

## **Agile Software Development-- Why it is Hot.**

**Jim Highsmith**

**Director, Agile Project Management Practice,  
& Fellow, Cutter Consortium**

# The Rising Interest in Agile Development

“More than two-thirds of all corporate IT organizations will use some form of an "agile" software development process within 18 months.”

Giga Information Group Inc. predicted at its application development conference in March 2002.

# Why is Agile Hot?

- Agile Software Development (ASD) focuses on innovation.
  - Why is this important?
  - How—emergent versus predictive?
- ASD assists developing software faster and better in highly volatile, demanding situations.
- ASD offers a work environment that appeals to many individuals. (egalitarian, self-disciplined, self-organizing)

**“We Live in a time where creativity, innovation, and imagination drive the world.”**

**--Tom Wujec & Sandra Muscat, *Return on Imagination***

# Innovation Drives Companies

## ■ U.S. Companies, New Product Time to Market

- 1990 avg. = 35.5 months
- 1995 avg. = 23 months
- 2000 avg. = 11 months

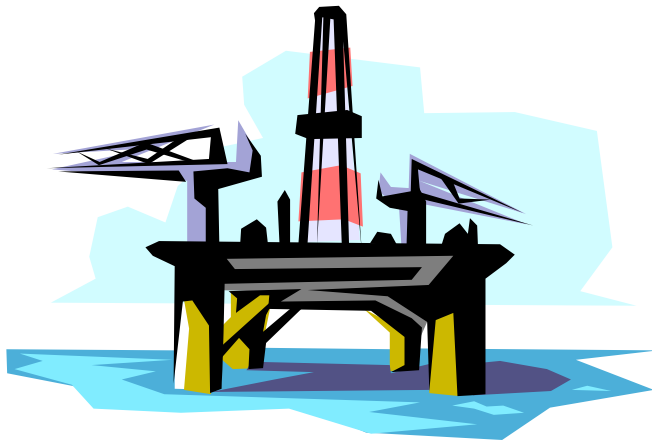
## ■ Cars, Concept to Production

- 1990 avg. = 6 years
- 2001 avg. = < 2 years
- Renault's goal = 9 days!!!

**Source, Tom Wujec & Sandra Muscat, *Return on Imagination***

# Problem Domain: Exploration versus Optimization

## Exploration Drilling



## Production Drilling



# Project Exploration Factor (EF)

	Technology Dimension			
Product Requirements Dimension	Bleeding Edge	Leading Edge	Familiar	Well known
Erratic	10	8	7	7
Fluctuating	8	7	6	5
Routine	7	6	4	3
Stable	7	5	3	1

# Project Portfolio Management

- Identify Exploration Projects
- Manage them Differently
- Measure their Success Differently

# The Diabolical Challenge of Exploration Projects

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

## **Exciting Features**

**Rapid delivery**

**High quality**

**High change**



# Agile Methodologies

Extreme Programming

DSDM

Agile Modeling

Lean Development

Scrum

Feature-Driven Development

Crystal Methods

Adaptive Software Development

# What is Agility?

- Agility is the ability to ***create change*** and ***respond to change***.
- Agility is the ability to balance flexibility and structure.

*“Agility is dynamic, context-specific, aggressively change-embracing , and growth-oriented. It is not about improving efficiency, cutting costs, or battenning down the business hatches to ride out fearsome competitive storms.”*

-- Agile Competitors and Virtual Organizations, by Steven Goldman, Roger Nagel, and Kenneth Preiss, 1995.

# The Agile Manifesto



[www.AgileAlliance.org](http://www.AgileAlliance.org)

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

Through this work we have come to value:

***Individuals and interactions*** over processes and tools

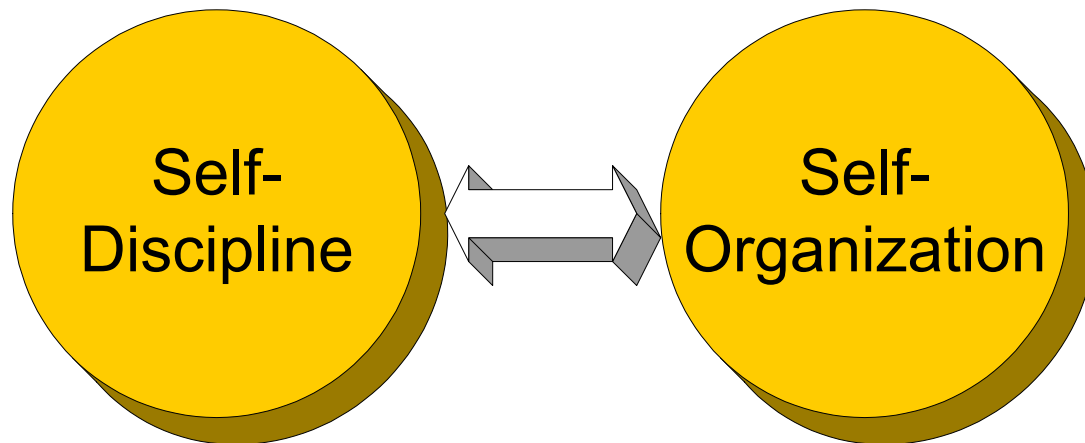
***Working software*** over comprehensive documentation

