~~Light~~ Methodologies: Problems, Principles, and Practices

Jim Highsmith
Director, E-Project Management Practice
Cutter Consortium

jimh@adaptivesd.com

Problem Domains: Exploration versus Exploitation

Exploration Drilling

Production Drilling

The Path thru the Quagmire of the Future

- RADICAL INNOVATION
- EXPLORATORY DEVELOPMENT
- COLLABORATION
- HARNESSING CHANGE

"Radical innovation is the competitive advantage for the new millenium."
-- Gary Hamel, *Leading the Revolution*

The Diabolical Challenge of Modern Software Development

To rapidly complete large projects that are both research-like and mission-critical in a turbulent business and technology environment.

Exciting Features

Rapid delivery

High quality

High change

Cutter Consortium's e-Project Research Findings

- Sample size
  - 40 companies
  - 37 projects in-process per company
- Importance
  - 9 (20%) e-Projects/company
  - 31% of budget on e-Projects
- Growth
  - 20 new e-Projects being initiated each year
  - 18 e-Projects/Comp "in-process" in 12 mths

Cutter Consortium's e-Project Research Findings

- e-Project Size
  - 11 months avg. schedule
  - 40 person avg. staff size
- Top 3 business activities supported
  - 60% customer service
  - 42% MIS/DSS (knowledge managemetn, data mining, CRM analysis, etc.)
  - 40% Order entry/fulfillment

Harvard Business School Studies

- Four software-development practices that spell success:
  - An early release of the evolving product design to customers
  - Daily incorporation of new software code and rapid feedback on design changes
  - A team with broad-based experience of shipping multiple projects
  - Major investments in the design of the product architecture.

"Product-Development Practices that Work: How Internet Companies Build Software," *MIT Sloan Management Review*, Winter 2001 by Harvard Business School professor Alan MacCormack.

Side-bar quote: "Now there is proof that the evolutionary approach to software development results in a ***speedier process and higher-quality products***."

HBS Study

- Key findings
  - Releasing early, low functionality versions of products resulted in dramatic differences in performance.
  - Evolutionary approaches reduce risks.
  - The quicker the feedback (hours) the higher the quality.
  - Uncertainty dictates short microprojects--down to the level of individual features.

Heavy-Agile Projects--What?

- HBS--ERP Systems Study--Rob Austin & Richard Nolan
- Horror stories
  - Large Manufacturer--$175 to $300 million overnight
  - Dell--Scrapped $200 million project
  - Fox Meyer Drug--$5 billion company bankrupt
- Characteristics
  - Very large, often largest company has ever undertaken
  - Very risky--technical, organizational, business
- "The old project approaches do not work in this new space." Rob Austin

HBS--ERP Systems Study--Austin & Nolan

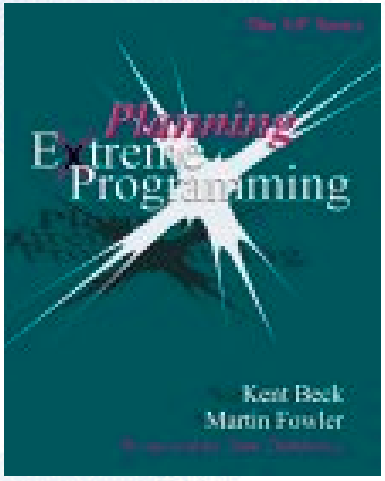
- "The first flawed assumption is that it is actually possible to plan such a large project well enough that success is primarily determined by degree of conformance to a plan.
- "The second flawed assumption embedded in planning-intensive approaches is that it is possible to protect against late changes to a large system project.
- A third flawed assumption is that it even makes sense to lock in big project decisions early."

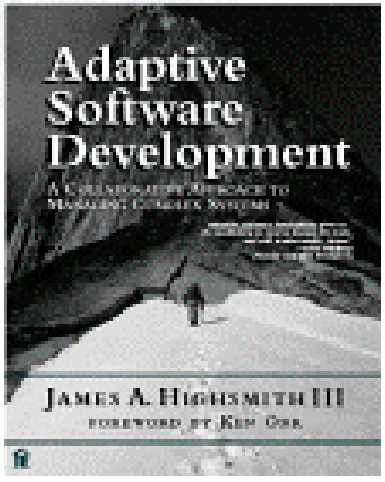
ERP Systems Study--Success Characteistics

- They are all iterative
- They all rely on fast cycles and insist on frequent delivery
- They get functionality in some form into business user hands very early in the project
- They are preceded by little or no traditional ROI-style analysis of the project as a monolithic whole.

Emerging Solution: Agile Methodologies

- Extreme Programming - Kent Beck +
- Crystal Methods - Alistair Cockburn
- Lean Development - Bob Charette
- SCRUM-K. Schwaber, J. Sutherland
- Adaptive Software Dev - Jim Highsmith





"I predict that Kent Beck and his XP movement will be as much a symbol of our times as Watts Humphry and the CMM were a symbol of the eighties and early nineties." - Tom DeMarco, Cutter Trends Report on Light Methodologies

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. We value:

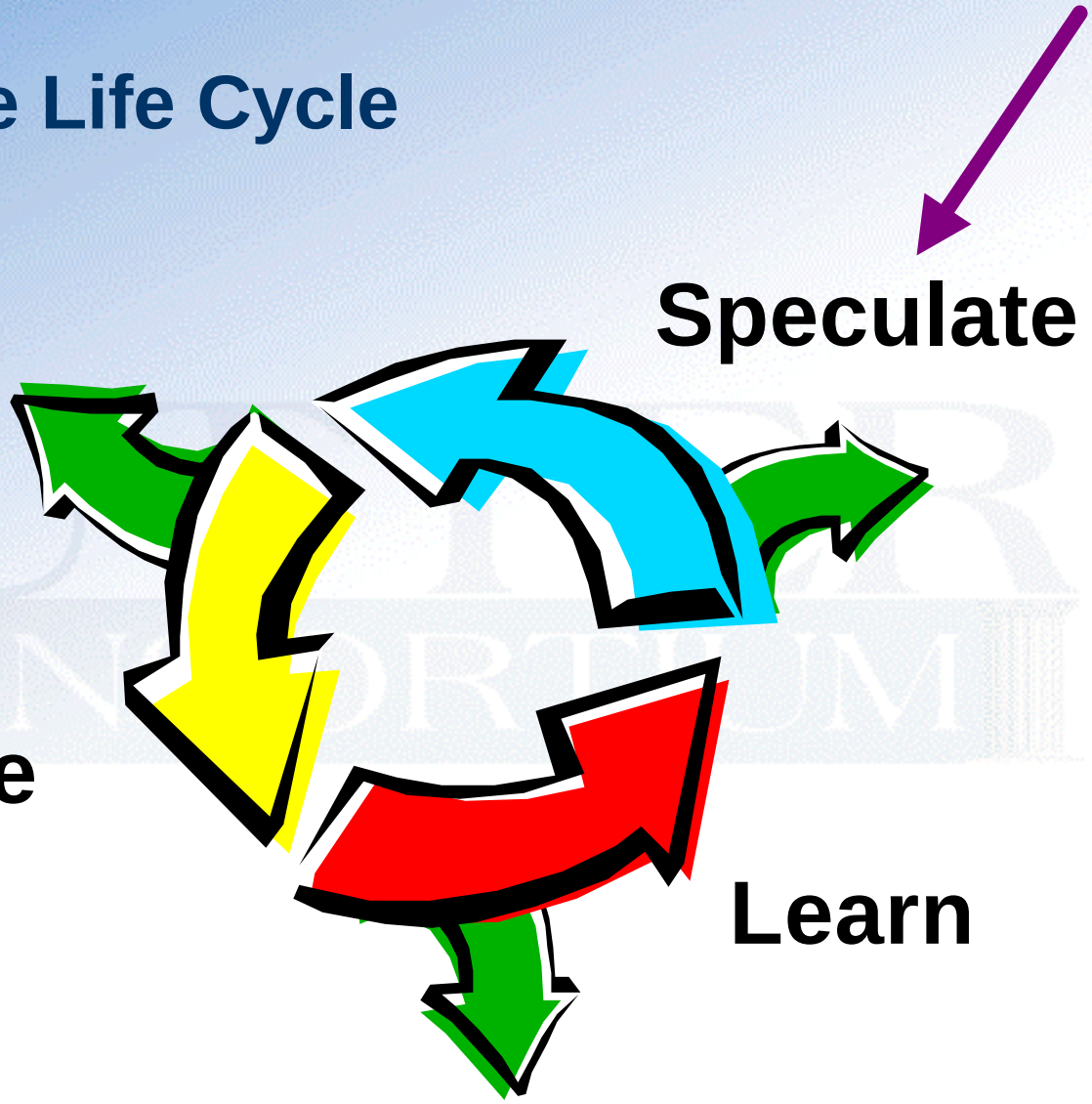
- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan.

Signers: Arie van Bennekum (DSDM), Kent Beck (XP), Mike Beedle, Alistair Cockburn (Crystal), Ward Cunningham (XP), Martin Fowler (XP), James Grenning (XP), Jim Highsmith (ASD), Andy Hunt (Pragmatic Programming), Ron Jeffries (XP), Jon Kern (FDD), Brian Marick, Robert Martin (XP), Steve Mellor, Ken Schwaber, (SCRUM), Jeff Sutherland (SCRUM), PragmaticDave Thomas (Pragmatic Programming).

Core Principles of Agile Methodologies

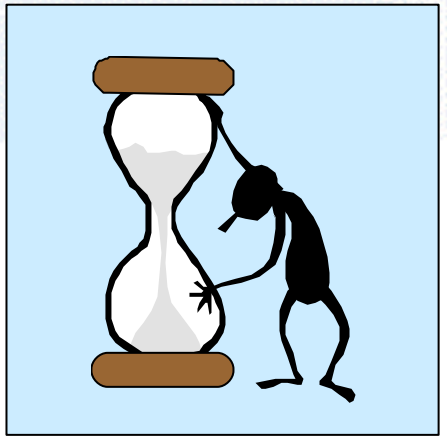
- Customer Value--Focus on results
- Individual Capability--Focus on individual skills
- Collaboration--Focus on innovation through group interaction
- Adaptation--Focus on feedback & harnessing change
- Minimalism--Focus on simplicity

The Adaptive Life Cycle

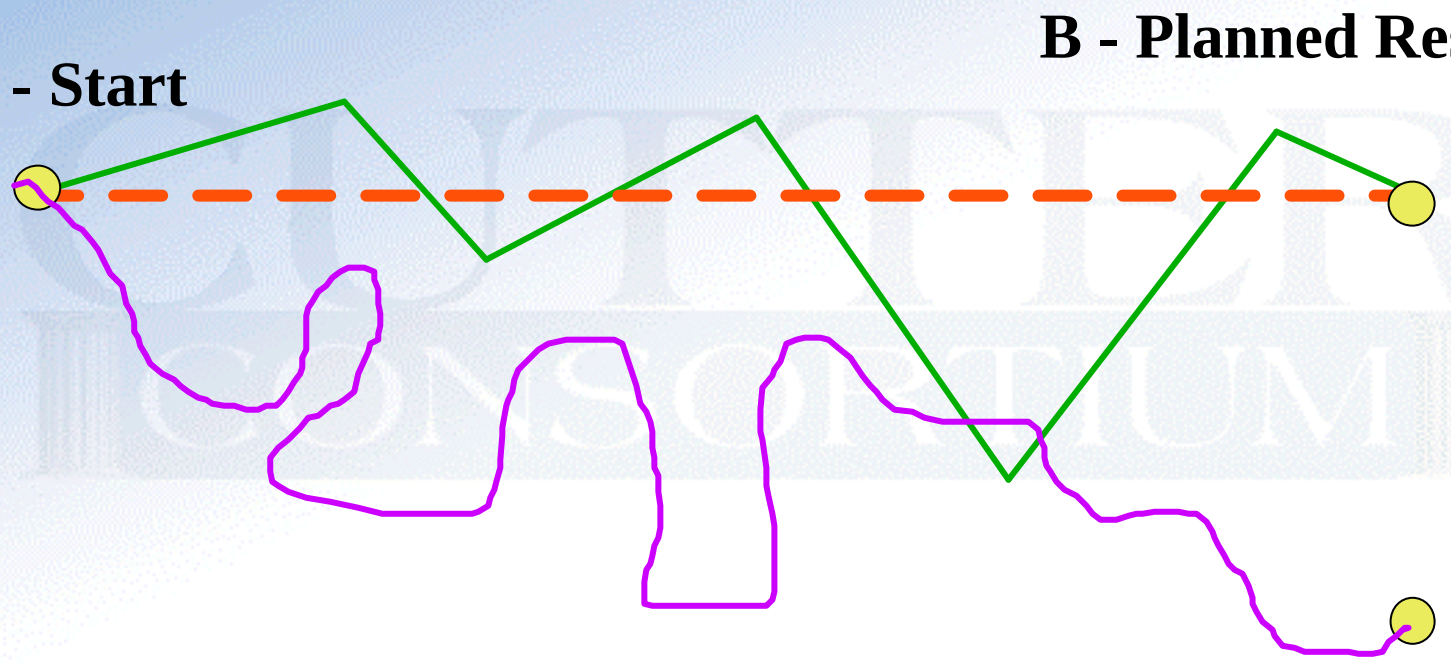


Envision-Explore, not Plan-Do

- Mission-Directed
- Feature-Driven
- Iterative (exploratory)
- Timeboxed
- Risk-Driven
- Change-Tolerant



SPECULATE-Collaborate-Learn



"In an extreme environment, following a plan produces the product you intended, just not the product you need."

