



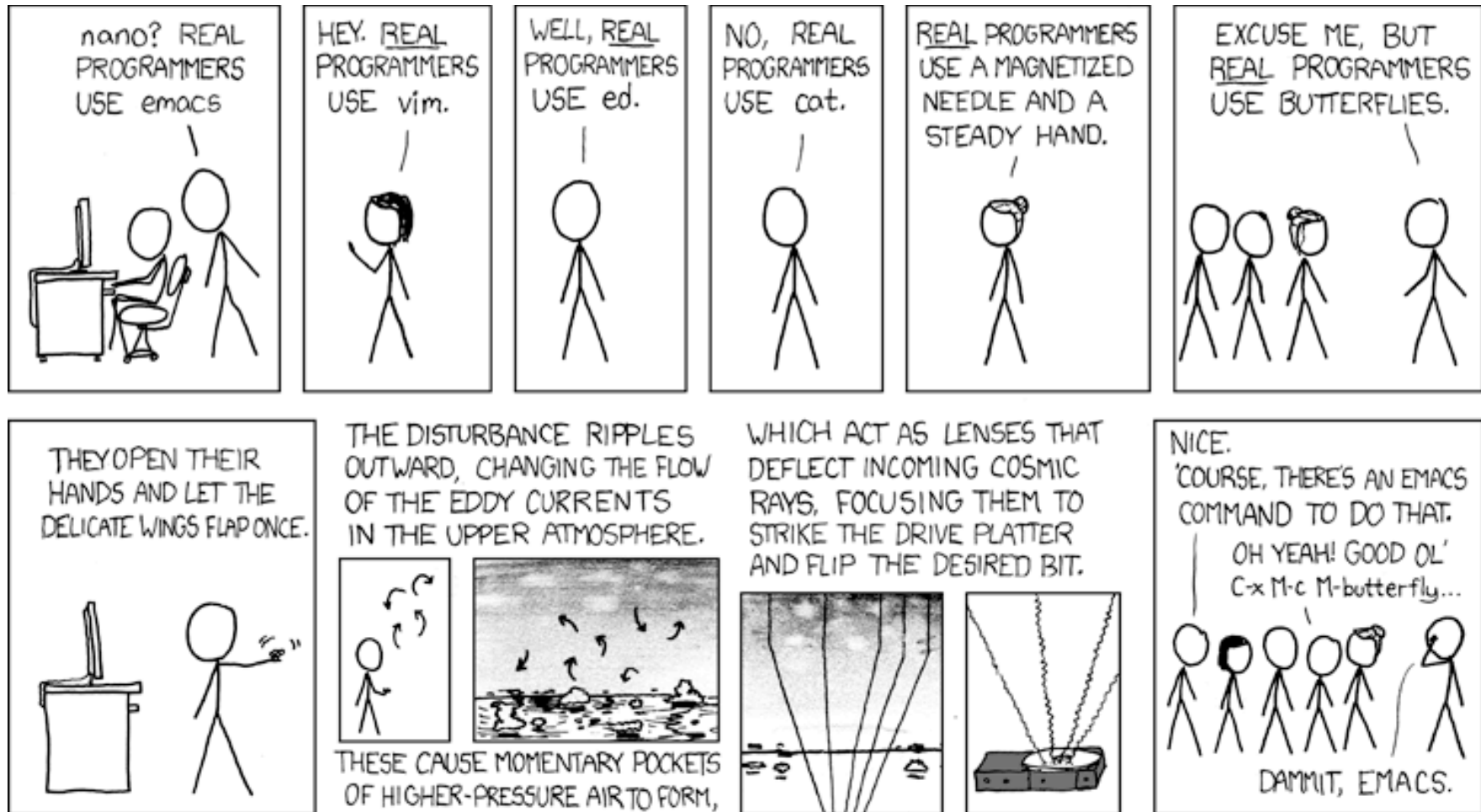
# CSSE 372 Software Project Management: Managing Agile Projects