***Customer collaboration*** over contract negotiation

***Responding to change*** over following a plan

©2001, Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas, **850+ signers**

# Self-organizing Workplace



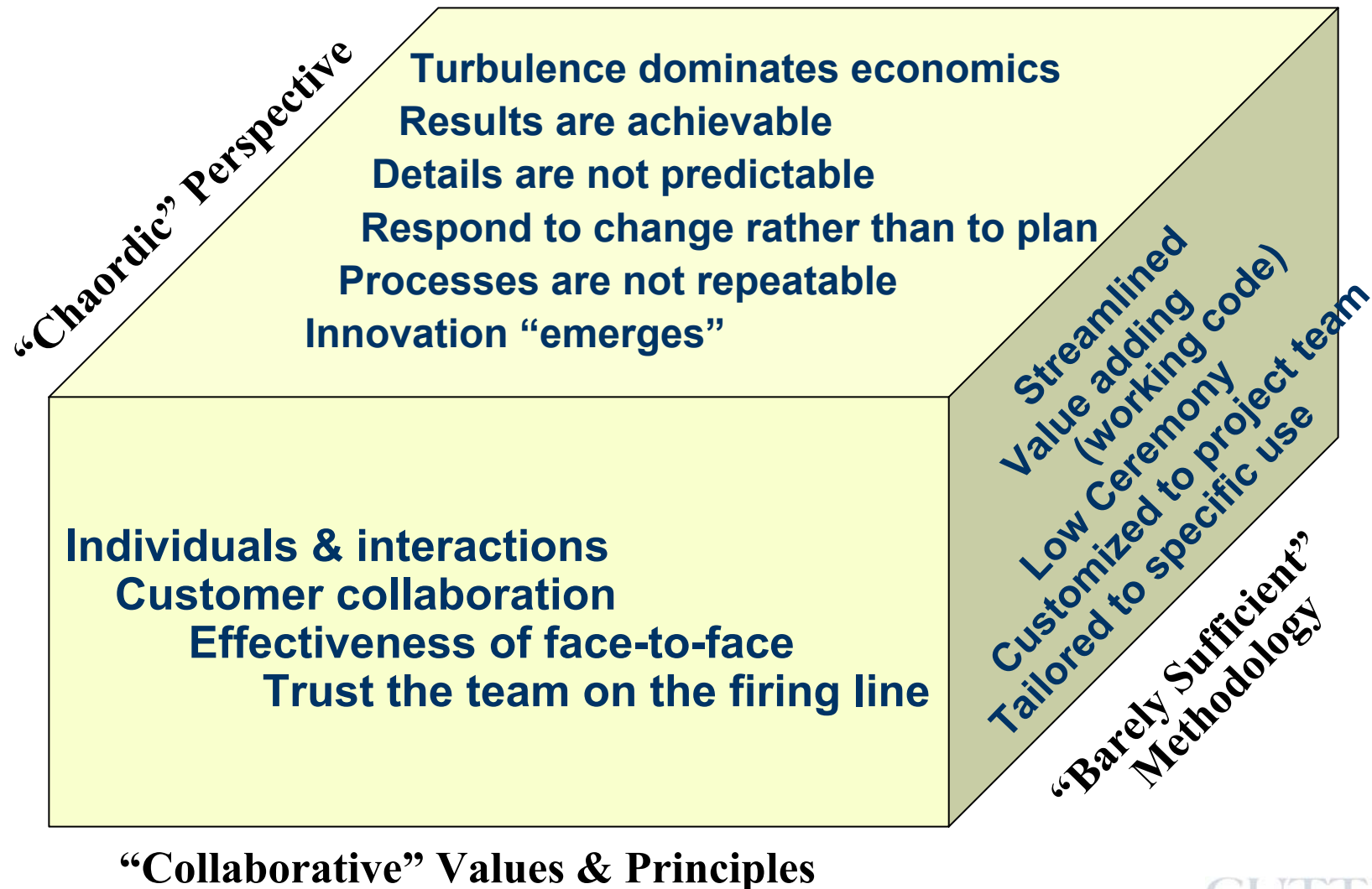
**R-E-S-P-E-C-T**

**TRUST**

# What Constitutes Self-Discipline?

- Accepting of individual accountability for performance results
- Confronting reality through rigorous data collection and analytical thinking
- Engaging in intense interaction, debate, discussions, and decisionmaking
- Willingness to work within the agreed upon self-organizing framework.

# The Dimensions of an Agile Ecosystem



# Agile Principles

- Deliver something useful
- Rely on people
- Encourage collaboration
- Technical excellence
- Do the simplest thing possible
- Be adaptable

# Deliver Something Useful

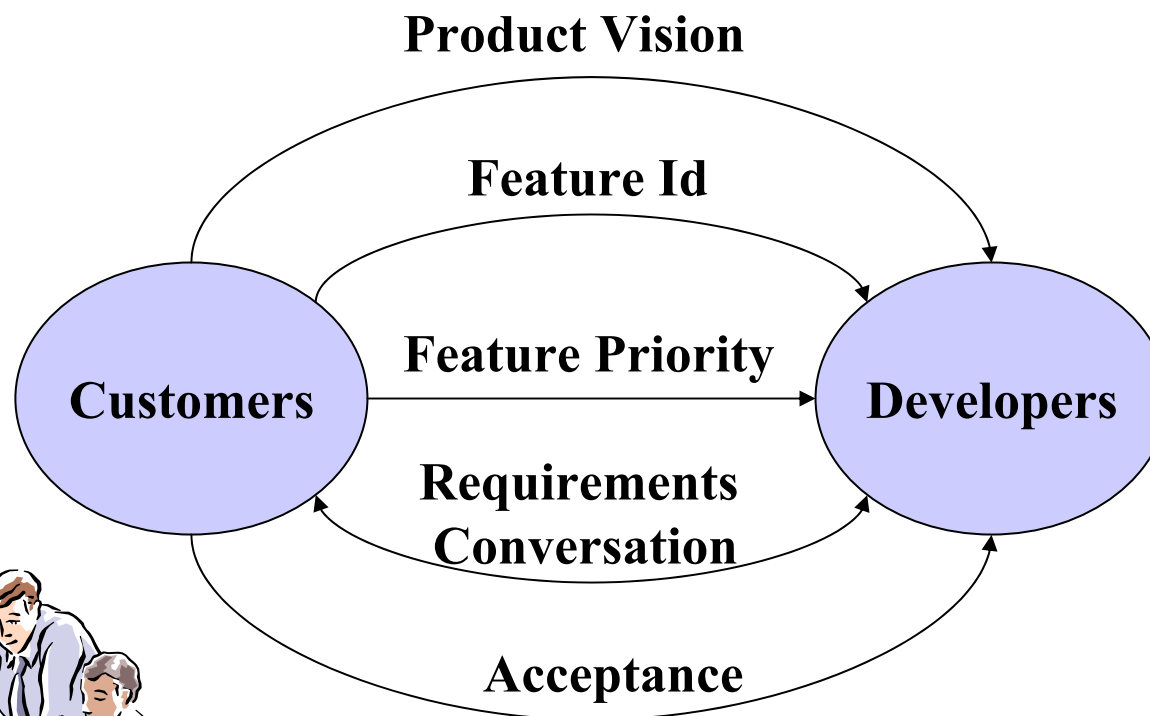
## ■ Agile Manifesto principles:

- Satisfy Customer: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Frequent Delivery: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Working Software: Working software is the primary measure of progress.