People and Collaboration

	Deliverables	Decisions	Knowledge
Interpersonal			
Cultural			
Structural			

"The act of collaboration is an act of shared creation and/or discovery."
Michael Schrage, *No More Teams*

Cultural Collaboration

Command-Control

Leadership-Collaboration

Command Control is too slow:

- **Can't process information fast enough**
- **Can't make decisions fast enough**

Simple Rules - Dee Hock

"Simple, clear purpose and principles give rise to complex, intelligent behavior."

"Complex rules and regulations give rise to simple, stupid behavior."

Agile Methodologies attempt to identify a few key practices (rules) and then let them evolve to meet specific problems through individual and group feedback.

Leadership-Collaboration

- Establish a Vision and Purpose
- Define acceptable boundary conditions
- Encourages innovation, and collaboration
- Share Power (decision making)
  - Leader empowers teams
  - Teams empowers leaders
- Macro-management, not micro-management

An Overview of the Major Agile Methodologies

The New Methodologies

- Paper by Martin Fowler (www.martinfowler.com)
  - Two key attributes of Agile Methodologies
  - Focused on ***adaptation*** rather than prediction
  - Focused on ***people*** rather that process
- Focused on what works in practice, not what should work
- Focused on Key Practices not All Practices
- Agile Methodologies, coming soon--paper by Martin Fowler & Jim Highsmith, above or www.crystalmethdologies.com.

Major Agile Methodologies

- No-Name, or Home-Grown
- Crystal Methods
- Scrum
- DSDM
- Lean Development
- Feature-Driven Development
- Extreme Programming
- Adaptive Software Development

Homegrown

- "So what's new?" Especially within software companies where they are obviously in the Exploratory realm.
- Typical--Trimble Navigation, Christchurch, NZ
  - GPS navigation products, Software Controller Group (50+/-)
  - "Lightweight means no written design documentation."
  - "We tend to have written requirements and specifications but we are currently looking to reduce the level of written material at this level too."
  - "Lightweight means accepting that we cannot identify and control every little task."

"Never do anything that is a waste of time and be prepared to wage long tedious wars over this principle," Michael O'Connor, project manger.

Homegrown

- "Lightweight means that we don't submit much in the way of status reports. We tend to just try things rather than analyze them to death."
- "Lightweight means we spend very little time on estimates."

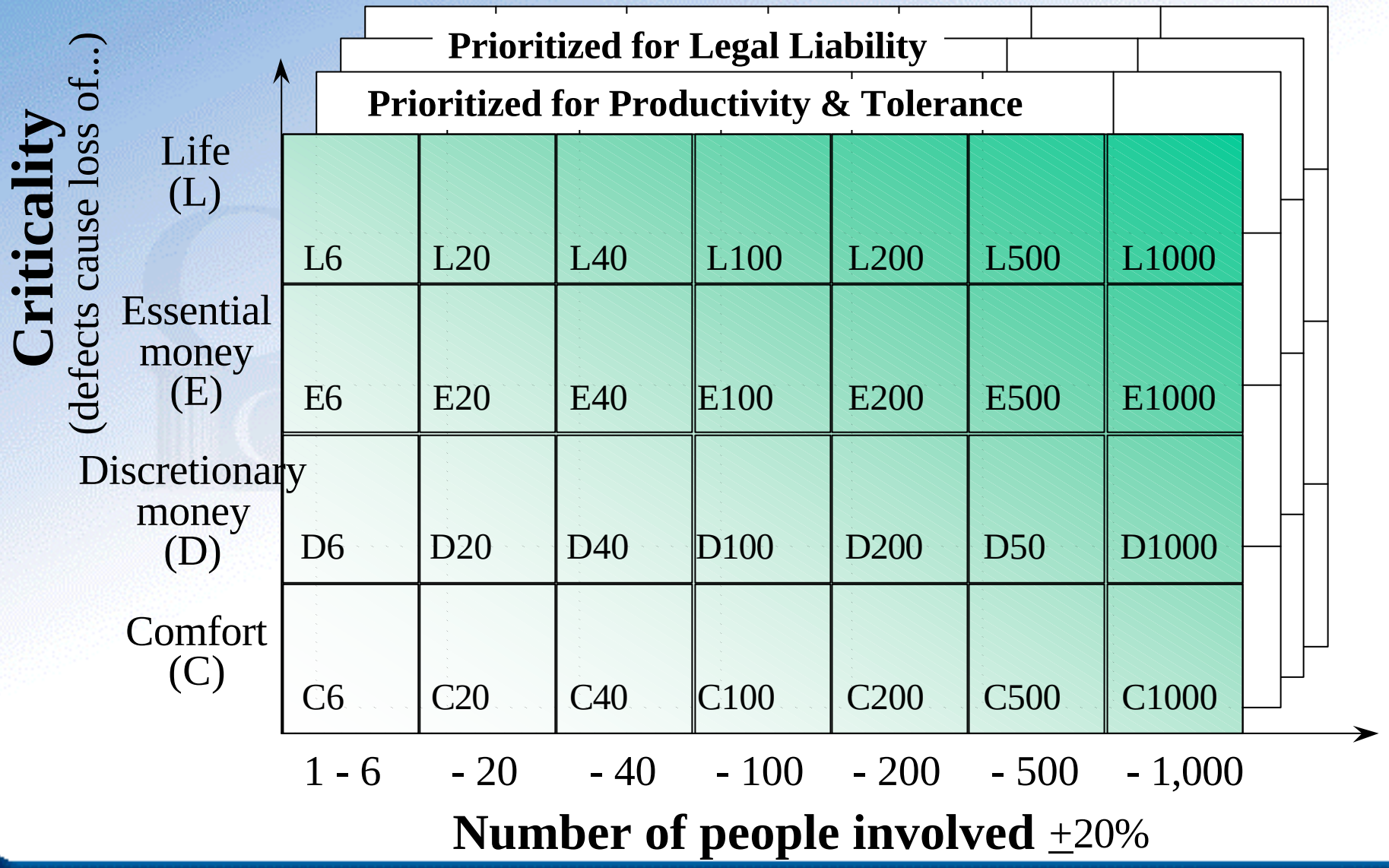
"I expect that there are a zillion little methods from other places that we are using unconsciously. Mostly though we find we have to expend time to fend off other people's ideas about good practices." Michael O'Connor, project manger.

Crystal Methods[1]

- Being developed by Alistair Cockburn.
- Alistair is the methodology archeologist in the group.
- References
  - www.crystalmethodologies.org (Cockburn & Highsmith)
  - members.aol.com/acockburn/ (Cockburn)
  - *Surviving OO Projects*, Addison-Wesley 1998.
  - *Writing Effective Use Cases*, Addison-Wesley 2000.
  - *Software Development as a Cooperative Game*, Addison-Wesley 2002.

