Selecting an Agile Process:
Choosing Among the Leading Alternatives

Mike Cohn
September 21, 2004

My background

The Addison Wesley Signature Series

USER STORIES APPLIED

FOR AGILE SOFTWARE DEVELOPMENT

MIKE COHN
Foreword by Kent Beck

- Programming for 20 years
  - Author of *User Stories Applied*
  - And four programming books
- Past consulting to Viacom, Fidelity Investments, Procter & Gamble, NBC, United Nations, Citibank, other smaller companies
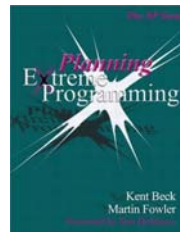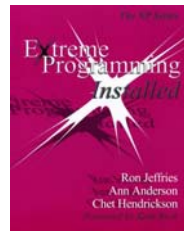- Founding member and director of the Agile Alliance

1

Today's agenda

❑ Extreme Programming

❑ Scrum

❑ DSDM

❑ Crystal Clear

❑ Feature-Driven Development

❑ Overcoming common objections to agile

Extreme Programming (XP)

- The Three Extremos
  - ☐ Kent Beck
 - ☐ Ward Cunningham
  - ☐ Ron Jeffries
- C3 Project

Characteristics

- "Turning all the dials up to 10"
- 1-4 week iterations
- User stories
- On-site customer
- Heavy, heavy emphasis on testing
- Do the simplest thing possible
- You Aren't Gonna Need It (YAGNI)

The Cost of change

- "The error [is] typically 100 times more expensive to correct in the maintenance phase than in the requirements phase."
  - *Software Engineering Economics*, Barry Boehm, 1981, p. 40.
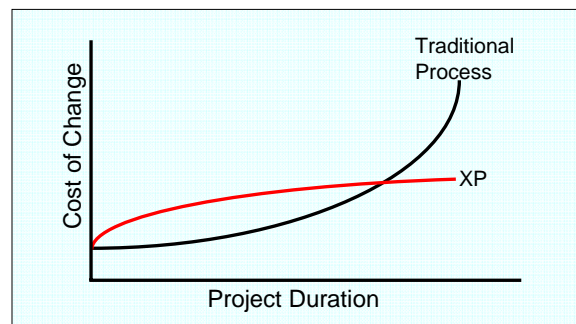
Customer practices

On-site customer

Small releases

Quality practices

Metaphor

Testing

Simple design

Refactoring

Pair programming

Project management practices

The Planning Game

Sustainable pace

Collective ownership

Coding standards

Continuous integration

User stories

- XP's approach to requirements
- Short statement of what's needed
  - Card, Conversation and Confirmation

A user can cancel a reservation.

Users can see photos of the hotels.

A user can make a hotel reservation.

Planning in XP

Business People: Scope, Priority, Release composition, Release dates

The Planning Game

Technical People: Estimates, Consequences, Process, Detailed scheduling

Iteration 1 | Iteration 2 | Iteration 3
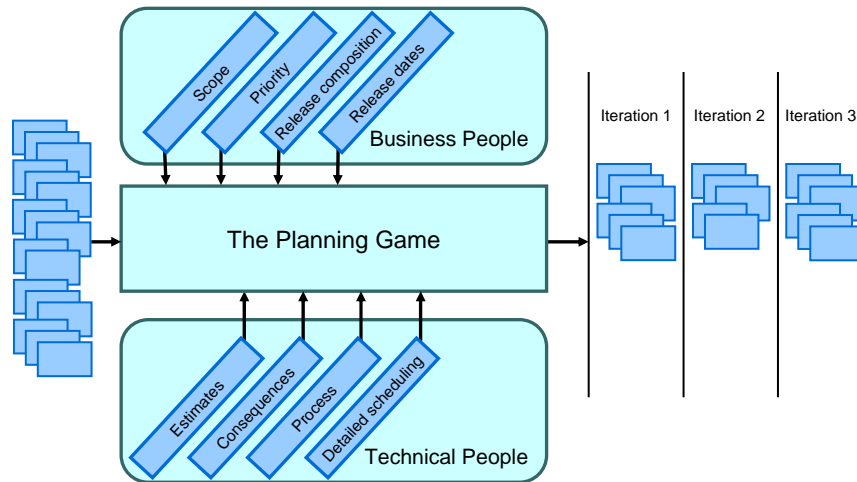
Choose XP if…

- …you have loosely-defined or volatile requirements
- …you have or can develop strong engineering skills and practices
- …customers can be involved on a daily (hourly) basis

Scrum origins

- Ken Schwaber
  - Main advocate
  - Originated with Jeff Sutherland
- 800 Certified ScrumMasters worldwide
- Harvard Business Review, 1986:
  - "The… 'relay race' approach to product development…may conflict with the goals of maximum speed and flexibility. Instead a holistic or 'rugby' approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today's competitive requirements."