**Shawn Bohner**  
**Office: Moench Room F212**  
**Phone: (812) 877-8685**  
**Email: [bohner@rose-hulman.edu](mailto:bohner@rose-hulman.edu)**



**ROSE-HULMAN**  
INSTITUTE OF TECHNOLOGY

# XKCD Reference



# Learning Outcomes: Plan (verb)

*Create a plan for an intermediate size software project & manage to the plan as project evolves.*

- **Examine Agile Principles and Practices**
- **Discuss Agile Planning**
- **Outline Agile Project Landscape**
- **Introduce Some Agile Methods**



# What makes Agile Different?

- What would you do different if you knew that your customer could afford only one week of software development?

- Smaller bites, less overhead, less risk, more VALUE...

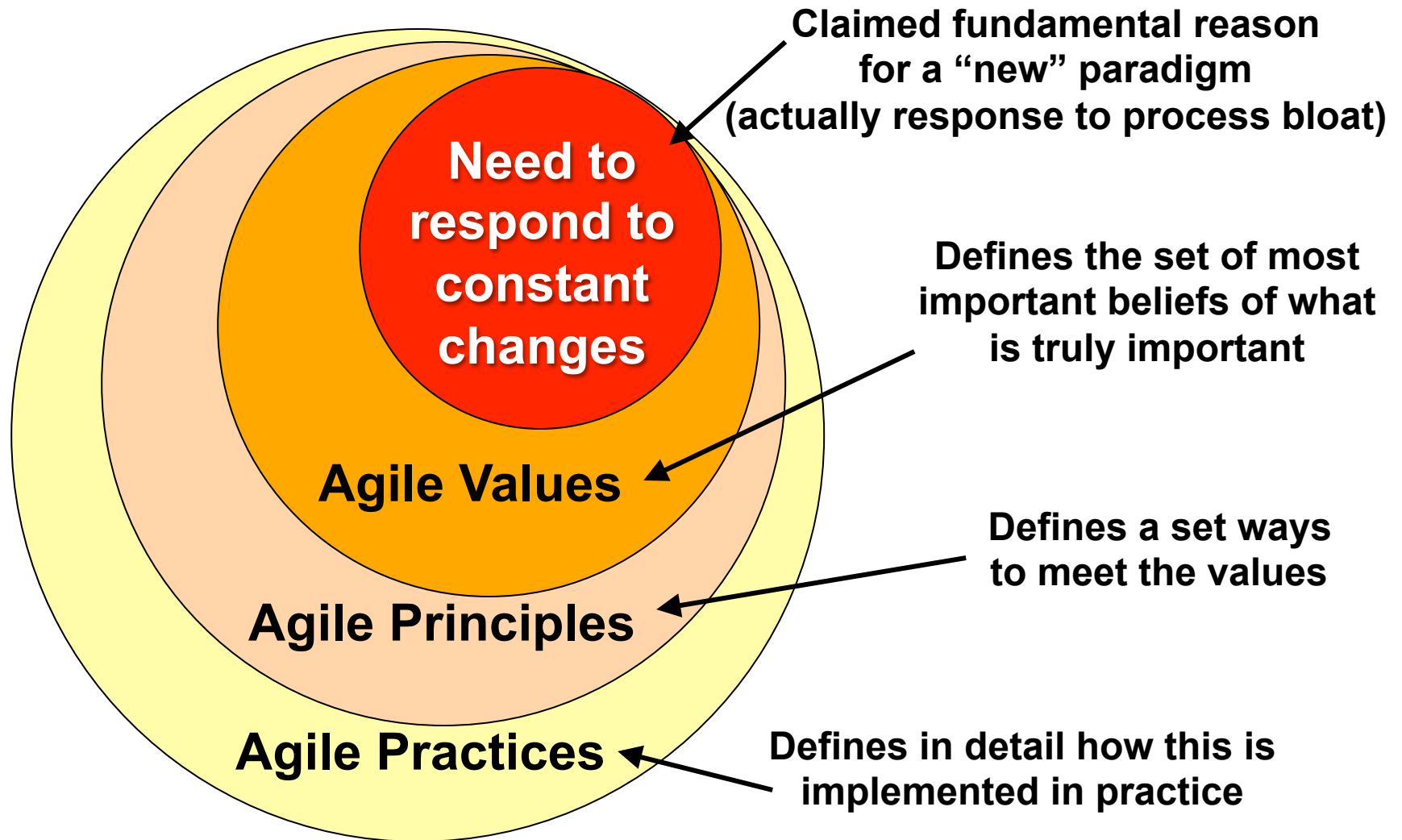


- So, what does the customer value?

- Design Documents? Rigorously defined requirements?
- Nope... they NEED a Working System - SOON!

**How can we get that to them “early and often?”**

# Understanding Agile





# SINGLE RESPONSIBILITY PRINCIPLE

Just Because You Can, Doesn't Mean You Should



# **Agile Principles (1 of 3)**

- **Key priority: satisfy customer through early and continuous delivery of valuable software**
- **Welcome changing requirements, even late in development**
- **Deliver working software frequently, from 2 weeks to 2 months – Prefer shorter timescale**
- **Business people and developers must work together daily throughout the project**



