SD BEST PRACTICES 2007 CONFERENCE & EXPO

September 18-21, 2007
Hynes Convention Center
Boston, MA

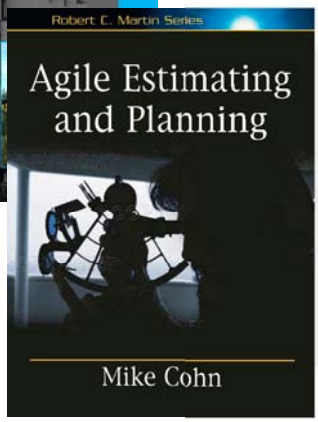
Selecting a Development Process:

Choosing Among the Leading Alternatives

Mike Cohn
Mountain Goat Software
mike@mountaingoatsoftware.com

the future of software development

Mike Cohn - background



Consultant, author, and speaker

- Founding member and director of Agile Alliance, Scrum Alliance, and Agile Project Leadership Network
- Founder of Mountain Goat Software

Today's agenda

- ☐ Considerations
- ☐ Team Software Process
- ☐ Scrum
- ☐ Extreme Programming
- ☐ OpenUP/Basic
- ☐ Feature-Driven Development

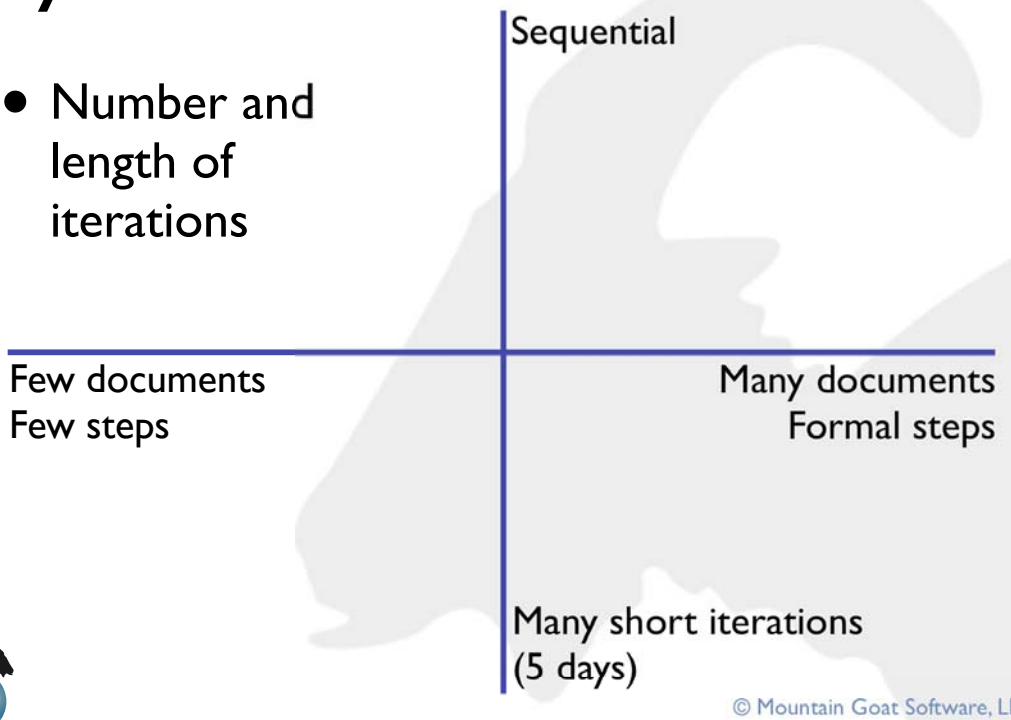
Ceremony

- The amount of formalism in a process
  - Documentation, method weight, reviews

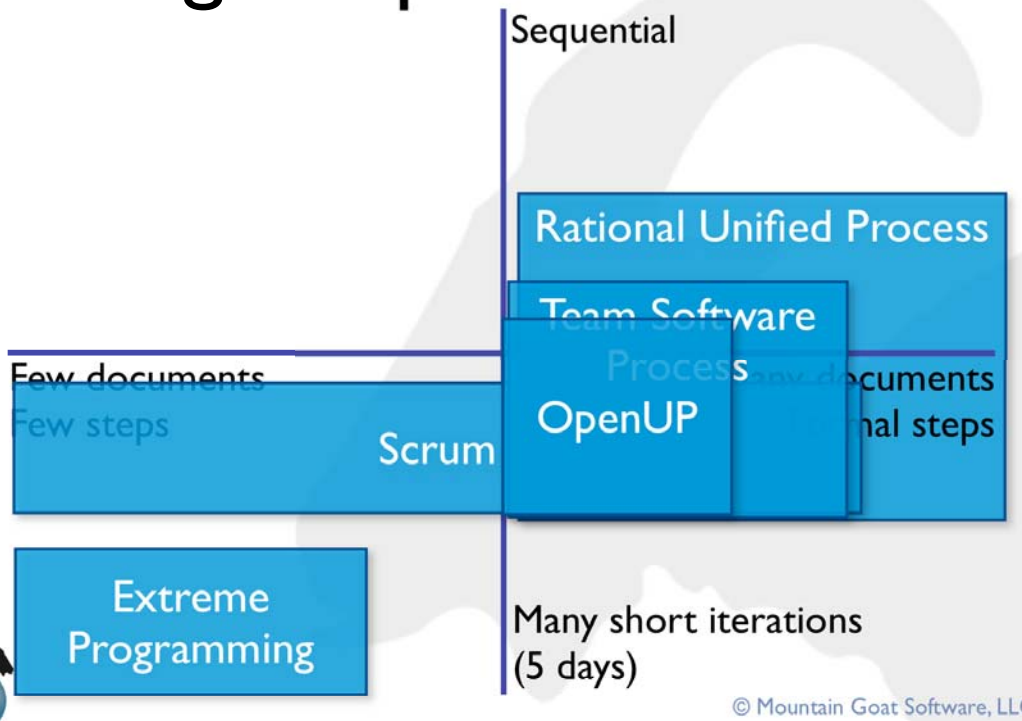
Few documents	Many documents
Few steps	Formal steps

Cycles

- Number and length of iterations



Placing the processes



Today's agenda

- [x] Considerations
- [ ] Team Software Process
- [ ] Scrum
- [ ] Extreme Programming
- [ ] OpenUP/Basic
- [ ] Feature-Driven Development

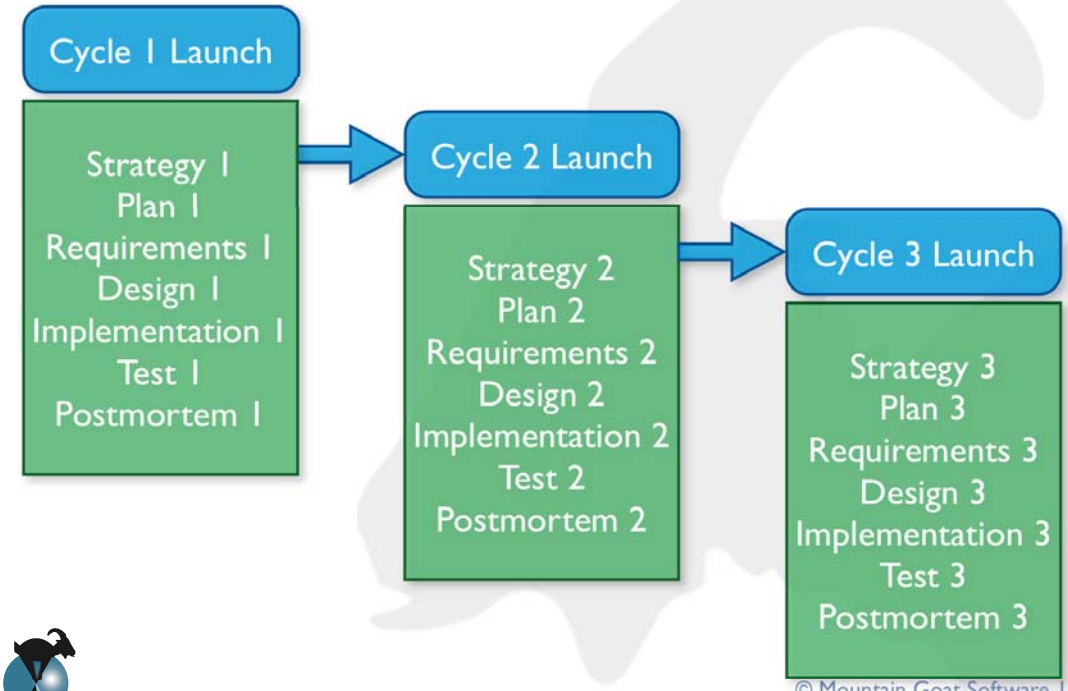
Team Software Process (TSP)