## ■ The Customer-Developer Interface (CDI)



# The Customer-Developer Interface



# Rely on People

## ■ Agile Manifesto principles

- **Skilled People:** Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- **Sustainable Development:** Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

## ■ Process versus skill

# Encourage Collaboration

	<b>Deliverables</b>	<b>Decisions</b>	<b>Knowledge</b>
<b>Interpersonal</b>			
<b>Cultural</b>			
<b>Structural</b>			

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

# Technical Excellence

## ■ Agile Manifesto principles

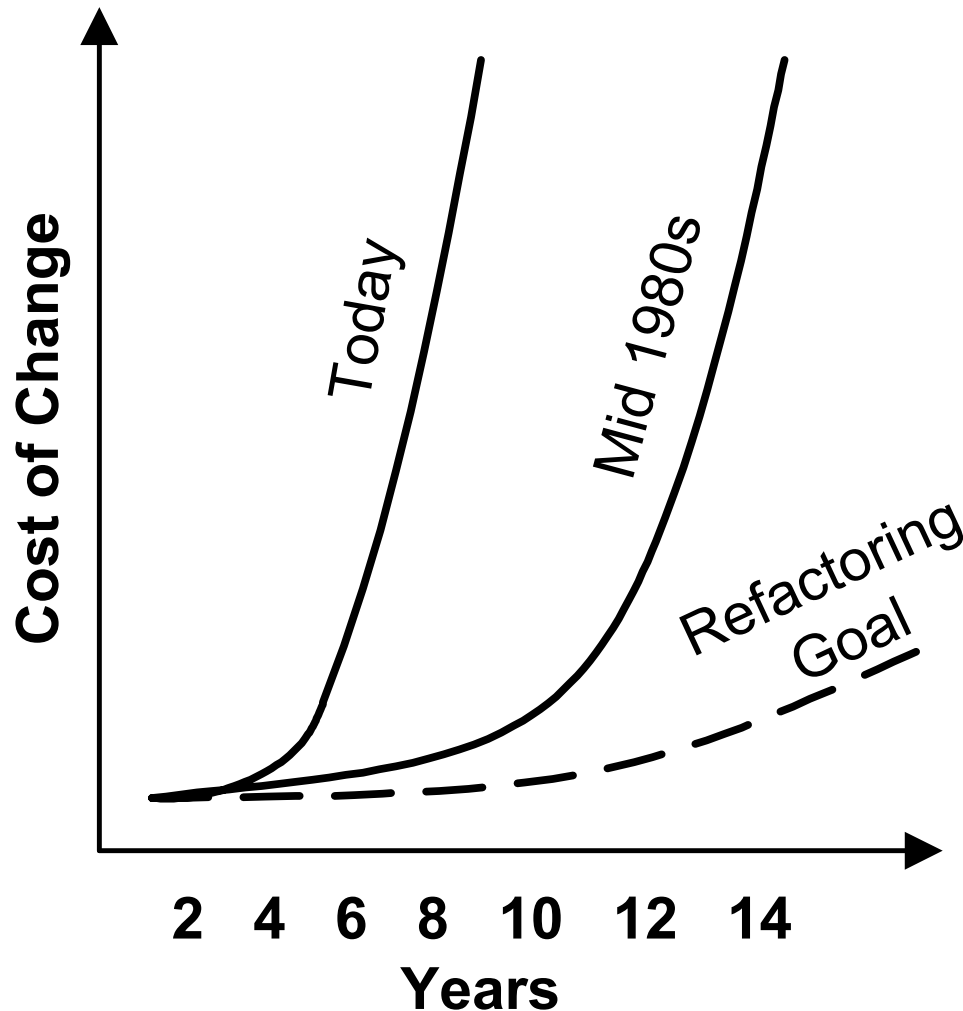
- Technical Excellence: Continuous attention to technical excellence and good design enhances agility.

## ■ Agile is not ad hoc

- Bureaucracy arises from lack of organizational discipline
- Adhocracy arises from lack of self-discipline
- Formality isn't discipline