Agile Software Development with Scrum

red
yellow
green
blue
red
blue
yellow
green
blue

Ken Schwaber ▪▪▪ Mike Beedle

Microsoft

Agile Project Management with Scrum

Ken Schwaber

Characteristics

- 30-day iterations called "sprints"
- Self-organizing teams
- No specific engineering practices prescribed
  - But many Scrum teams are adopting much of XP
- Uses generative rules to create an agile environment for delivering projects

Scrum roles

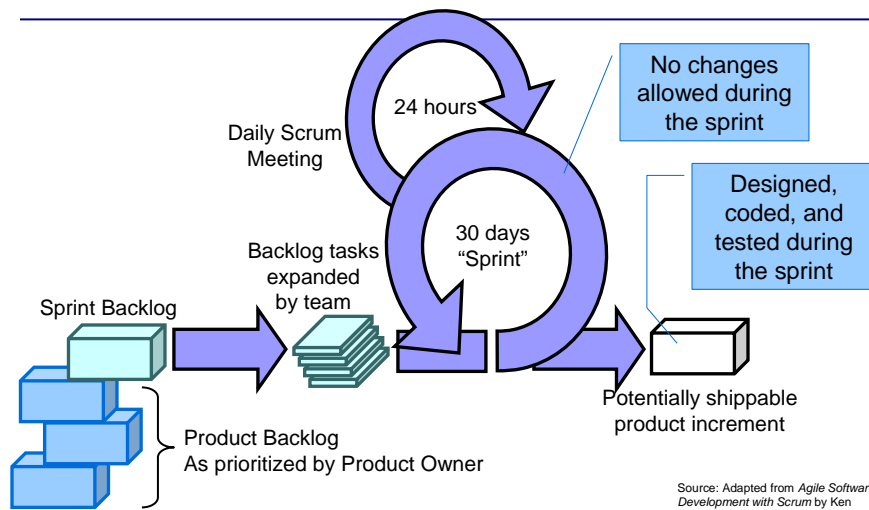| Scrum team | ▪ 5-10 full-time individuals
▪ Self-organizing
▪ Change composition only at end of sprint |

| Product Owner | ▪ Customer or user of the project
▪ Makes prioritization decisions |

| ScrumMaster | ▪ Not quite the project manager
▪ Removes impediments
▪ Enforces Scrum values |

Overview

24 hours

No changes allowed during the sprint

Daily Scrum Meeting

Designed, coded, and tested during the sprint

Backlog tasks expanded by team

30 days "Sprint"

Sprint Backlog

Potentially shippable product increment

Product Backlog
As prioritized by Product Owner

Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

Product backlog

- A list of all desired work on the project
  - Can be stories or engineering tasks
- Prioritized by the Product Owner

	Item #	Description	Est	By
Very High				
	1	Finish database versioning	16	KH
	2	Get rid of unneeded shared Java in database	8	KH
	-	**Add licensing**	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
	-	**Analysis Manager**		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
	-	**Enforce unique names**	-	-
	7	In main application	24	KH
	8	In import	24	AM
	-	**Admin Program**	-	-
	9	Delete users	4	JM
	-	**Analysis Manager**		
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
	-	**Query**	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
	-	**Population Genetics**	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	**Add icons for v1.1 or 2.0**	-	-
	-	**Pedigree Manager**	-	-
	20	Validate Derived kindred	4	KH
Medium				
	-	**Explorer**	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

Sprint Backlog

- Team takes top items from Product Backlog
  - Creates engineering tasks
- Team members sign up
- Only the team can add items during the sprint
- Work is re-estimated as needed