[1] **Slides on Crystal courtesy of Alistair Cockburn**

Choosing a Methodology

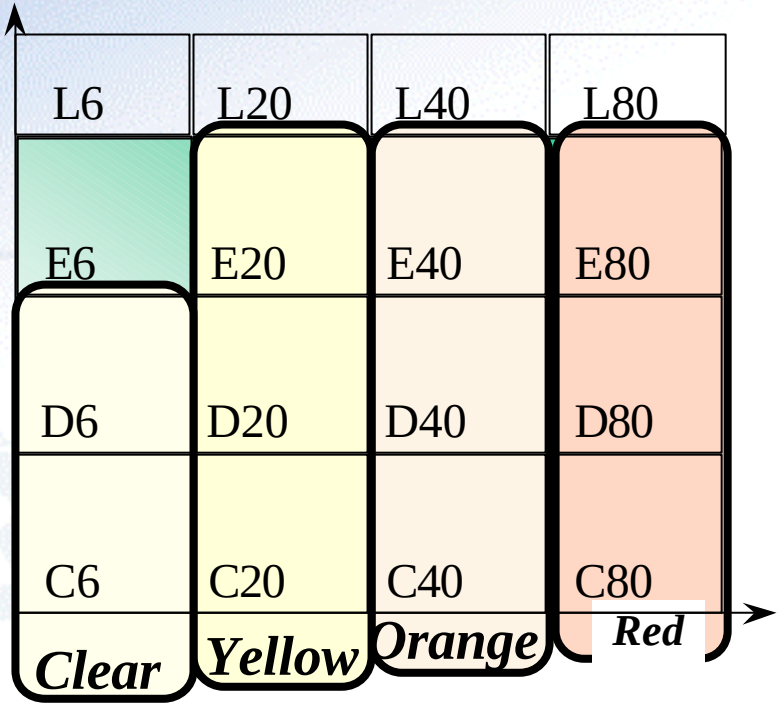


Crystal family of methodologies

***Priorities*:**
High Productivity, High Tolerance

***Common philosophy*:**
**Strong on communications,
Light on deliverables.**

**"Sw dev't is a cooperative game,
using markers that
remind and incite."**



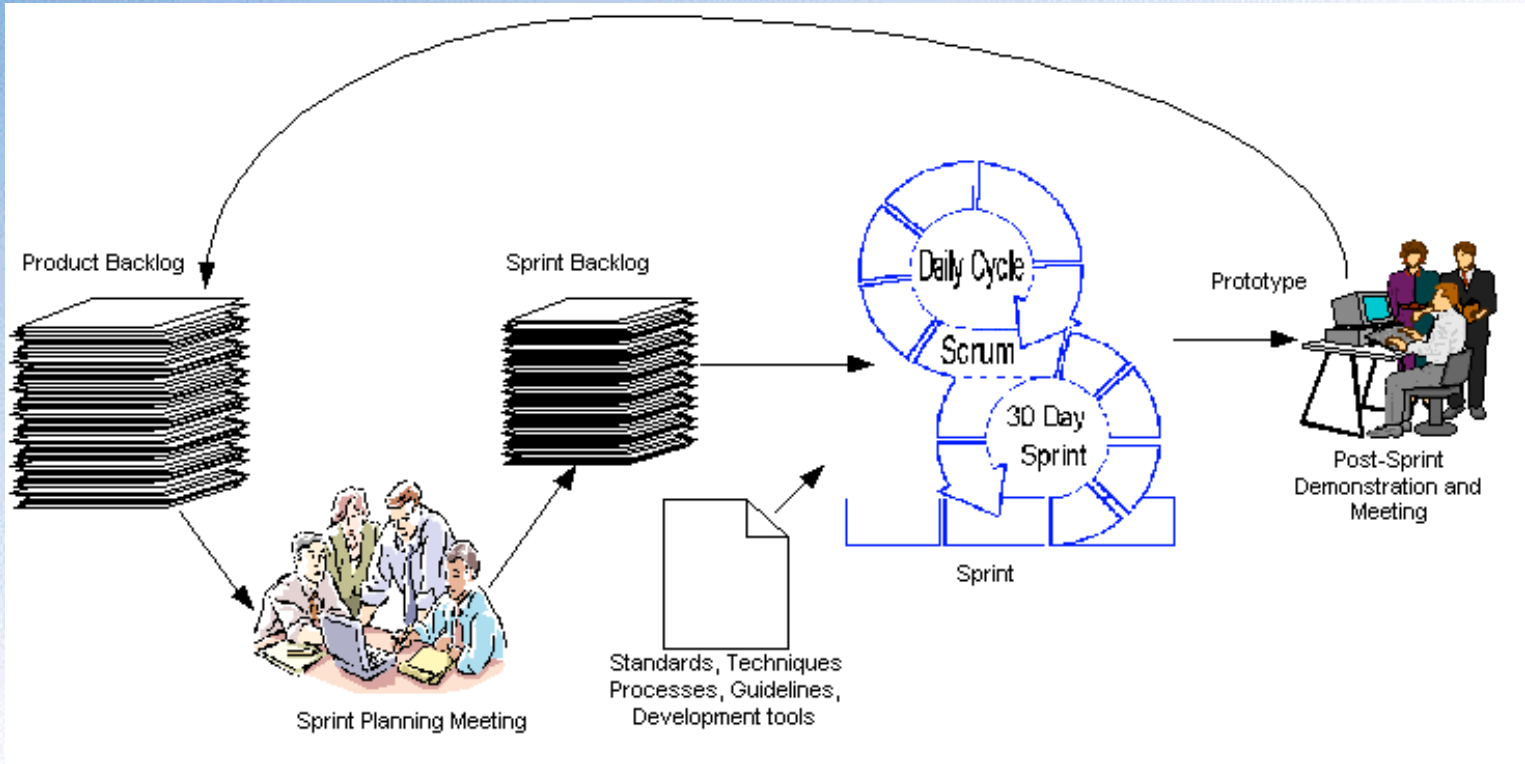
***Key practices*:**
*** Short, rich, informal communications paths**
*** Frequent delivery lowers need for intermediate products.**
*** Use people's natural strengths (talking, looking around)
beware natural weaknesses (careless, low on discipline)**

Scrum[1]

- Developed by Ken Schwaber and Jeff Sutherland
- Differentiate between ***defined*** and ***empirical*** processes
- Name comes from Rugby Scrum
- Grounded in Complexity theory
- Focused on projects:
  - Unpredictable, increasingly complex, business environment,
  - Managing development projects while maximizing their flexibility and ability to deliver best possible products
- References:
  - www.controlchaos.com.
  - http://jeffsutherland.com.
  - Ken is working on a Scrum book