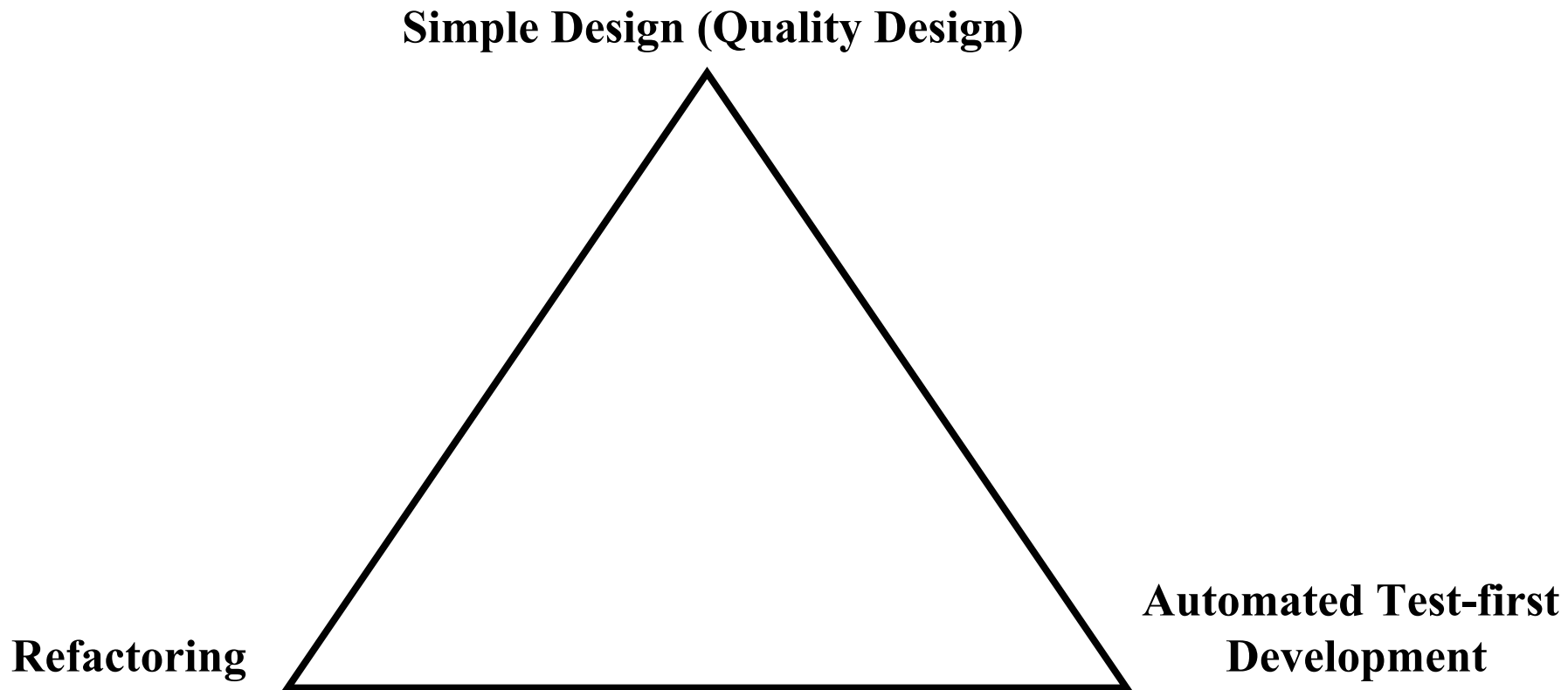
## ■ The XP triangle

# Impact of Change

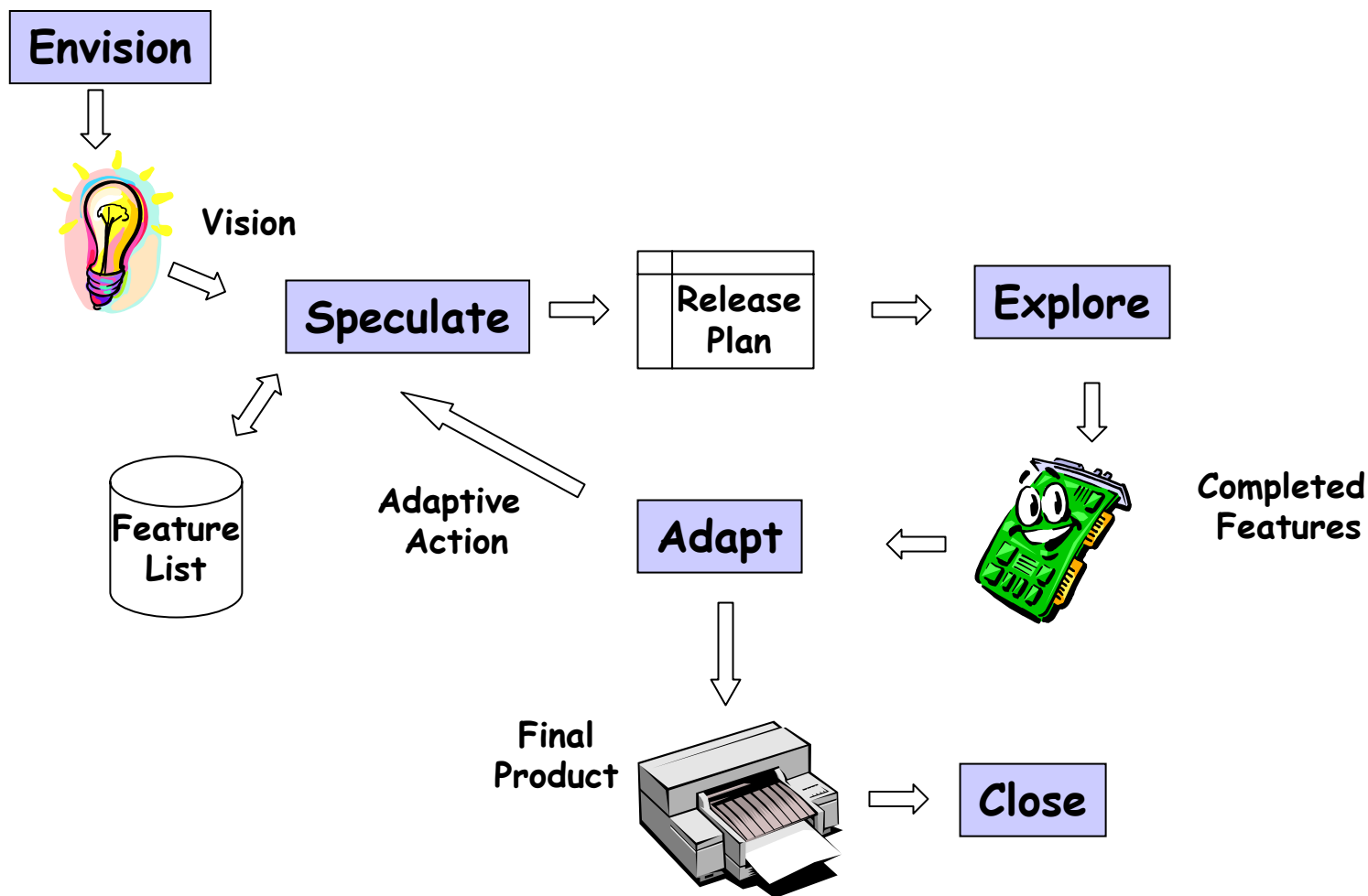


**Rework is a reaction to change, not the result of poor work or a flawed process.**

# The XP Triangle



# The Agile Lifecycle Model



# Simplest Thing Possible

- **Simplicity: Simplicity--the art of maximizing the amount of work not done--is essential.**
  
- **The three faces of simplicity**
  - Minimalism
  - Quality design
  - Generative rules