Days Left in Sprint		15	13	10	8	
Who	Description	7/22/2002	7/24/2002	7/26/2002	7/31/2002	
	Total Estimated Hours:	554	458	362	270	0
-	**User's Guide**	-	-	-	-	-
SM	Start on Study Variable chapter first draft	16	16	16	16	
SM	Import chapter first draft	40	24	6	6	
SM	Export chapter first draft	24	24	24	6	
	Misc. Small Bugs					
JM	Fix connection leak	40				
JM	Delete queries	8	8			
JM	Delete analysis	8	8			
TG	Fix tear-off messaging bug	8	8			
JM	View pedigree for kindred column in a result set	2	2	2	2	
AM	Derived kindred validation	8				
	Environment					
TG	Install CVS	16	16			
TBD	Move code into CVS	40	40	40	40	
TBD	Move to JDK 1.4	8	8	8	8	
	Database					
KH	Killing Oracle sessions	8	8	8	8	
KH	Finish 2.206 database patch	8	2			
KH	Make a 2.207 database patch	8	8	8	8	
KH	Figure out why 461 indexes are created	4				

9

Scrum meetings

Sprint planning

- First half includes Product Owner who explains high priority needs
- Second half is team only
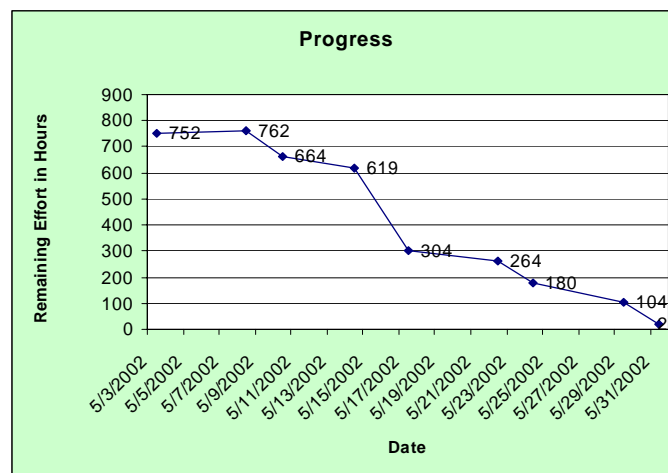- Create sprint backlog

Daily scrum

- What did you do yesterday?
- What will you do today?
- What's in your way?

Sprint review

- Team demos what was accomplished
- 2-hour prep time rule
- No PowerPoint!
- Anyone can attend

Sprint Burndown Chart

Progress

Remaining Effort in Hours vs Date

- 752
- 762
- 664
- 619
- 304
- 264
- 180
- 104
- 0

Dates: 5/3/2002, 5/5/2002, 5/7/2002, 5/9/2002, 5/11/2002, 5/13/2002, 5/15/2002, 5/17/2002, 5/19/2002, 5/21/2002, 5/23/2002, 5/25/2002, 5/27/2002, 5/29/2002, 5/31/2002

Scaling Scrum

- Jeff Sutherland used Scrum at IDX with > 600 developers
- I've used it with > 100
- Scale with "MetaScrums" or "Scrum of Scrums"

Choose Scrum if…

Far from
Agreement

Anarchy

Requirements

Complex

Complicated

Close to
Agreement

Simple

Close to
Certainty

Technology

Far from
Certainty

Source: *Strategic Management and Organizational Dynamics* by Ralph Stacey in *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

- …requirements are changing or emergent
- …you're willing to let the team self-organize
- …you need a management framework more than a set of engineering practices
- …you need a proven, scalable agile process

DSDM

Dynamic
Systems
Development
Method
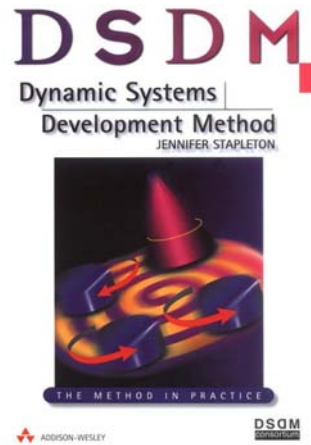
Characteristics

- Highly iterative
- Strong emphasis on prototyping
- Uses timeboxes to control scope
- Strong focus on business value

Principles

1. Active user involvement is imperative.

2. 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. All changes during development are reversible.