- Created by Watts Humphrey
  - Of Software Engineering Institute and Capability Maturity Model (CMM)
  - Builds on his Personal Software Process
- High discipline, highly defined
- A "cyclic development strategy"
  - Another way of saying "iterative and incremental"

Goals of the TSP

1. Build on the Personal Software Process
2. Develop products in cycles
3. Establish standard measures for quality and performance
4. Provide precise measures
5. Use role and team evaluations
6. Require process discipline
7. Provide guidance on teamwork problems

Team Software Process (TSP)



Cyclic development

- A *testable version* is produced at the end of each cycle
  - Purpose of testing is to assess the product, not fix it
  - Defects should be removed during the cycle

Designed to solve team problems

"The success or failure of a project is seldom due to technical issues. You almost never find yourself asking 'has the state of the art advanced far enough so that this program can be written?' Of course it has. If the project goes down the tubes, it will be non-technical, human interaction problems that do it in. The team will fail to bond, or the developers will fail to gain rapport with the users, or people will fight interminably over meaningless methodological issues."

Tom DeMarco

Teams, members, roles and goals

- A team has one or more members
- Each member fills one or more roles

Team leader	Quality/process manager
Development manager	Support manager
Planning manager	

- Goals are assigned throughout this hierarchy
  - Team, member, role

Goals and goals and...

Team goals

- Produce a quality product
- Run a productive and well-managed project
- Finish on time

Team member goals

- Be a cooperative and effective team member
- Do consistently disciplined personal work
- Plan and track all your personal work
- Produce quality products

...more goals

Dev manager goals

- Produce a superior product
- Fully use the team members' skills and abilities

Some of these goals point out some underlying beliefs of TSP, such as:

- You can't control what you can't measure
- Managers "use the team members' skills"

Scripts and forms

- TSP uses scripts to define the steps of building a software system
  - 21 activity scripts
  - 10 role scripts
- Often quite detailed
- Scripts prescribe completing 21 supporting forms

The inspection script

Purpose		To help engineers produce quality products
Entry criteria		A completed and reviewed product with available materials
Step	Activities	Description
1	Plan the inspection	The developer • Arranges with the quality/process manager or some other qualified team member to be the moderator • Handles the mechanics of setting up and running…
2	Hold the inspection	The moderator describes the inspection process The producer familiar the inspection team with the product.
…	…	…
Exit criteria		INS and LOGD forms completed and filed in the …

Choose TSP if...

- …you have the discipline to follow all of its specific steps
- …are already using the complementary Personal Software Process (PSP)
- …team members need this level of guidance
- …you do not have emergent requirements

Today's agenda

- [x] Considerations
- [x] Team Software Process
- [ ] Scrum
- [ ] Extreme Programming
- [ ] OpenUP/Basic
- [ ] Feature-Driven Development