## **Agile Principles (2 of 3)**

- **Build projects around motivated individuals**
  - Give them the environment and support their needs, and trust them to get the job done
- **Most efficient/effective method of conveying information to/within a team is face-to-face**
- **Working software is the primary measure of progress**
- **Continuous attention to technical excellence and good design enhances agility**



## **Agile Principles (3 of 3)**

- **Agile processes promote sustainable development – maintain pace indefinitely**
- **Simplicity, “the art of maximizing the amount of work not done,” is essential**
- **Best architectures, requirements, and designs emerge from self-organizing teams**
- **Team reflects on effectiveness, then tunes and adjusts its behavior accordingly (regularly)**

# Exercise: What are the key elements of Agile Projects?

- Let's do a quick competition...
- Break up into groups
- In two minutes, identify as many elements of Agile Projects as you can and write them on the board



# Agile Planning

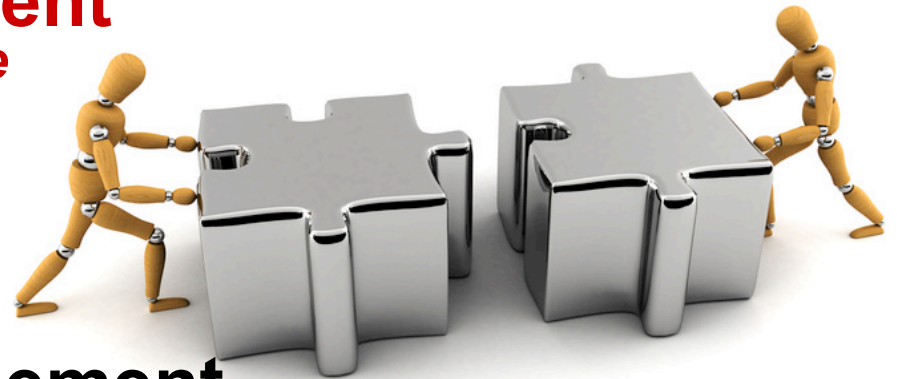
## (Release and Iteration Planning)

Backlog	Release A			
Feature 1	Feature 1, Feature 2, Feature 3a			
Feature 2				
Feature 3				
Feature 4				
Feature 5				
Feature 6				
Feature 7				
Feature ...				
	Release Backlog	Iteration 1	Iteration 2	Iteration 3
	Story A	Story A	Story C	Story F
	Story B	Story B	Story D	Story G
	Story C		Story E	
	Story D			
	Story ...			