Principles, continued

6. Iterative and incremental development is necessary to converge on an accurate business solution.

7. Requirements are baselined at a high level.

8. Testing is integrated throughout the lifecycle.

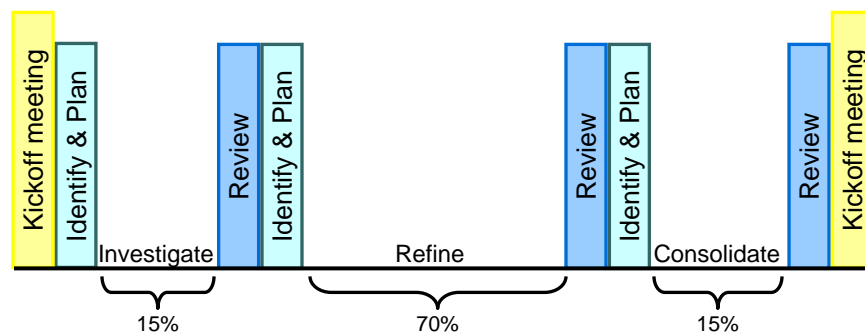9. A collaborative and cooperative approach between all stakeholders is essential.

Three pizzas and a cheese
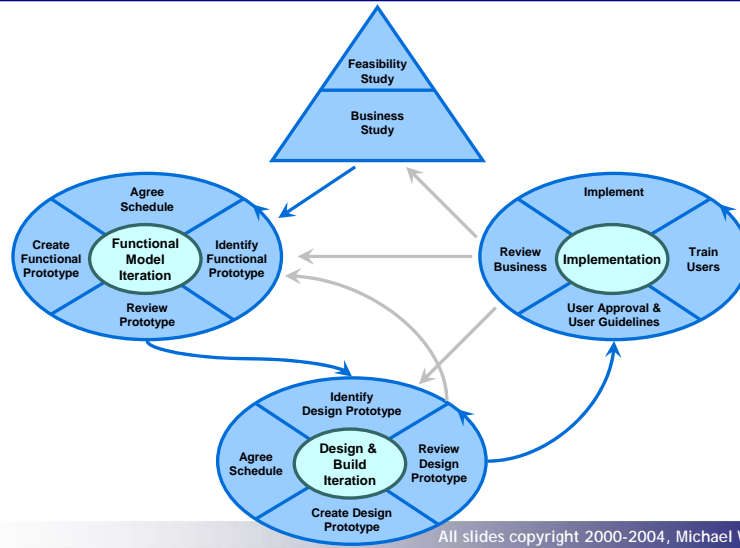
An idealized timebox

14

Timeboxing requires prioritization

- MoSCoW Rules
  - □ **M**ust have
    - fundamental to the system
  - □ **S**hould have
    - important requirement with short-term workaround, would normally be mandatory on a less time-constrained project
  - □ **C**ould have
    - can be left out of this increment
  - □ **W**ant to have but won't have this time
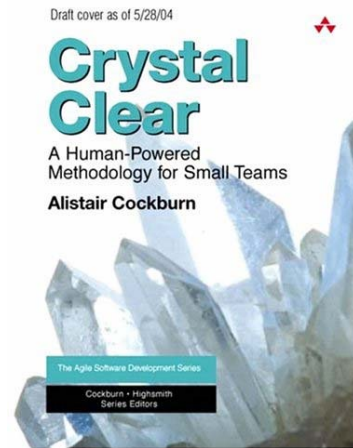   - Would like to have this increment but can wait for a future increment

Choose DSDM if…

- …the project has tight time constraints
- …the application is interactive or UI-intensive
- …you have clearly identifiable users
- …the project is small or can be made small by decomposing it
- …requirements can be prioritized
- …requirements are not clear or change frequently

Crystal Clear

- Alistair Cockburn
  - Project anthropologist
  - Interviews project teams around the world
- "Software development is a cooperative game of invention and communication."
  - —Alistair Cockburn

Draft cover as of 5/28/04

Crystal Clear
A Human-Powered Methodology for Small Teams

Alistair Cockburn

The Agile Software Development Series

Cockburn • Highsmith
Series Editors

The Crystal family

	Clear	Yellow	Orange	Red
Life (L)	L6	L20	L40	L80
Essential Money (E)	E6	E20	E40	E80
Discretionary Money (D)	D6	D20	D40	D80
Comfort (C)	C6	C20	C40	C80

Properties

Frequent delivery
- Delivery to customers, not just iterating
- Deliver to full user base every months
  - User viewings during interim

Reflective improvement
- Periodically think about your process:
  - What works?
  - What doesn't?