Scrum 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

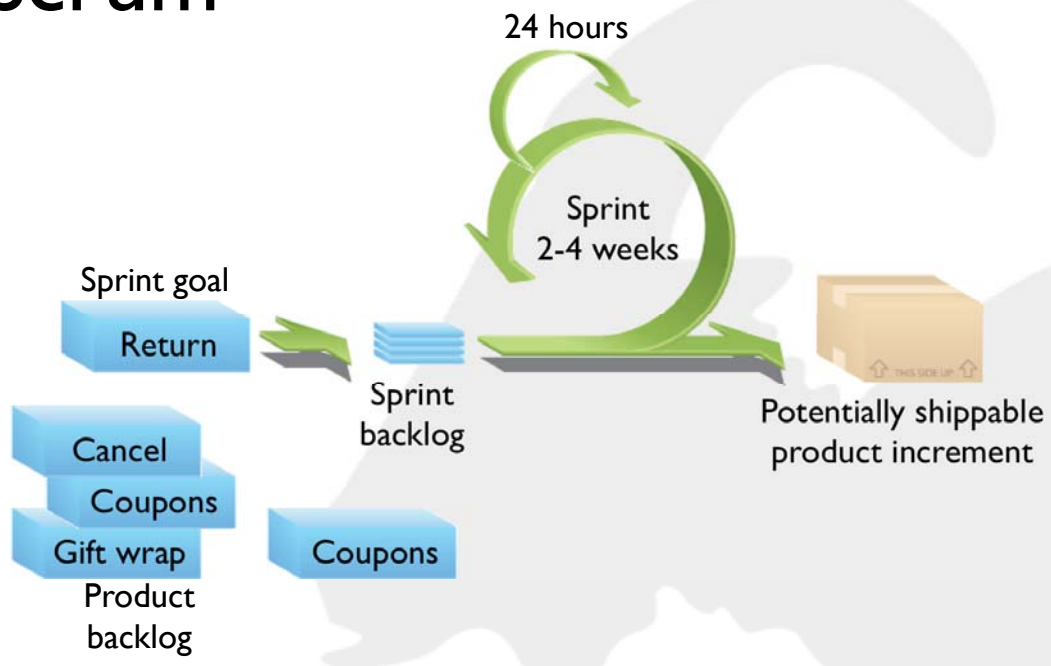
Scrum roles

Scrum team
- 7–9 full-time individuals
- Self-organizing
- Ideally full-time on project

Product owner
- Customer or user of the project
- Makes prioritization decisions

ScrumMaster
- Not quite the project manager
- Removes impediments
- Enforces Scrum values

Scrum



Scrum roles and responsibilities

Product Owner

- Defines the features of the product, decides on release date and content
- Is responsible for the profitability of the product (ROI)
- Prioritizes features according to market value
- Can change features and priority every sprint
- Accepts or rejects work results

Scrum Master

- Ensures that the team is fully functional and productive
- Enables close cooperation across all roles and functions and removes barriers
- Shields the team from external interferences
- Ensures that the process is followed.

Team

- Cross-functional, seven plus/minus two members
- Selects the sprint backlog
- Has the right to do everything within the boundaries of the project guidelines to reach the sprint goal
- Organizes itself and its work
- Demos work results to the Product Owner

Scrum meetings

Sprint planning

- Team selects product backlog items to complete during sprint
- Tasks are identified and estimated