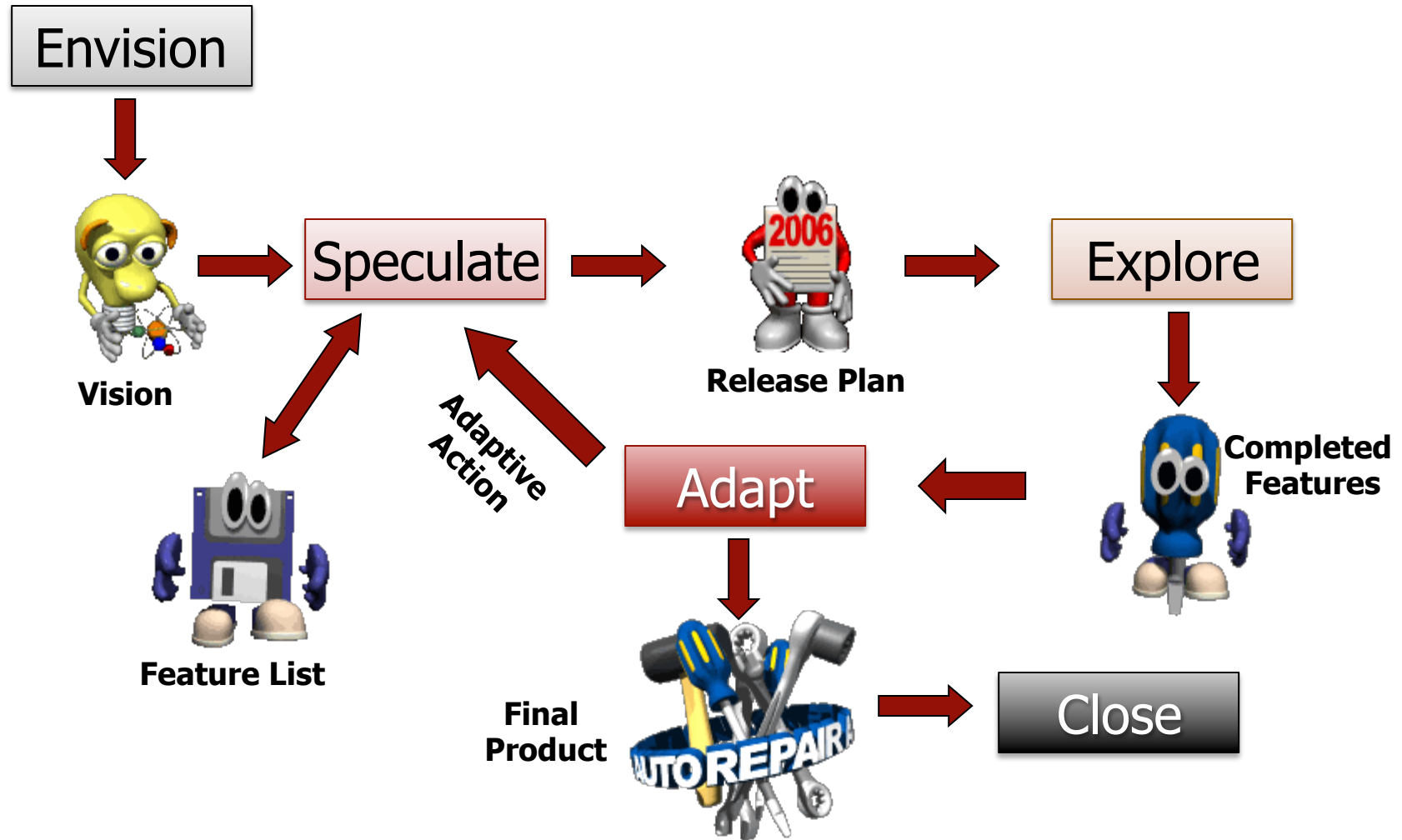
**Always have a plan, but Elaborate at the last “responsible moment”!**