Osmotic communication
- Communication happens without thinking
- People need to close to each
- Keep an *expert in earshot*

Properties

Personal safety
- Ability to speak without fear of reprisal
  - Ignorance, inability or a mistake
- A first step toward trust

Focus
- Knowing what's important
- Minimizing (eliminating) other distractions

Easy access to expert users
- A place to deploy the frequent deliveries
- Rapid feedback on
  - delivered software
  - decisions
- Up to date requirements

Properties

Technical environment	▪ Three technical requirements 1. Automated testing 2. Configuration management 3. Frequent integration ▪ Best teams combine all three into: ▪ Continuous-integration-with-tests

Choose Crystal Clear if…

- …you want to do a lot of the process definition yourself
- …you have a small, collocated team
- …requirements are mostly known or knowable in advance
- …project involves loss of Comfort of Discretionary Money, not Essential Money or Life

Feature-Driven Development

- Originates in *Java Modeling in Color with UML* by Coad, Lefebvre and De Luca in 1999
- Jeff DeLuca
  - Main advocate
- Palmer and Felsing book in 2002

Features

- Serve as primary unit of work
  - Similar to XP stories or Scrum backlog items
  - Small enough to do in two weeks
- Feature Set
  - Collection of features
  - Assigned to a Chief Programmer and her team
- Major Feature Set
  - A domain area, one or more Feature Sets

Example features

- A short description of an action of value to users of the system

  > Estimate the closing price of a stock.

  > Calculate the total cost of an order.
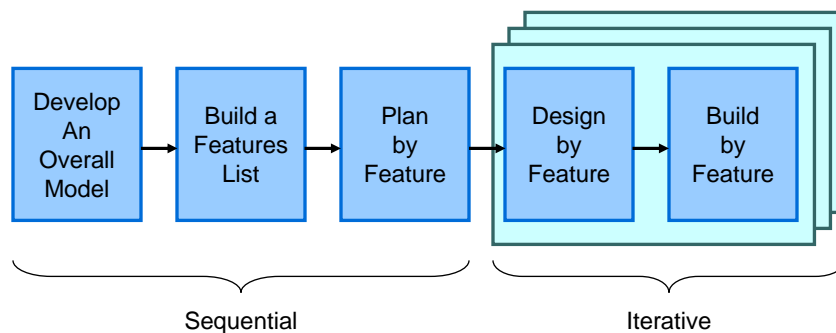
  > Change the password for a user.

  > Retrieve the room number of a guest.

- Format
  - <action> the <result> <by|for|of|to> a(n) <object>

Five processes

| Develop An Overall Model | Build a Features List | Plan by Feature | Design by Feature | Build by Feature |

Sequential   Iterative

Sample Development Plan

Major Feature Set	Feature Set	Feature	Chief Programmer	Date
Interfacing	Reservations	Make a reservation for a guest	Chris	Aug 2004
Interfacing	Reservations	Cancel a reservation for a guest	Chris	Aug 2004
Interfacing	Reservations	Update a reservation for a guest	Chris	Sept 2004
…	…	…	…	…
Reporting	Future Reservations	View future reservations for a hotel	Tod	Sept 2004
Reporting	Future Reservations	View future reservations for a competitive set	James	Sept 2004
…	…	…	…	…
Reporting	Rates	View Internet rates for a hotel	Andrew	Aug 2004

Eight "Best Practices"

- Need all 8 to be FDD
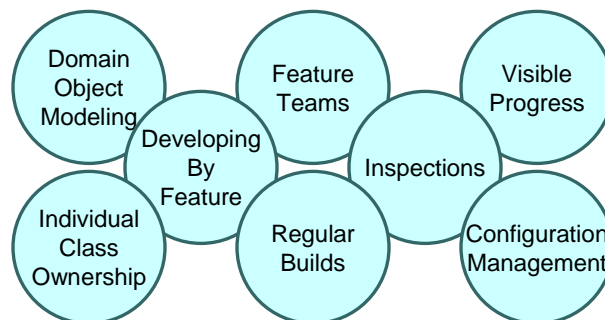
21

Tracking progress

Name of Feature Set ⟶ **Reservations**
Number of Features in Set ⟶ **(3)**
Percentage Complete ⟶ **65%**