Daily scrum

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

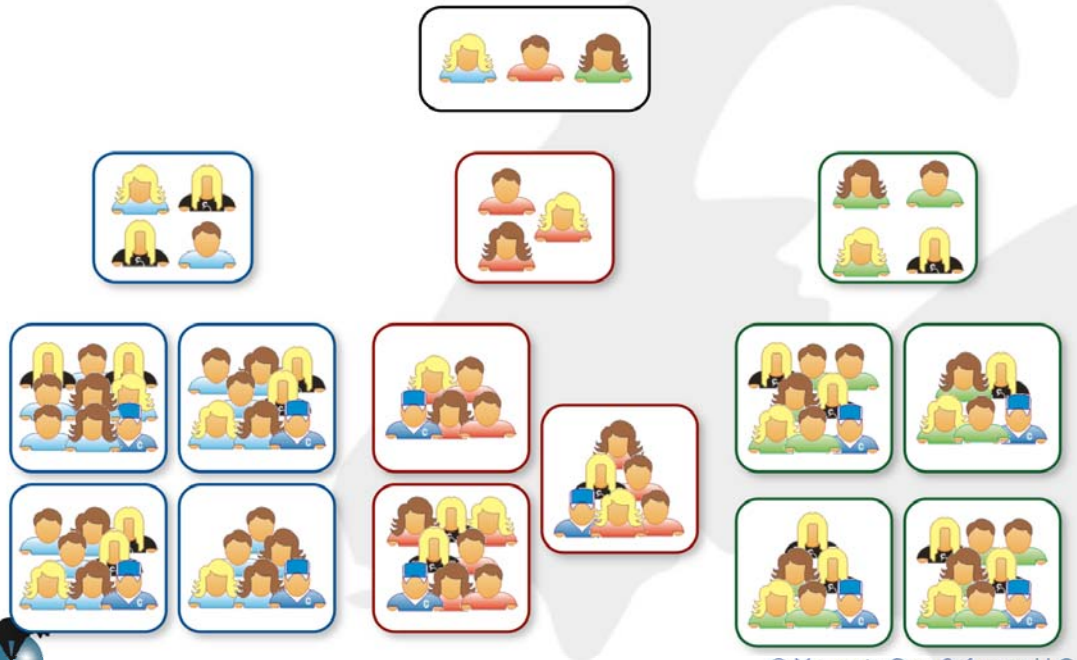
Sprint review

- Team demonstrates working software built in the sprint

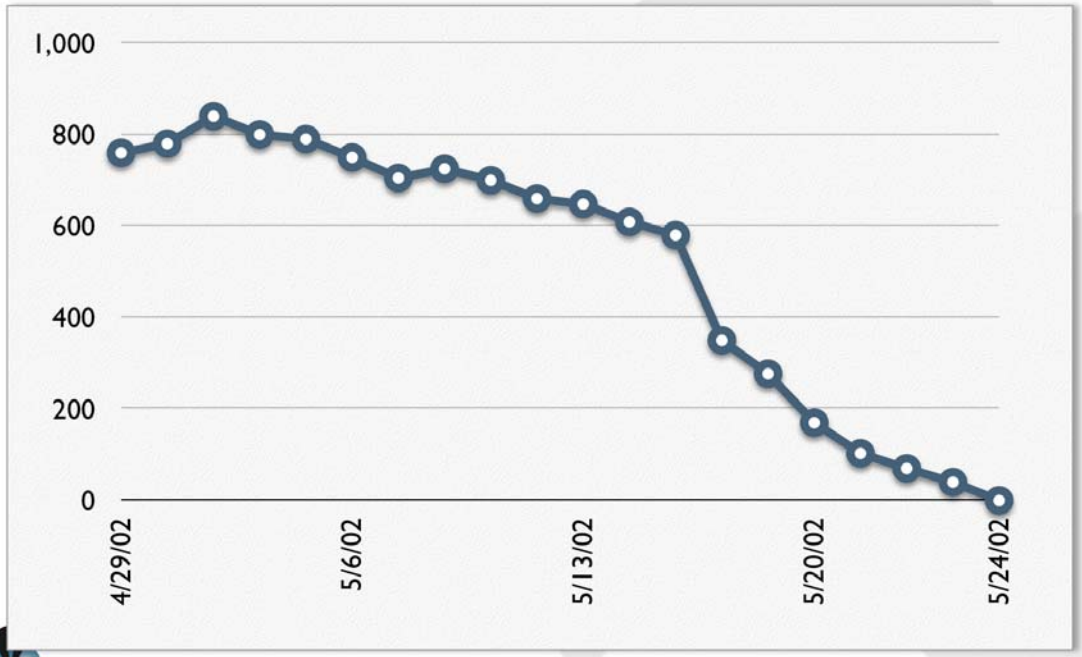
Sprint retrospective

- Team reflects on sprint and looks for ways to improve

Scrum of scrums of scrums



A sprint burndown chart



Choose Scrum if...

- …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 want to better manage risk
- …you need a proven, scalable agile process

Today's agenda

- ☑ Considerations
- ☑ Team Software Process
- ☑ Scrum
- ☐ Extreme Programming
- ☐ OpenUP/Basic
- ☐ Feature-Driven Development

Extreme Programming

- Adds specific engineering practices to Scrum's project management approach
- A tough target to hit but highly productive if you do

XP's customer practices

On-site customer

Small releases

XP's quality practices

- Metaphor
- Testing
- Simple design
- Refactoring
- Pair programming

XP's quality practices

- The Planning Game
- Sustainable pace
- Collective ownership
- Coding standards
- Continuous integration

Choose Extreme Programming 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
- …it's important you hit the bull's eye right off and you think you can

Today's agenda

- ☑ Considerations
- ☑ Team Software Process
- ☑ Scrum
- ☑ Extreme Programming
- ☐ OpenUP/Basic
- ☐ Feature-Driven Development

OpenUP/Basic

- A subset of the Open Unified Process
- Delivered through the Eclipse Process Framework
- Derived from the Basic Unified Process
  - (If they're unified why are there so many of them?)
- Targeted at teams of 3-6 people
- Targeted at projects 3-6 months long

