CPSC 310 – Software Engineering

Lecture 3 – Agile Process Models
Scrum

What is “agility”?

- Adapt to change
  - Particularly changes in requirements
  - Use frequent, short iterations to flatten cost curve
- Focus on working software
  - What works is how success is measured
- Process based on collaboration
  - Heavy customer involvement
- A sustainable process
  - Find a cadence that delivers reliably

Agile Methods

- Extreme Programming
  - Specific practices - customer driven development, small teams, daily builds
- Scrum
  - Project management approach, relying on self-organizing independent teams
- Several others …

Extreme Programming

- Developed by Kent Beck at Chrysler in mid-90s

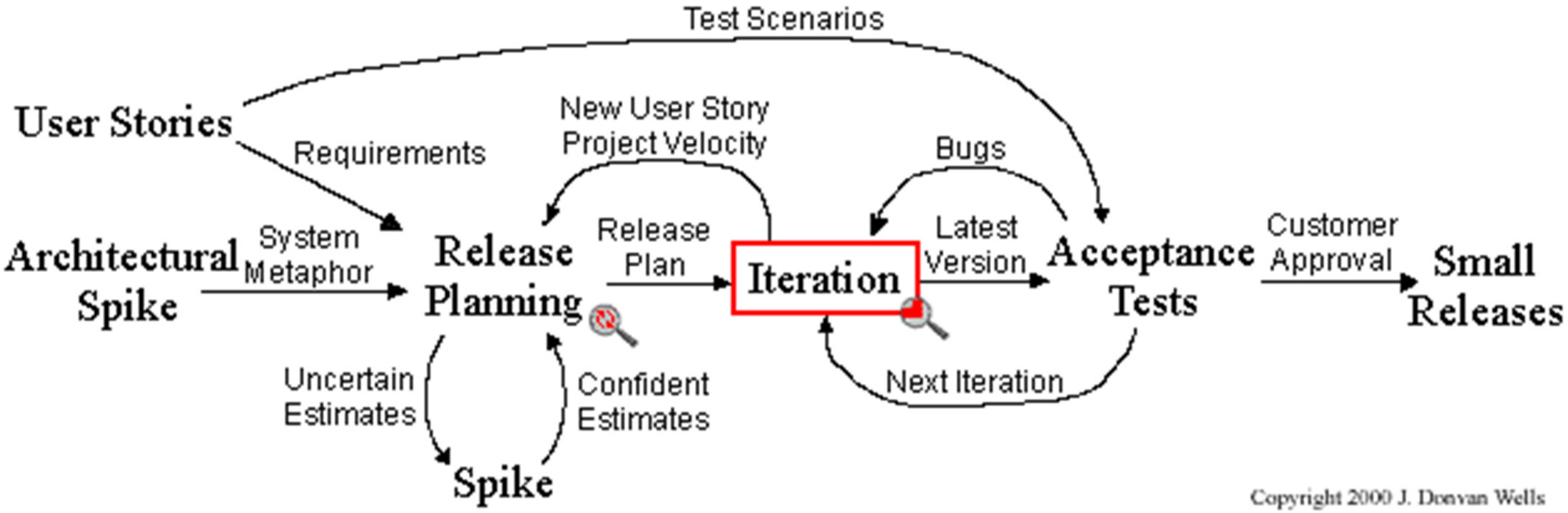
Values of XP

Five principal values:

1. **Communication**: common metaphors, frequent verbal communication, customer involvement
2. **Simplicity**: do the simplest thing that could possibly work, then refactor
3. **Feedback**: from the code (unit tests), the customer (co-location), the team (planning game)
4. **Courage**: be willing to throw things away
5. **Respect**: don't do things that make work for others

Thanks to Greg Wilson for slides:
http://www.cdf.toronto.edu/~csc301h/fall/lec/02-agile.html

Extreme Programming Project



Thanks to Greg Wilson for slides:
http://www.cdf.toronto.edu/~csc301h/fall/lec/02-agile.html

