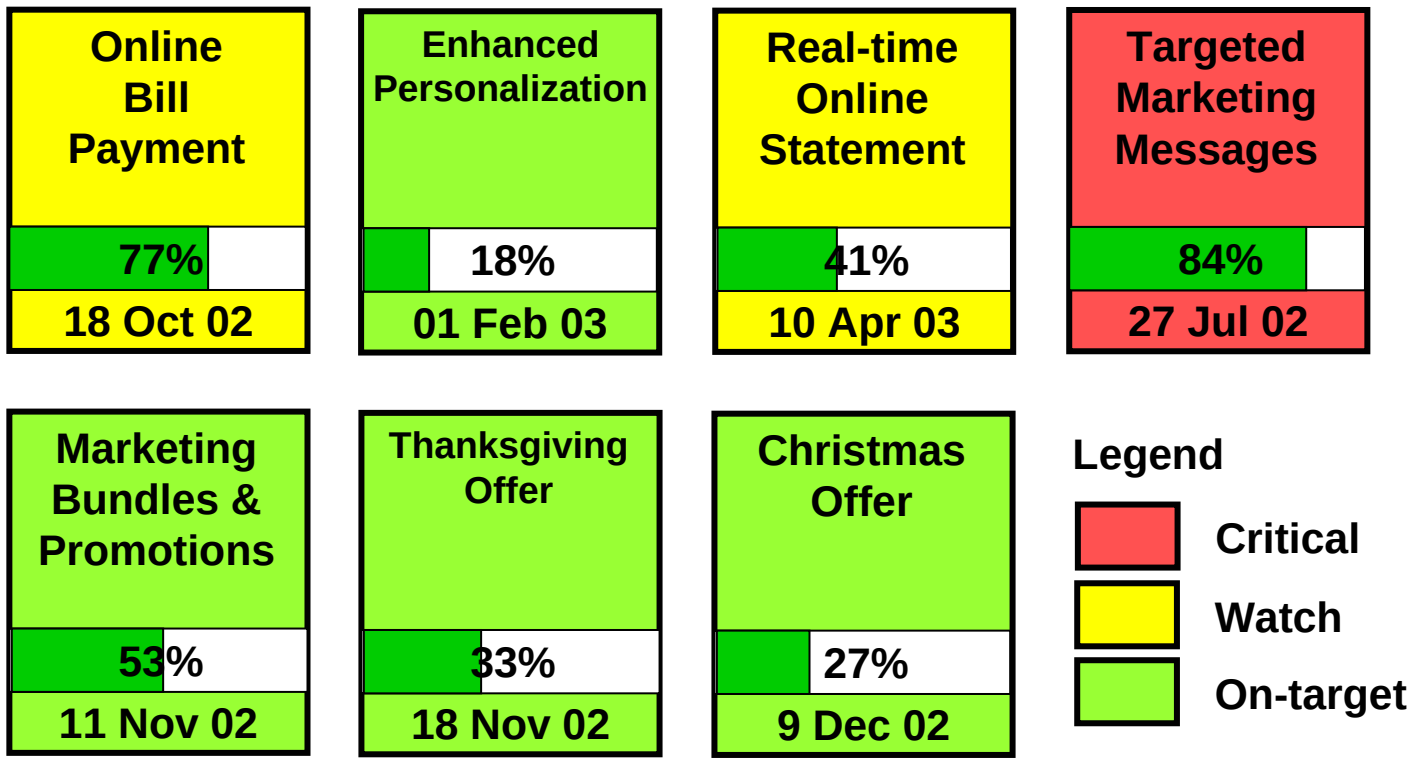
…and ambiguity in the requirements for an essentially new product category

Ops Review – Identify Constraint



- **Current CCR is System Test**
- **Testers relieved of all non-essential tasks, extra PMs assigned to complete administrative tasks, Analysts assigned to future Test Plans**
- **Requirements release restricted to 100 per quarter**
- **Plan to recruit 5 temporary staff immediately**