# Agile Landscape: Agile Management Frameworks

- **Agile Project Management**  
*Jim Highsmith, Sanjiv Augustine*
- **Agile Management**  
*David Anderson*
- **eXtreme Project Management**  
*Rob Thomsett, Doug DeCarlo*



# Agile Project Management Framework



# Agile Landscape: Methodologies (1 of 2)

- **Evolutionary Spiral** [Boehm 1988]
- **Rapid Application Development (RAD)**  
[Martin 1991]
- **dX or AUP (Agile UP or Agile Modeling)** [Ambler 2002]
- **Operational Prototyping**  
[Davis 1988]
- **Pragmatic Programming**  
[Martin 2000]
- **Lean Software Development**  
[Charette 1996, Poppendieck 2003]



# Agile Landscape: Methodologies (2 of 2)

- **Adaptive Software Development (ASD)**

[Highsmith 1999]

- **eXtreme Programming**

*Kent Beck, Ward Cunningham,  
Ron Jeffries*

- **Feature Driven Development**

*Jeff DeLuca*

- **Crystal Methods**

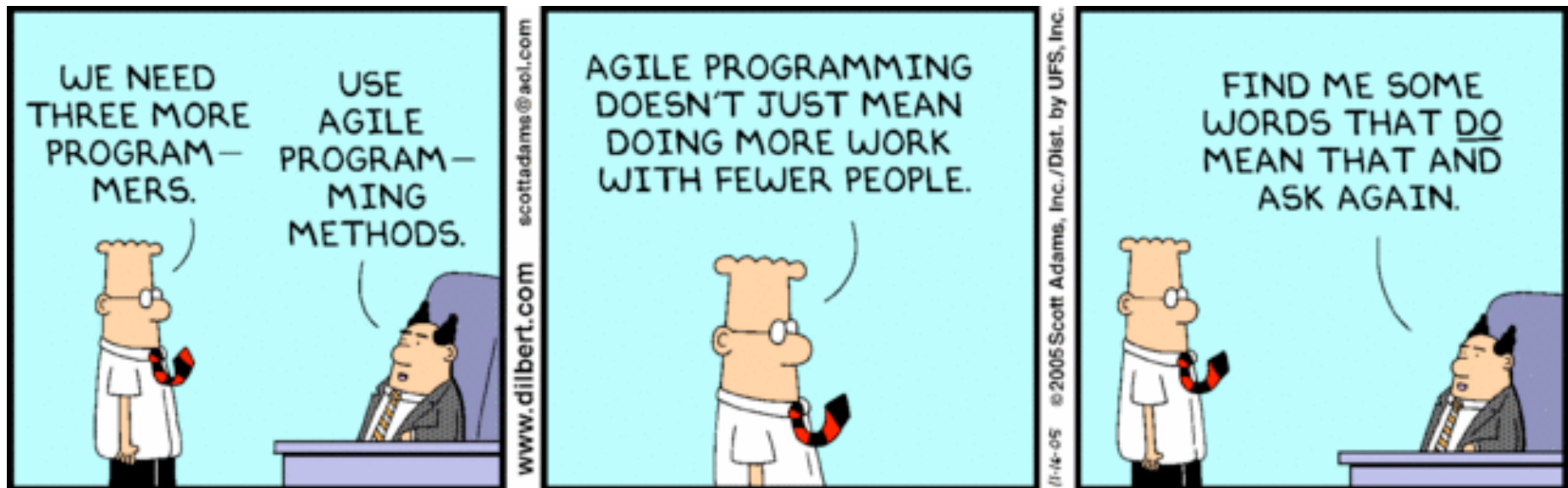
*Alistair Cockburn*

- **Scrum**

*Ken Schwaber and Jeff Sutherland*

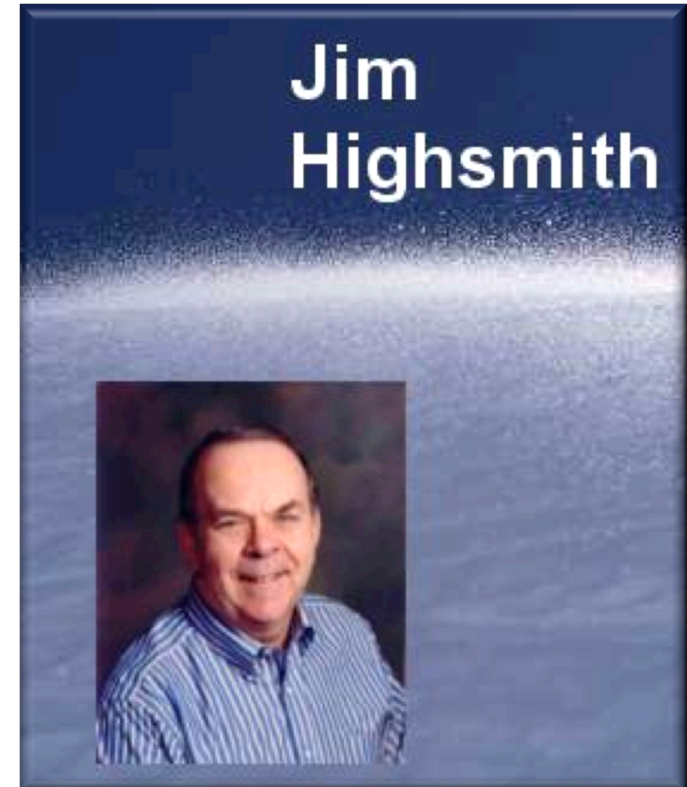


# “Agile Management” according to Dilbert



# Adaptive Software Development (ASD)

- **Mission-driven** planning
- **Component-based**
- Uses “**time-boxing**”
- Explicit **risks** consideration
- Emphasizes **collaboration** for requirements gathering
- Emphasizes “**learning**” throughout the process