Comparison with RUP

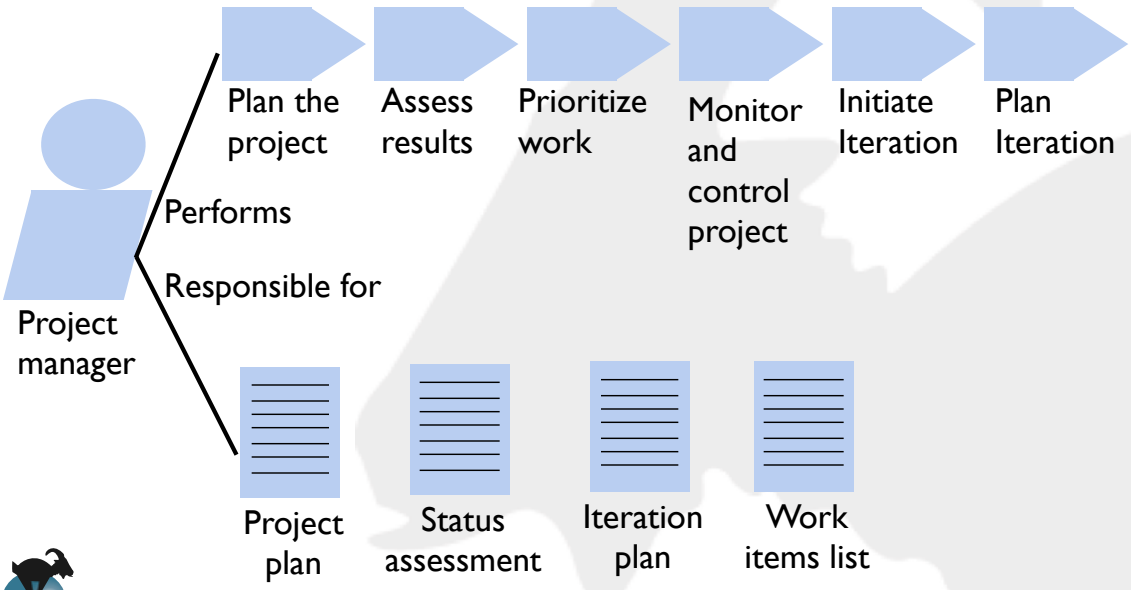
	OpenUP/Basic	RUP
Pages	190	~2000
Roles	6	~45
Tasks	23	~150
Artifacts	17	~80

Source: *What is the Eclipse Process Framework*,
Per Kroll and Scott Ambler,
IBM Rational Software Development Conference 2006.

OpenUP roles

- Analyst
- Architect
- Developer
- Tester
- Project manager

Roles, tasks, and artifacts



Open UP phases

Inception

- Establish scope and produce business case
- Mitigate business risks
- Get buy-in and decision to undertake project

Elaboration

- Reduce major risks to enable cost and schedule estimates
- Mitigate major technical risks
- Implement a baseline architecture
- Implement key capabilities

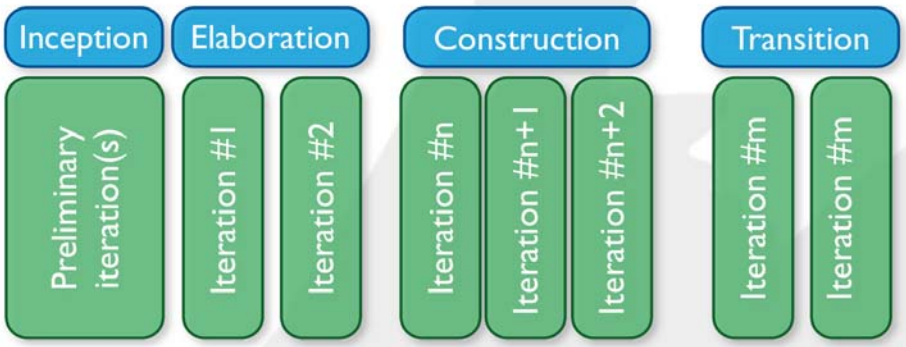
Construction

- Majority of implementation
- Ends with deployment of a beta version

Transition

- Ensure software meets users' needs
- Prepare for release
- Solicit user feedback and make minor adjustments

Iterate within each phase



Choose OpenUP/Basic if...