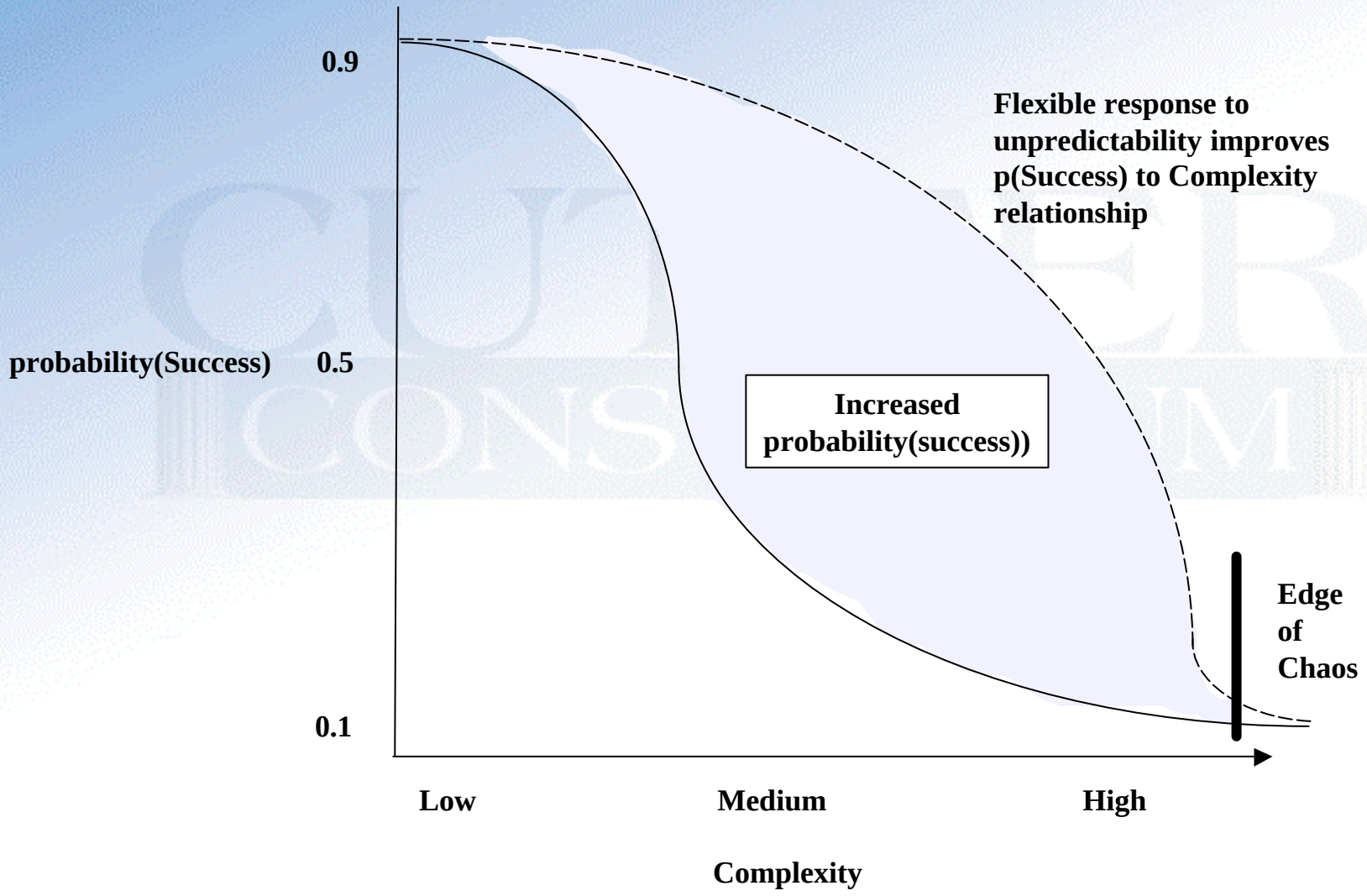
[1] **Slides on Scrum courtesy of Ken Schwaber**

Scrum Process Overview



- Software used for competitive edge
- Pushing the envelope of technical knowledge
- Rapid technical advances
- Increasing difficulty applying latest technologies
- Difficulty planning products in unpredictable environments

Complexity/Success Graph - Scrum



Dynamic Systems Development Method[1]

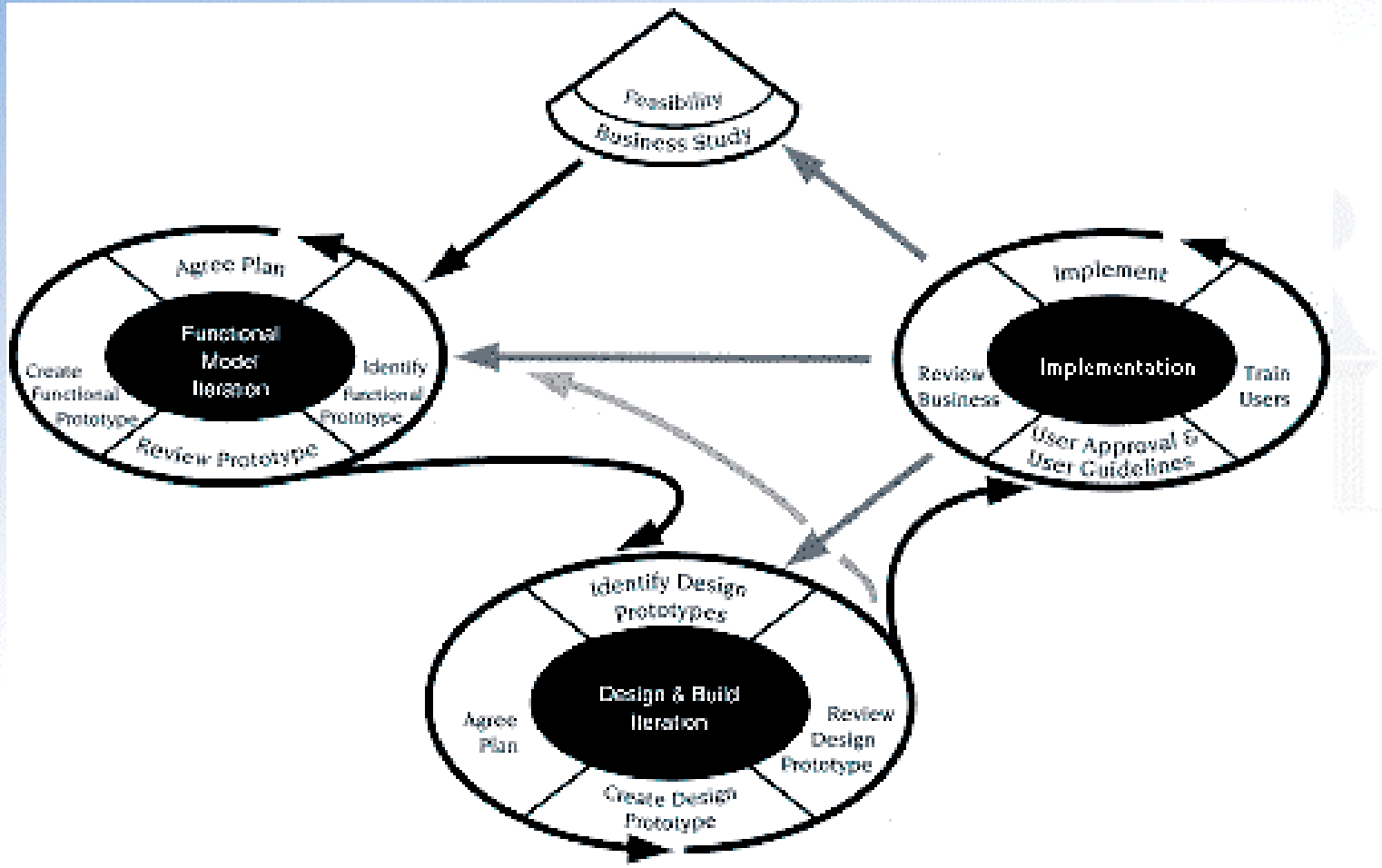
- Developed in the UK in the mid-1990s
- Grew out of RAD approach
- Business study, prototyping, project management
- Managed by the DSDM Consortium
- Expanded within Europe, some presence in the US
- Best supported with documentation & training (at least in Europe)
- References:
  - www.dsdm.org
  - *Dynamic Systems Development Method: The Method in Practice*, Jennifer Stapleton, Addison-Wesley 1997.
  - Dane R. Falkner - DSDM North American Chairman, www.surgeworks.com.