# Be Adaptable

## ■ Agile Manifesto principles

- Changing Requirements: Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

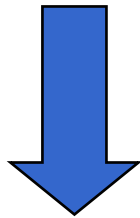
## ■ Predictability or Adaptability

- The more adaptability you want/need, the less predictable the project.
- Understanding this issue is absolutely key to agility.
- Conformance to plan, or conformance to actual?

## ■ Harnessing change

# A Chaordic Perspective

## 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.”

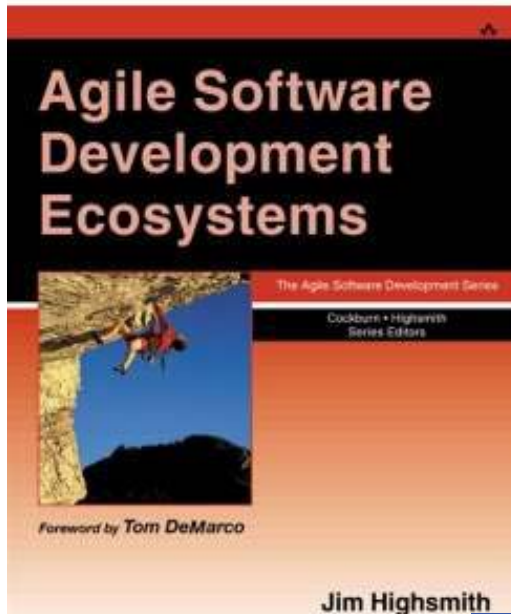
“Complex rules and regulations give rise to simple, stupid behavior.”