Report Buffer Usage



Throughput Accounting

Requirement / Release	Group	Rating
List the Features for the Project/Release	1	Essential
Set the Milestones for a Feature	1	Essential
Set the CPW for the Feature	1	Essential
Set the Subject Area for the Feature Set	1	Essential
List the Feature Completion Dates for the CPW	1	Essential
List the Virtual Team members for the CPW	1	Essential
Set the Feature Set for the Feature	1	Essential
List the Feature Completion Dates for the Release	2	Essential
Total the Features for a Product	2	Essential
List all Feature Sets for the Subject Area	3	Desired
List the Subject Areas for the Product	3	Desired
List Change Requests for the Release	4	Optional
Total the number of open issues in the Issue Log for a given Release	4	Optional

Price per Seat / User Role	Project Mgr	Development Mgr	Program Mgr	Product Mgr
Free				
$0 - $500	Feature Group 4			
$500 - $1,000				
> $1,000		Feature Group 1 Group 2	Feature Group 2 Group 3	

Table 16-2 (page 145)

Price per Seat / User Role	Project Mgr	Development Mgr	Program Mgr	Product Mgr
Free				
$0 - $500	$10,000			
$500 - $1,000				
> $1,000		$70,000 $30,000	$30,000 $20,000	

Table 16-3 (page 145)

Product Mix Prioritization

Requirement / Release	Group	# Man Hours to Test	Group Total Testing	Throughput	$ / hour of CCR
List the Feature Completion Dates for the Release	2	32			
Total the Features for a Product	2	32	64	$60,000	$937.50
List the Features for the Project/Release	1	8			
Set the Milestones for a Feature	1	8			
Set the CPW for the Feature	1	16			
Set the Subject Area for the Feature Set	1	20			
List the Feature Completion Dates for the CPW	1	16			
List the Virtual Team members for the CPW	1	8			
Set the Feature Set for the Feature	1	8	100	$70,000	$700
List all Feature Sets for the Subject Area	3	16			
List the Subject Areas for the Product	3	20	36	$20,000	$555.56
List Change Requests for the Release	4	64			
Total the number of open issues in the Issue Log for a given Release	4	64	128	$10,000	$78.13

About David J. Anderson

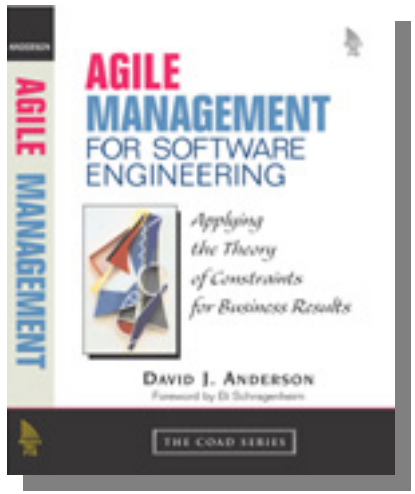
David Anderson is a software engineering methodologist and Program Manager with Microsoft Corporation in Redmond WA. He has 22 years experience in the software development business starting with computer games in the early 1980's. As a pioneer in the agile software movement David has run around 20 software projects in the Fortune 100. He is currently creating the next generation of MSF (Microsoft Solution Framework), a set of process guidance and development tooling which enables the latest thinking in working practices and management techniques for software development.

David authored the popular and well received textbook, Agile Management for Software Engineering – Applying the Theory of Constraints for Business Results, published in 2003 by Prentice Hall, which introduced the concepts of Drum-Buffer-Rope, Critical Chain and Throughput Accounting for Software Development.

David has held management positions with Sprint PCS and Motorola before being attracted to Microsoft and the opportunity to bring his paradigm shifting thinking in software management to a wider audience.

He holds a degree in Computer Science & Electronics from the University of Strathclyde, Glasgow, Scotland where he specialized in control systems engineering.

Contact Details



David J. Anderson
Program Manager, Microsoft
dander@microsoft.com
http://www.agilemanagement.net/