- You like RUP but want something lighter
  - You want an open source version of RUP
- You have a small project (3-6 people; 3-6 months)
- You want a semi-agile process based on role, task and artifact guidance

Today's agenda

- [x] Considerations
- [x] Team Software Process
- [x] Scrum
- [x] Extreme Programming
- [x] OpenUP/Basic
- [ ] Feature-Driven Development

Feature-Driven Development

- Originates in *Java Modeling in Color with UML* by Coad, Lefebvre and De Luca in 1999
- Jeff DeLuca is main advocate
- *A Practical Guide to Feature-Driven Development* by Palmer and Felsing in 2002

Features and feature sets

- 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

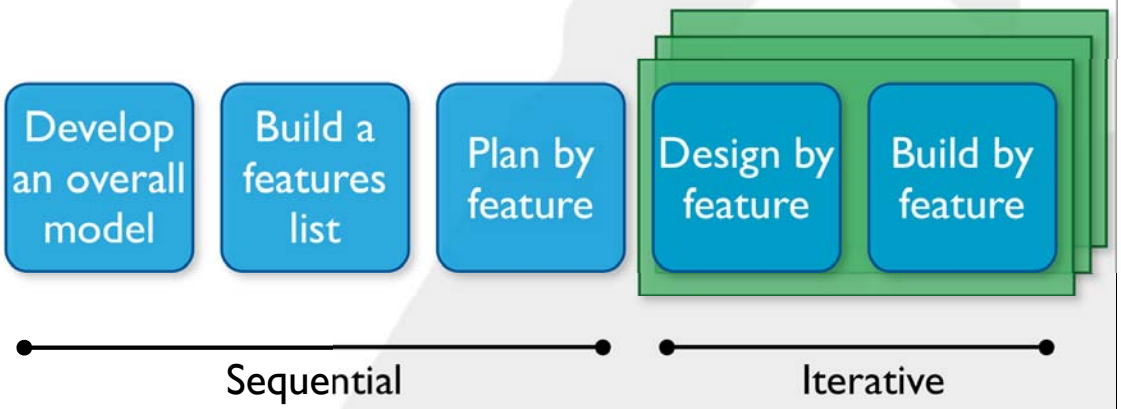
Examples

- 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

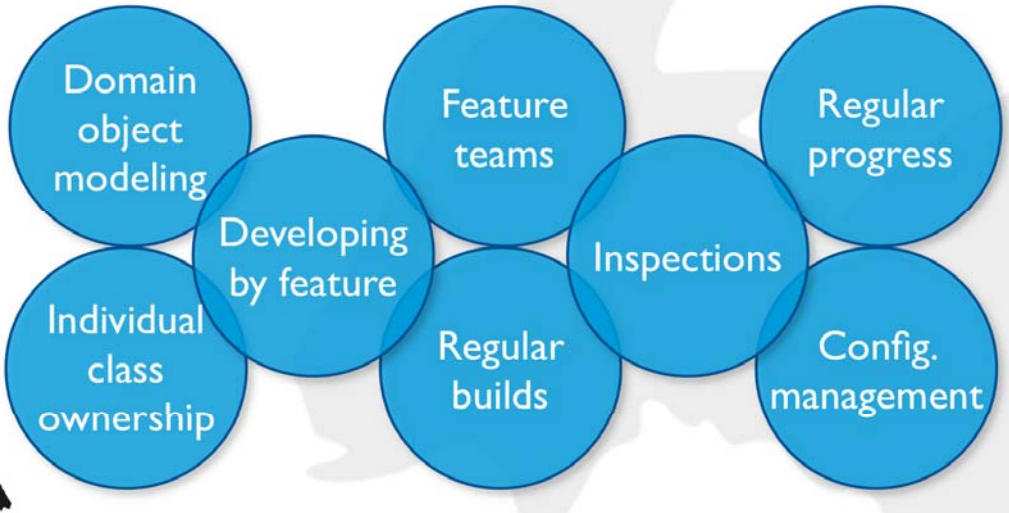


Sample development plan

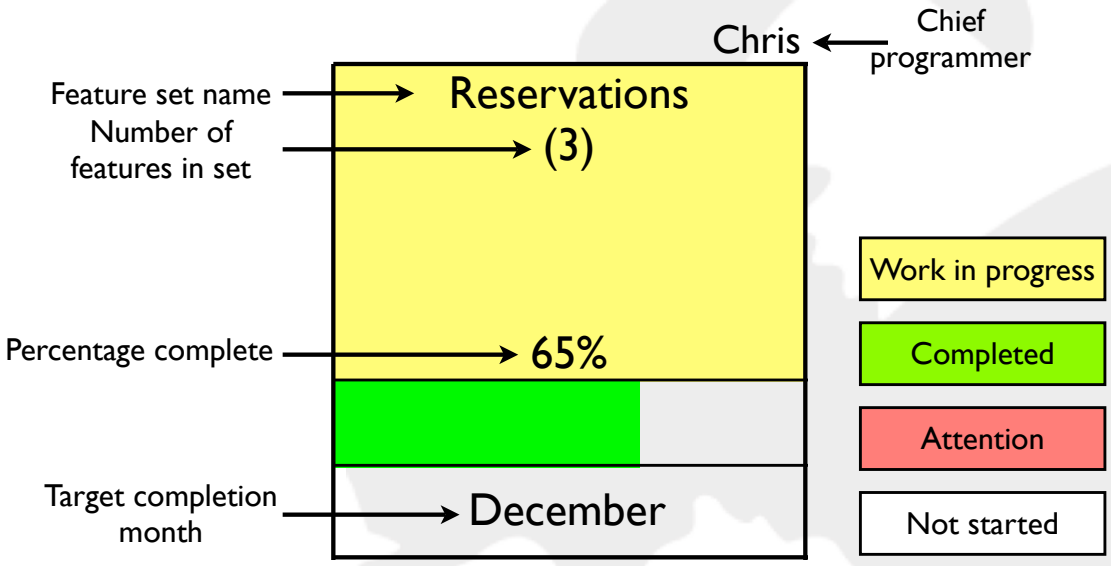
Major feature set	Feature set	Feature	Chief programmer	Date
Interfacing	Reservations	Make a reservation for a guest	Chris	December
Interfacing	Reservations	Cancel a reservation for a guest	Chris	December
Interfacing	Reservations	Update a reservation for a guest	Chris	January
Reporting	Future reservations	View future reservations of competitors	James	December
Reporting	Rates	View internet rates for a hotel	Kristy	February