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

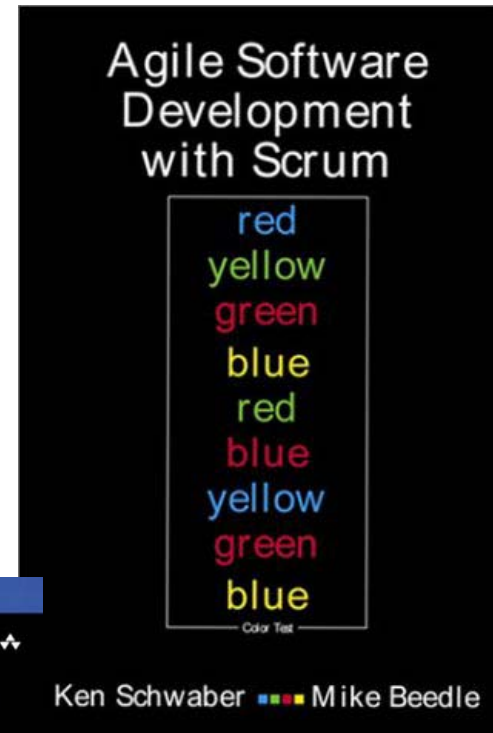
# Leadership-Collaboration

- 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

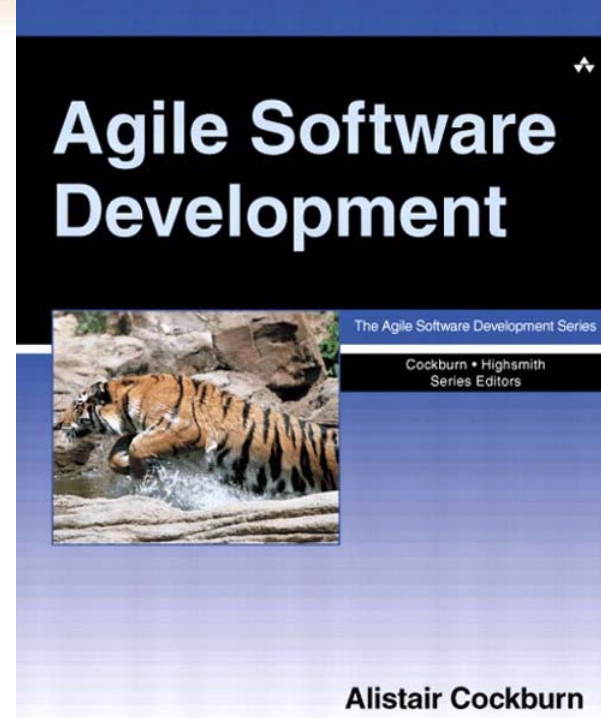
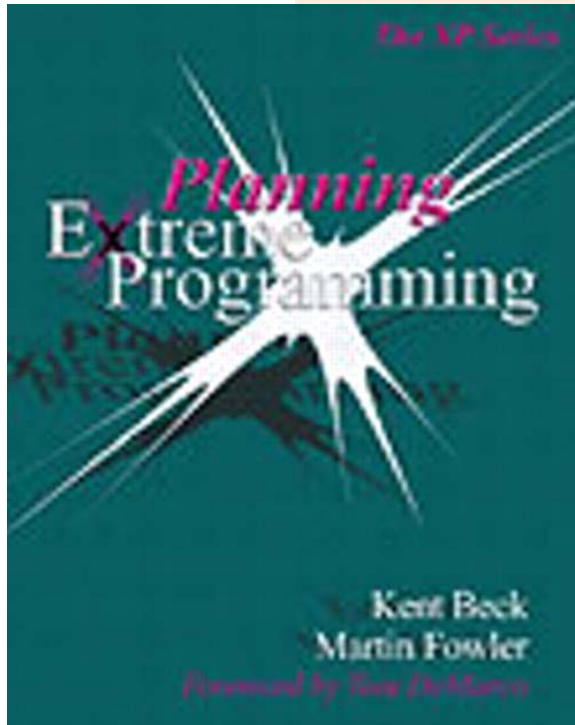




Jim Highsmith



Ken Schwaber ■■■ Mike Beedle



Alistair Cockburn