XP Practices

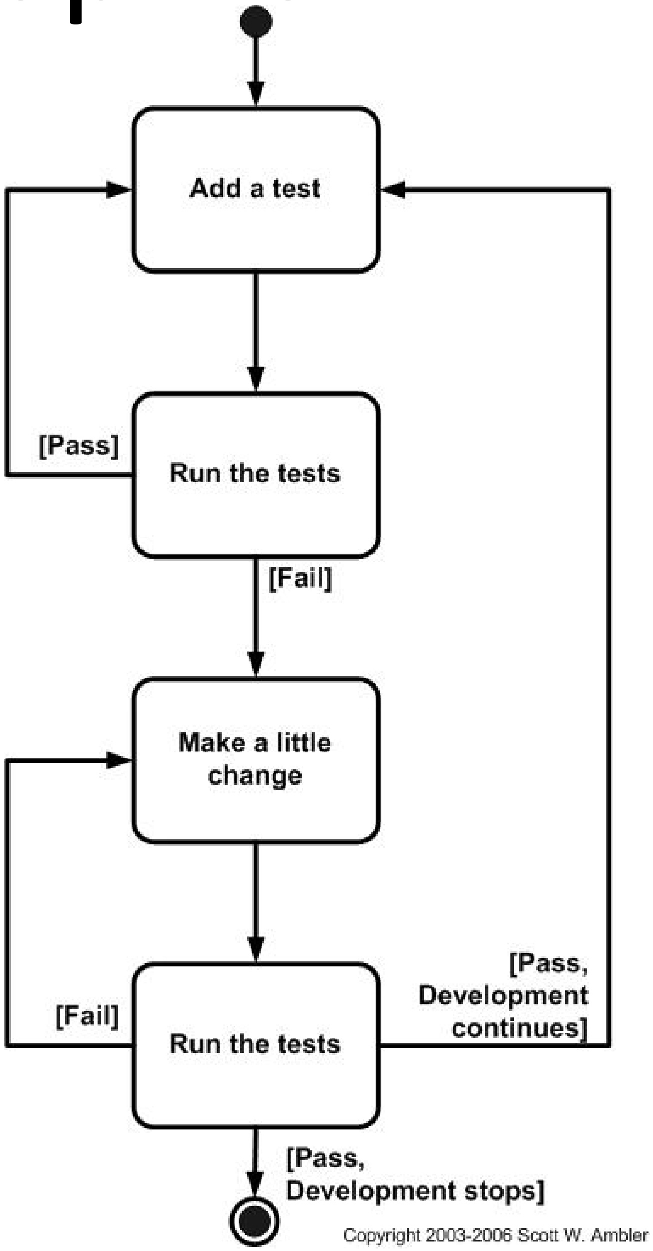
- Pair programming
- Test-driven development
- Continuous integration
- Shared metaphor
- Small releases
- Planning game
- …

http://www.extremeprogramming.org/rules.html

Test-Driven Development

- Test cases are written first
  - Cover new functionality or improvement
- Then the necessary function is implemented
- Code is "complete" when all tests pass