Eight best practices

- Need all eight to be doing FDD



Tracking progress



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

Where to go next

Team Software Process	• *Introduction to the Team Software Process* by Watts S. Humphrey
Scrum	• www.mountaingoatsoftware.com/scrum • *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle • *Agile Project Management with Scrum* by Ken Schwaber
Extreme Programming	• www.xprogramming.com • c2.com/cgi/wiki?ExtremeProgrammingRoadmap • extremeprogramming@yahoogroups.com

Where to go next

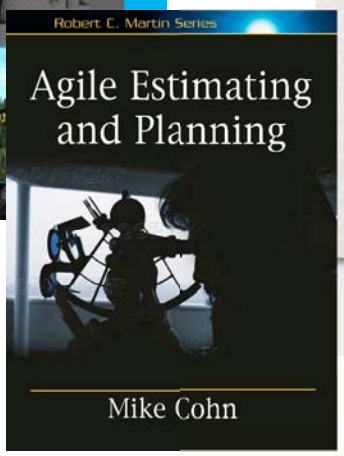
OpenUP	• www.eclipse.org/epf • *Agility and Discipline Made Easy* by Per Kroll and Bruce MacIsaac
FDD	• www.featuredrivendevelopment.com
Agile in General	• www.agilealliance.org • www.mountaingoatsoftware.com

Upcoming public classes

Date	What	Where
Oct 18-19	Certified Scrum Product Owner	Denver
Oct 23-24 Oct 25	Certified ScrumMaster Agile Estimating and Planning	San Diego
Jan 15-16 Jan 17	Certified ScrumMaster Agile Estimating and Planning	Atlanta
Feb 26-27 Feb 28	Certified ScrumMaster Agile Estimating and Planning	Santa Clara
Other classes in London and Oslo if you're up for a longer trip.		

More info at www.mountaingoatsoftware.com

Mike Cohn contact info



mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)

MOUNTAIN GOAT SOFTWARE