[1] **Slides on DSDM courtesy of NA DSDM Consortium.**

Nine DSDM Principles

1. Active user involvement is imperative.
2. DSDM teams must be empowered to make decisions.
3. The focus is on frequent delivery of products.
4. Fitness for business purpose is the essential criterion for acceptance of deliverables.
5. Iterative and incremental development is necessary to converge on an accurate business solution.
6. All changes during development are reversible.
7. Requirements are baselined at a high level.
8. Testing is integrated throughout the lifecycle.
9. A collaborative and co-operative approach between all stakeholders is essential.

DSDM Development Process

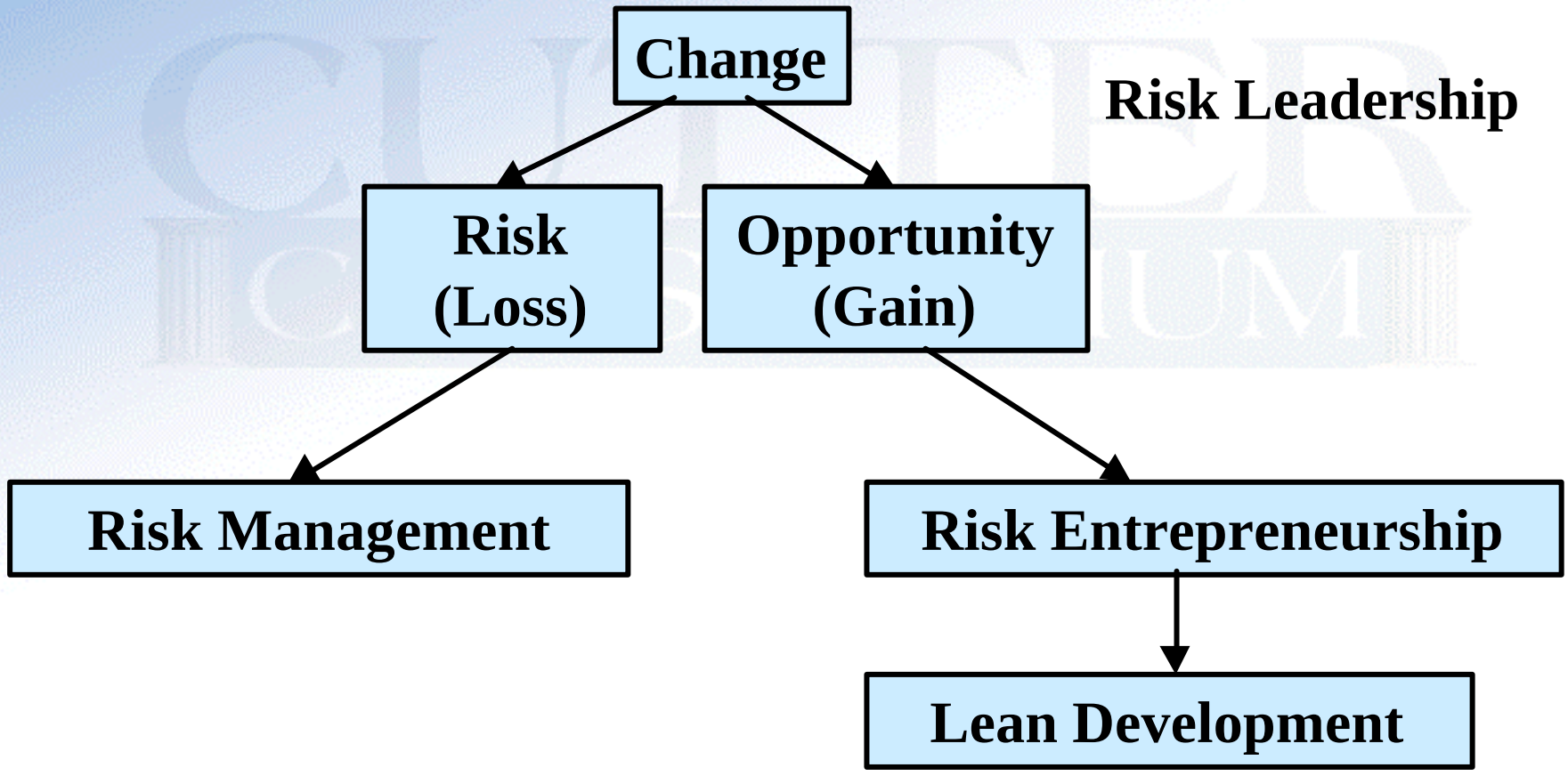


Lean Development

- Developed by Dr. Bob Charette
- Bob is world renown for his work in risk management, and chairs the IEEE committee on Standard for Information Technology - Software Life Cycle Processes - Software Risk Management Process.
- Lean Development is the most strategic oriented of the Agile Methodologies
- References:
  - Bob is producing a CD with video, a methodology manual, and several papers. It will be available from the Cutter Consortium by around June 2001 (www.cutter.com).