- **Refactor** before adding feature if design could be better



Copyright 2003-2006 Scott W. Ambler

Scrum

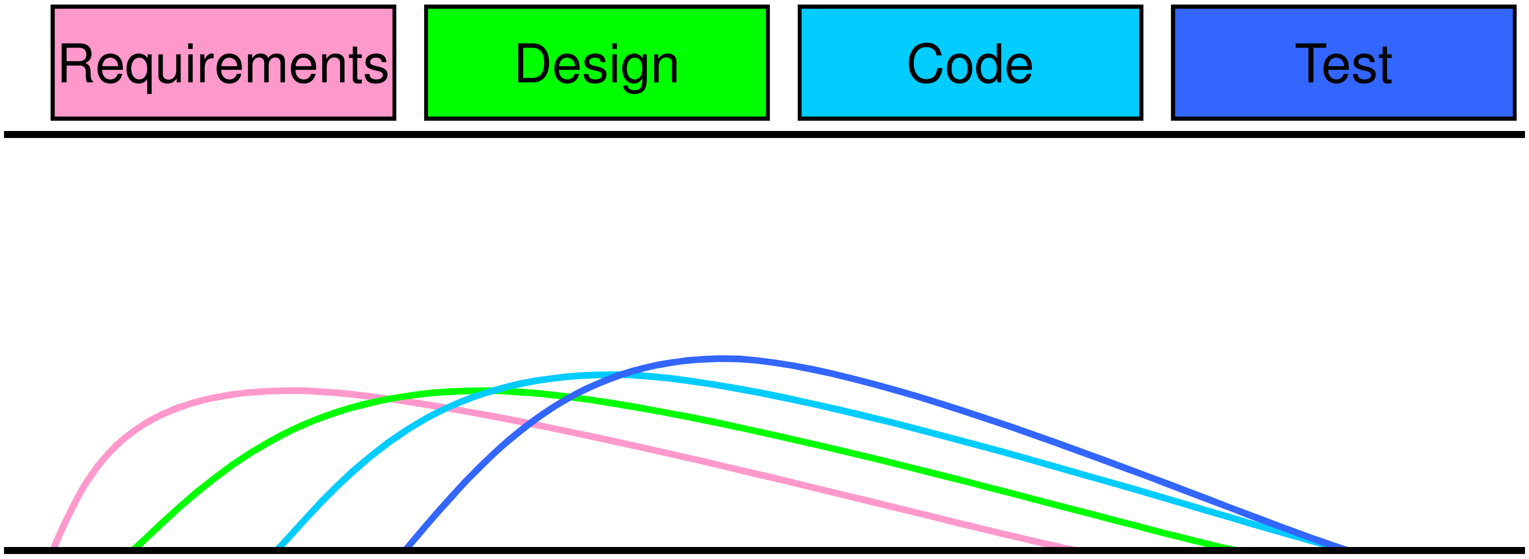
Management framework for incremental, overlapping, product development

- Self-organizing, cross-functional teams
- Product progresses in a series of two- to four-week (fixed length) iterations: ***sprints***
- Every iteration produces a potentially shippable (properly tested but not complete) product
- Requirements are captured as items in a list: ***product backlog***
- No specific engineering practices prescribed (unlike XP)
  - e.g. unit testing, refactoring, code standards, etc…

www.mountaingoatsoftware.com

The original “Scrum”

Sequential vs. Overlapping Dev.