Target Completion Month ⟶ **Aug 2002**

65%

Tod ⟵ Name of Chief Programmer

☐ Work in Progress
☐ Completed
☐ Attention
☐ Not Yet Started
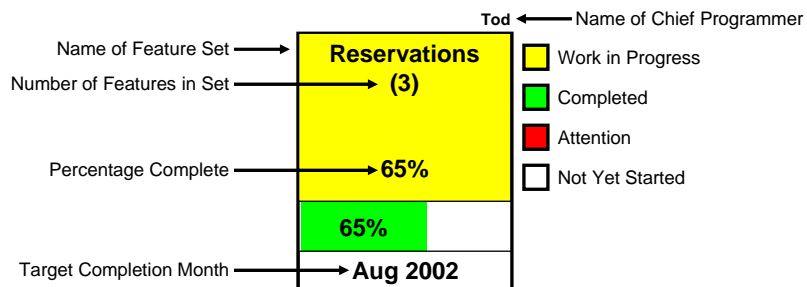
Choose FDD if…

- …you are willing to trade some agility for a well-defined way of scaling
- …your organization has solid UML skills
- …most requirements are knowable in advance or somewhat stable
- …you do not view self-organizing teams as a critical success factor

Today's agenda

❑ Extreme Programming
❑ Scrum
❑ DSDM
❑ Crystal Clear
❑ Feature-Driven Development
❑ **Overcoming common objections to agile**

---

1. It only works with talented people

- No, but you do need one "level three" developer
- Can a project with no level 3 developers work with ANY process?

1	Person learns to follow precise directions and get predictable results.
2	Person learns that there are multiple techniques.
3	Skills assimilated and can move between techniques without conscious thought.

Source: *Agile Software Development*,
Alistair Cockburn, p. 14.

2. It only works on trivial projects

- IDX
  - Scrum with over 600 developers
  - FDA regulated
- Caterpillar
  - Large team, XP, multi-year
- Other Scrum projects
  - ISO 9001
  - CMMI Level 3 per Mark Paulk of SEI
- We don't yet know what is possible

3. It's not appropriate for all projects

- OK, use it when you can
- We don't need a single process for all projects within a company

4. Agile is hacking

- More emphasis on testing in XP than any other process I've seen
  - Most importantly, programmers embrace it
- Planning is still part of the process
  - Scrum with two-week iterations:
    - 26 sprints per year;
    - 4 hours
    - 7 developer + 1 ScrumMaster
    - 26*4*8 = 832 hours of planning per year
  - "Don't confuse more exact with better."
    - —Brian Marick

What to learn from agile processes

- Communication is key
  - On-site customer, programmers in shared space
  - Communicate in person, not via documents
- Rapid feedback
- Cut out bureaucracy
- "Barely sufficient"
- Short increments
  - 1 week to 3 months

What to learn from agile processes

- Measure progress only by working code
- Customize the process
- Acknowledge the rapidly decreasing precision of plans
- You Aren't Gonna Need It (YAGNI)
  - □ Programmers won't need all the architecture they design
  - □ Customers don't need all the features
- Measure success with ROI not KLOC

Where to go next?

Agile Alliance

Agile in General	▪ www.agilealliance.com ▪ especially Articles and Roadmap sections ▪ www.mountaingoatsoftware.com
Scrum	▪ www.mountaingoatsoftware.com/scrum ▪ www.controlchaos.com ▪ scrumdevelopment@yahoogroups.com
Agile Planning	▪ www.mountaingoatsoftware.com/agileplanning ▪ groups.yahoo.com/agileplanning

Where to go next?

Agile Alliance

XP
- www.xprogramming.com
- c2.com/cgi/wiki?ExtremeProgrammingRoadmap
- extremeprogramming@yahoogroups.com
- http://www.extremeprogramming.org

DSDM
- www.dsdm.org

FDD
- www.featuredrivendevelopment.com

Crystal
- http://alistair.cockburn.us/crystal

For more on user stories

The Addison Wesley Signature Series

USER STORIES APPLIED

FOR AGILE SOFTWARE DEVELOPMENT

MIKE COHN
Foreword by Kent Beck

- www.userstories.com
- groups.yahoo.com/group/userstories

My contact information

Email —
- mike@mountaingoatsoftware.com
- mike.cohn@computer.org

- www.mountaingoatsoftware.com
- www.userstories.com
— **Websites**

28