Lean Development

Change-Tolerant Organizations



Lean Manufacturing

- Automobile manufacturing
  - Craft Development
  - Mass Production
  - Lean Production
  - Mass Customization (the future)

"The the principles of lean production can be applied equally in every industry across the globe and that the conversion to lean production will have a profound effect on human society."
-- Womack, Jones, & Roos, *The Machine that Changed the World*

Principles of Lean Development

1. Satisfying the customer is the highest priority.
2. Always provide the best value for the money.
3. Success depends on active customer participation.
4. LD is a team effort.
5. Everything is changeable.
6. Domain, not point solutions.
7. Complete, don't construct.
8. An 80% solution today instead of 100% tomorrow.
9. Minimal is essential.
10. Needs determine technology.
11. Product growth is feature growth.
12. Never push LD beyond its limits.

Alcatel (France)

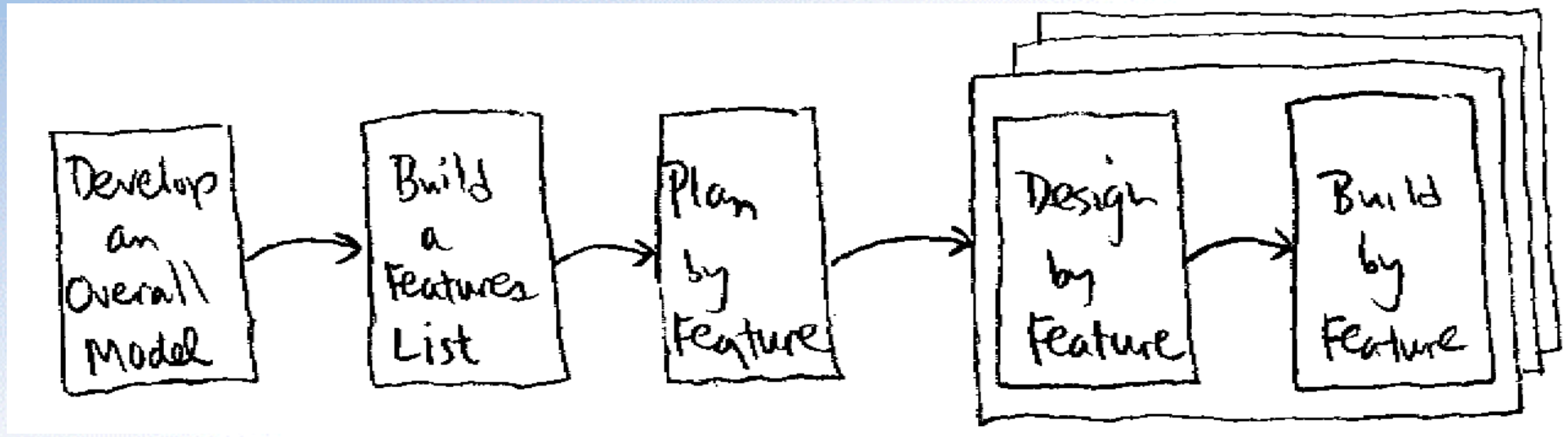
- Goal - 1/3 the time, 1/3 the budget, 1/3 the defects
- Results (6 project to 50 person-yrs)
  - 40% reduction in cost & time
  - Reduced defect levels
  - Improved customer satisfaction

"You have to set the bar high enough to force rethinking traditional practices." -- Dr. Bob Charette

Feature-Driven Development

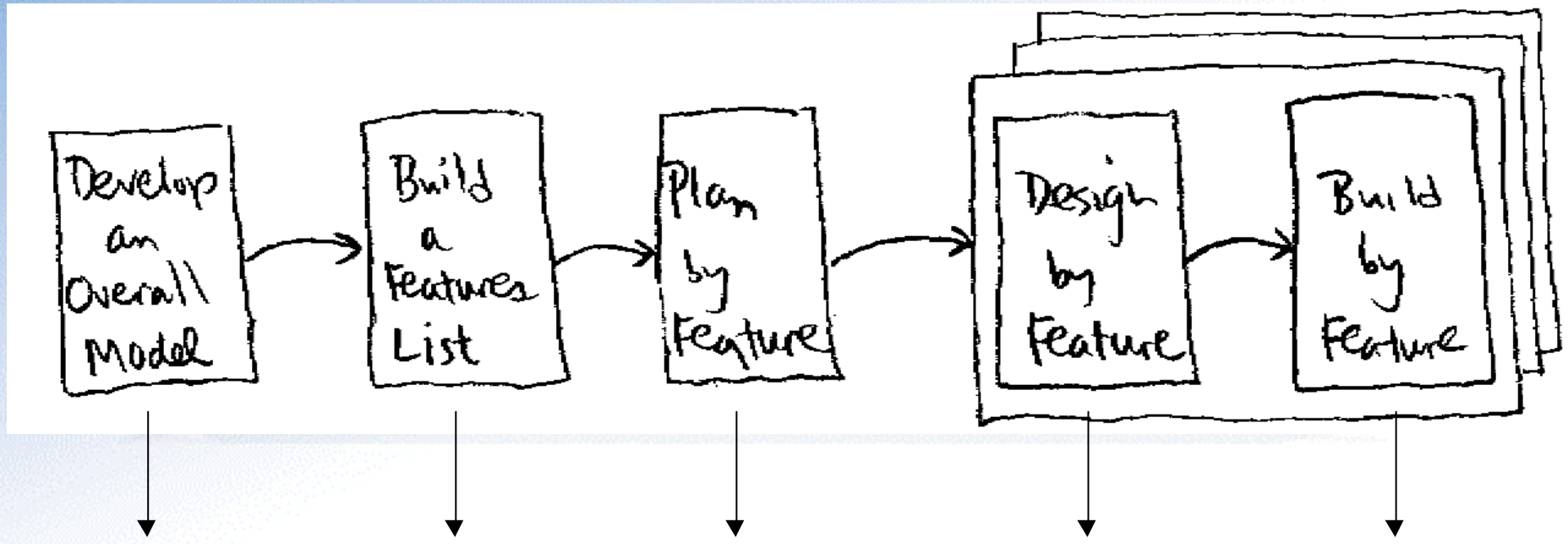
- Developed by Peter Coad
- Discussed in Chapter 6 of *Java Modeling In Color With UML: Enterprise Components and Process* by Peter Coad, Eric Lefebvre, Jeff De Luca (Prentice Hall, 1999).
- Minimalist 5-step process
- Model-driven approach complements TogetherSoft's round-trip software engineering software
- www.togethersoft.com

FDD Overview[1]



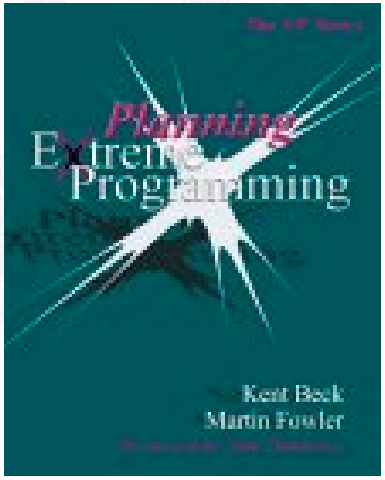
[1] Slides on FDD courtesy of Togethersoft