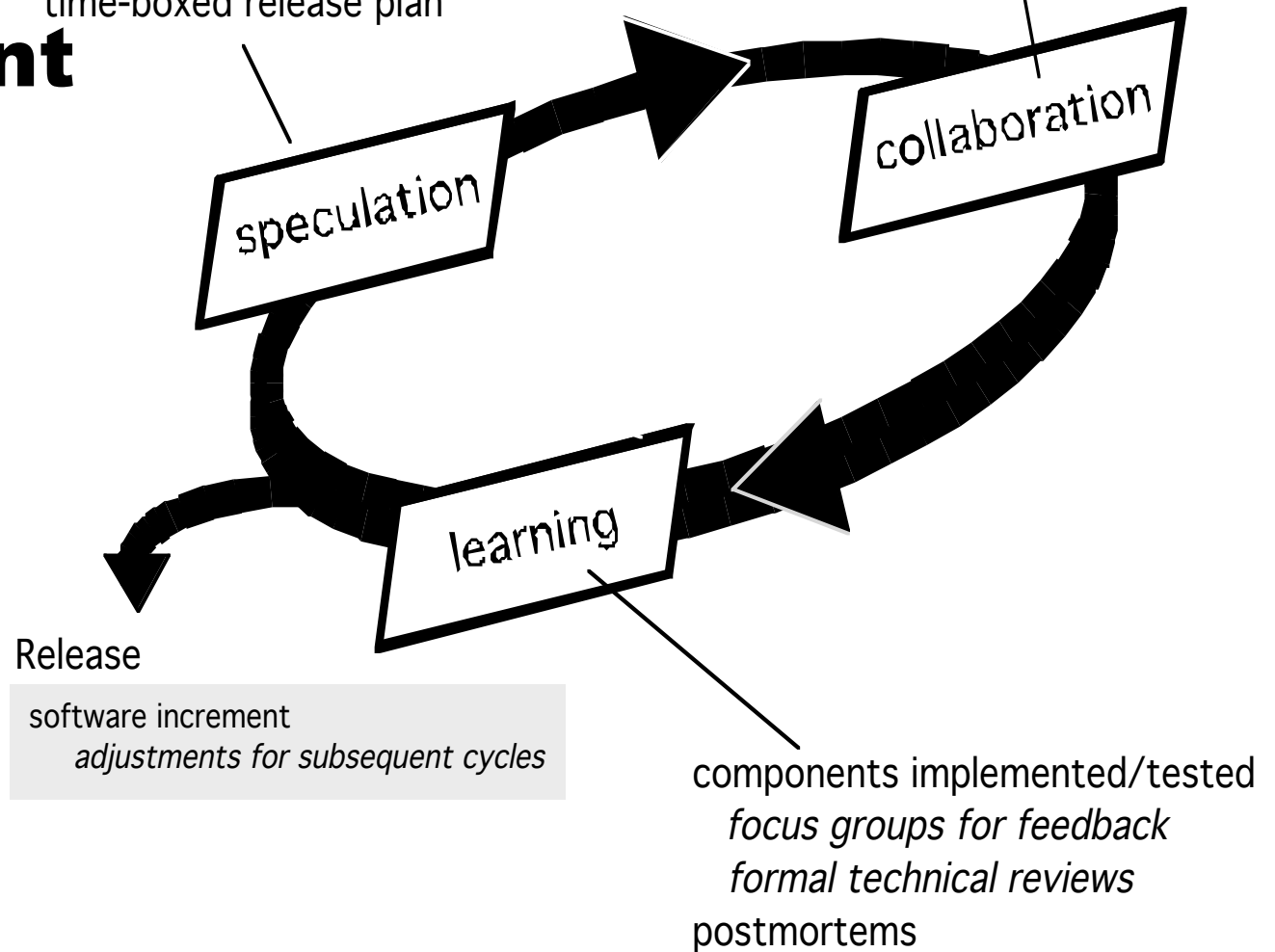




# Adaptive Software Development

adaptive cycle planning  
*uses mission statement*  
*project constraints*  
*basic requirements*  
time-boxed release plan

Requirements gathering  
JAD  
*mini-specs*





# Adaptive Life Cycle Model

*Proceeds from iteration to iteration based on limited specification of solution*

- Each iteration learns from the proceeding ones and redirects the next iteration in an attempt to converge on an acceptable solution
- Client decides if iteration may release a partial solution
- Scope sets boundaries & high-level project parameters
- Plan Phase uses tools, templates, and processes for the coming cycle functionality
- Launch Phase - Same as Iterative Model
  - Establish sub-teams to work on concurrent swim lanes