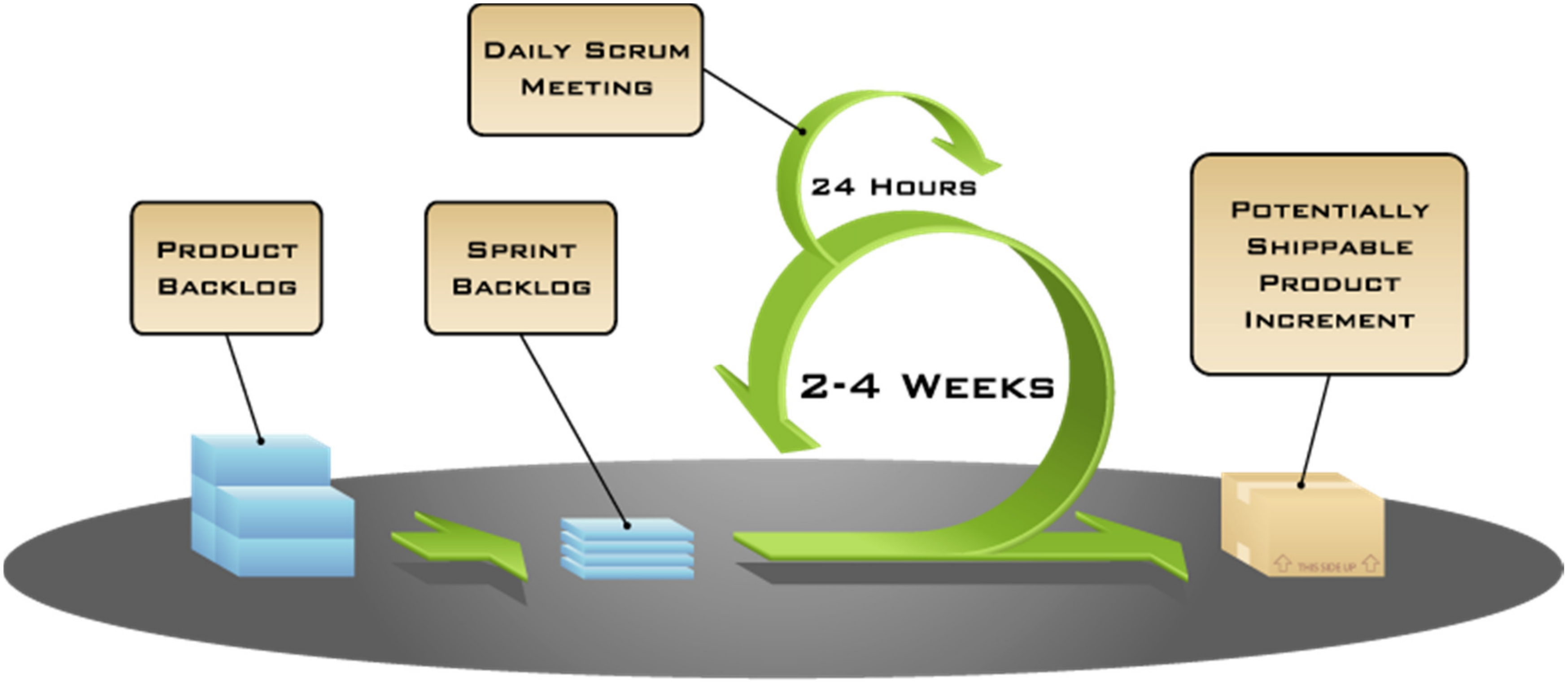


www.mountaingoatsoftware.com

Scrum Framework

1. **Roles** : Product Owner, Scrum Master, Team
2. **Artifacts** : Product Backlog, Sprint Backlog, and Burndown Chart
3. **Ceremonies** : Sprint Planning, Daily Scrum Meeting, Sprint Review

Scrum



Copyright © 2005, Mountain Goat Software

www.mountaingoatsoftware.com

Scrum Roles

- Product Owner
  - Defines features of the product
  - Prioritizes features according to market value
  - Adjust features and priorities every iteration, as needed
- Scrum Master
  - Facilitates Scrum process
  - Helps resolve impediments
  - Shields team from external interferences
  - NOT the manager
- Team
  - Self-organizing, self-managing, cross-functional
    - Developers, designers, managers, clients, etc…
  - 7 (+/- 2) people

Scrum Artifacts

- Product Backlog
  - ***prioritized*** list of backlog items (PBIs)
  - PBIs specify a ***customer-centric feature*** (User Story form)
  - effort estimated by Team, priority estimated by Product Owner
- Sprint Backlog
  - contains list of ***engineering tasks*** that are negotiated by team and product owner from the Product Backlog
  - negotiated PBIs broken down into specific *tasks*
- Burndown Chart
  - Total ***remaining team task hours*** within one sprint

Sprint Backlog

- List of tasks Scrum team commits to for sprint

- Based on priorities and team’s perception of required time (normally between 4 and 16 hours)

- Tasks in Sprint Backlog represent **developer’s (technical)** perspective, not customer perspective