FDD and Its Five Parts



Extreme Programming

- Best known of the Agile Methodologies
- Developed by 3 extremos--Kent Beck, Ward Cunningham, and Ron Jeffries
- Altered view of cost of change
- A system of practices
- Key new practices
  - refactoring
  - pair programming
  - organic (emergent design)
- References:
  - Extreme Programming: a series of books from Addison-Wesley
  - www.objectmentor.com (XP Training Workshops)
  - www.xprogramming.com (Ron Jeffries)



Extreme Programming Practices

- The Planning Game
- Small releases
- Metaphor
- Simple design
- Refactoring
- Test-First Development
- Pair programming
- Collective ownership
- Continuous integration
- 40-hour week
- On-site customer
- Coding standards

XP Values

Communication

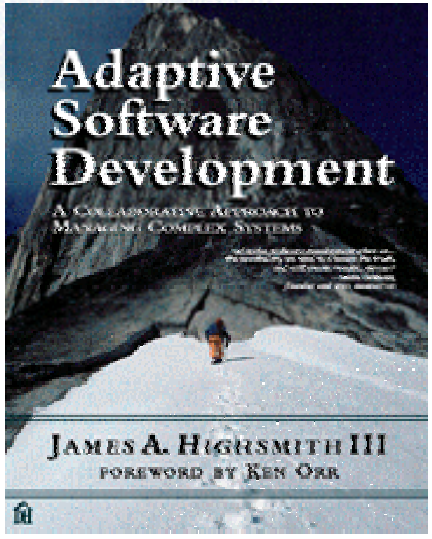
Simplicity

Feedback

Courage

Adaptive Software Development

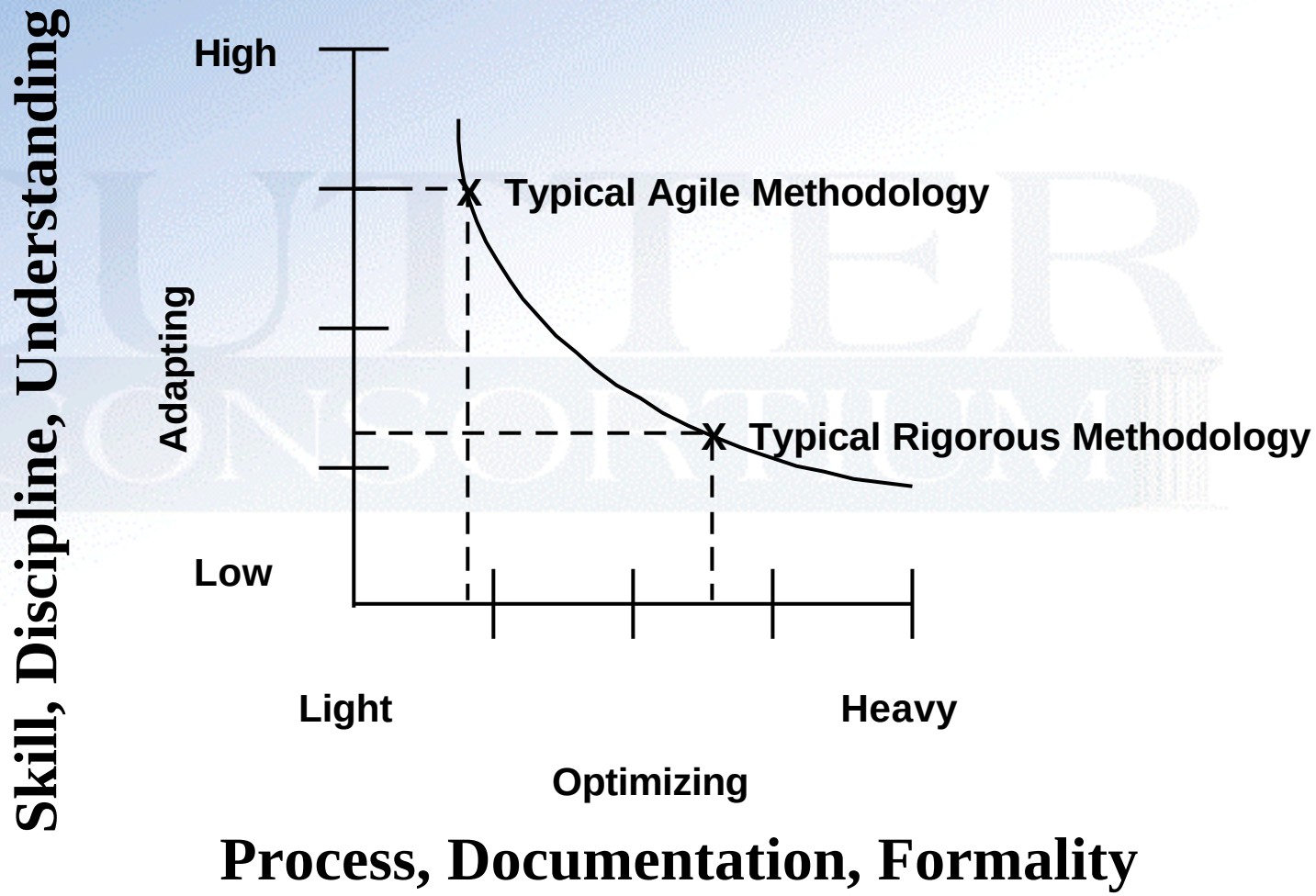
- Developed by Jim Highsmith
- Possibly the philosophical benchmark, according to Martin Fowler
- Grew out of RAD approach in the mid 1990s
- Practices for scaling to larger projects
- Introduces an “Agile” management style called Leadership-Collaboration
- References:
  - *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*, James Highsmith, Dorset House 2000.
  - www.adaptivesd.com
  - www.crystalmethodologies.org



Agile vs. Rigorous Methods

- Documentation is not Understanding (tacit)
  - Typical Requirements Documents: 15% Complete, 7% Correct, Not cost effective to increase (Source: Elemer Magaziner)
- Formality is not Discipline
  - High quality requires discipline
- Process is not Skill
  - High change overwhelms formal process

Modern Agile/Rigorous Debate



Why Agile Methodologies?

Radical Innovation

"Companies fail to create the future not because they fail to predict it but because they fail to imagine it." --Gary Hamel, *Leading the Revolution*

Community

"People, and Relationships, are the new bottom line of business, not simply for humanistic reasons, but as a way to promote adaptability and business success." -- Roger Lewin and Birute Regine, The Soul at Work: Embracing Complexity Science for Business Success