# Adaptive Strengths and Limitations

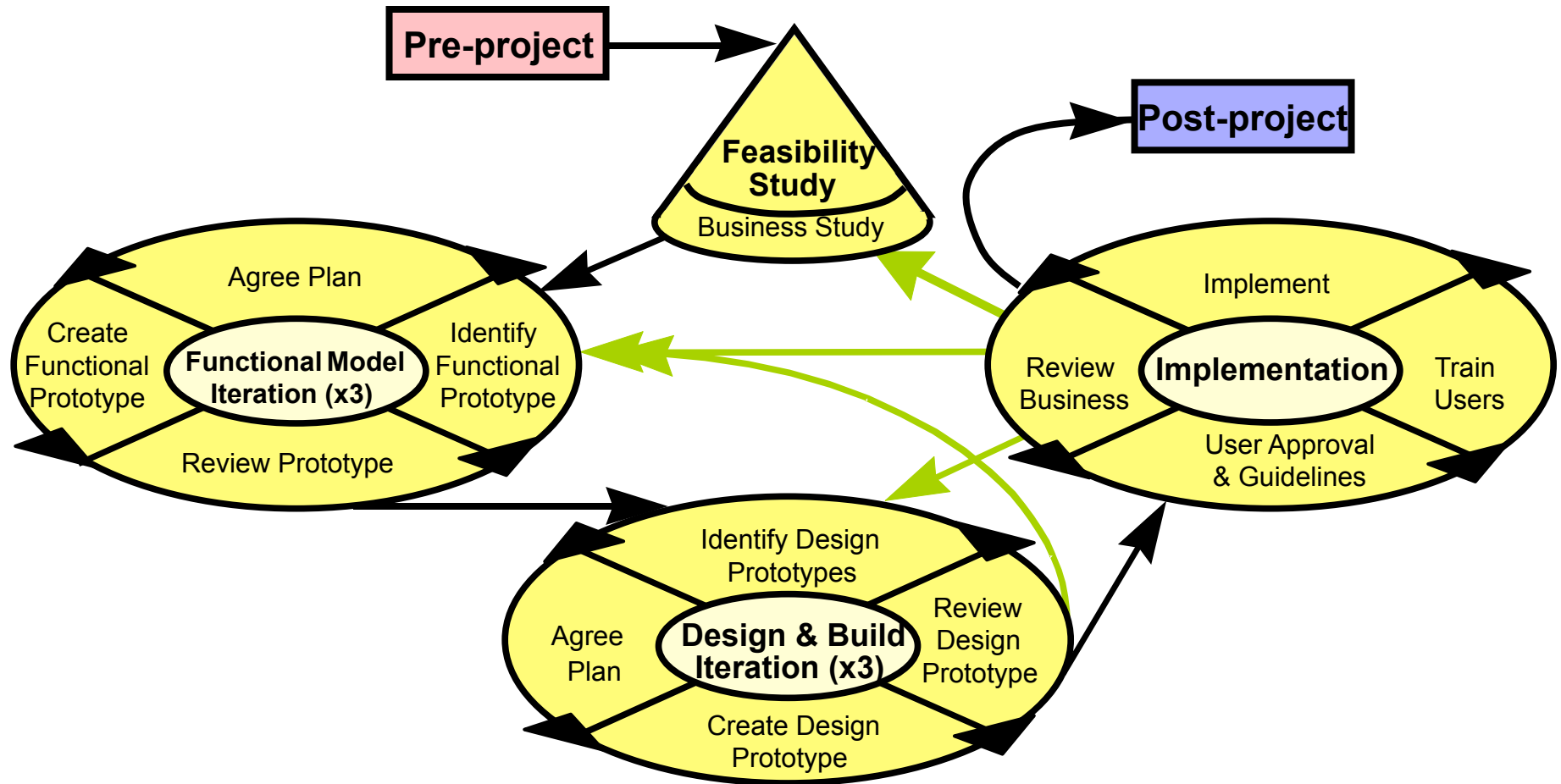
## ■ Strengths

- **Avoids wasted time on non-value added work**
  - Does not waste time planning uncertainty
  - Manages issues processing scope change requests
- Endeavors to maximize project value within time and cost constraints

## ■ Limitations

- Considerable management overhead and involvement
- Must have meaningful client involvement
- **Cannot identify exactly what will be delivered at the end of the project**

# Dynamic Systems Development Method (DSDM)





# **DSDM's Guiding Principles**

- **Active user involvement is imperative**
- **Teams must be empowered to make decisions**
- **The focus is on frequent delivery of products**
- **Fitness for business purpose is the essential criterion for acceptance of deliverables**
- **Iterative/incremental development is needed to converge on an accurate solution**
- **All changes during development are reversible**
- **Requirements are baselined at a high level**
- **Testing is integrated throughout the life-cycle**



# Homework and Reading Reminders

- **Read Agile Book Chapter 4 (referencing back to Chapter 3 for any details)**
  
- **Final Project – SW Proj. Mgt. Plan (SPMP)**
  - **Completed by team...**
  - **Due by 11:55pm, Friday, November 2<sup>nd</sup>, 2012.**
  - **No late days –review swap with another team**