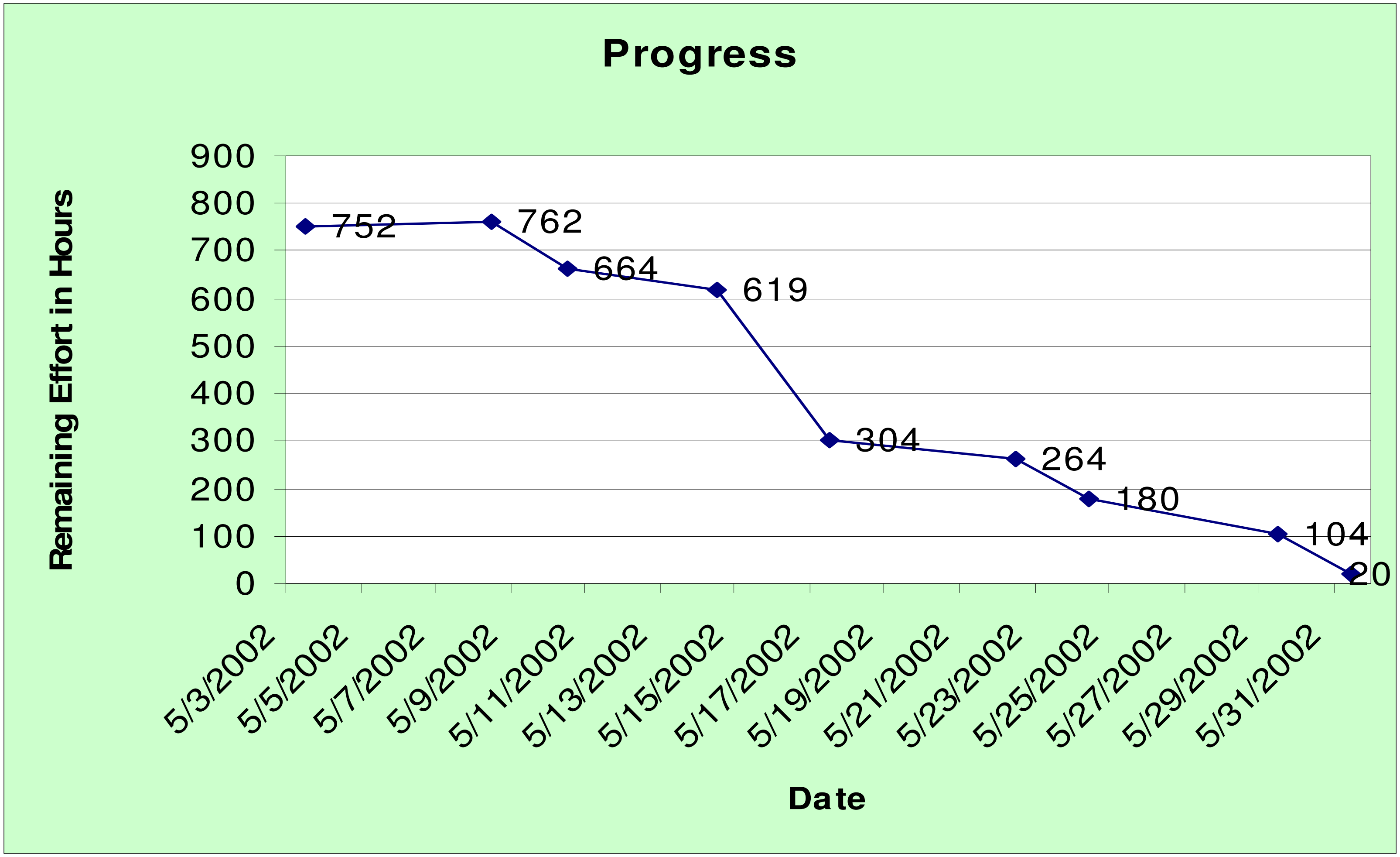
Example – Sprint Backlog Items

Weather App:

- PBI: As a subscriber, I want to see a 10-day forecast of conditions so that I can plan at least a week ahead
- Tasks:
  - Parse the weather data in day packets
  - Push several days data to the client

- PBI: As a subscriber, I want to see precipitation accumulations and forecast so that I can plan my activities.
- Tasks:
  - Parse snow/rain data from the provider’s data
  - Push the snow/rain data to the client
  - Redesign client screen for new data
  - Refactor the server code

Sprint Burndown Chart



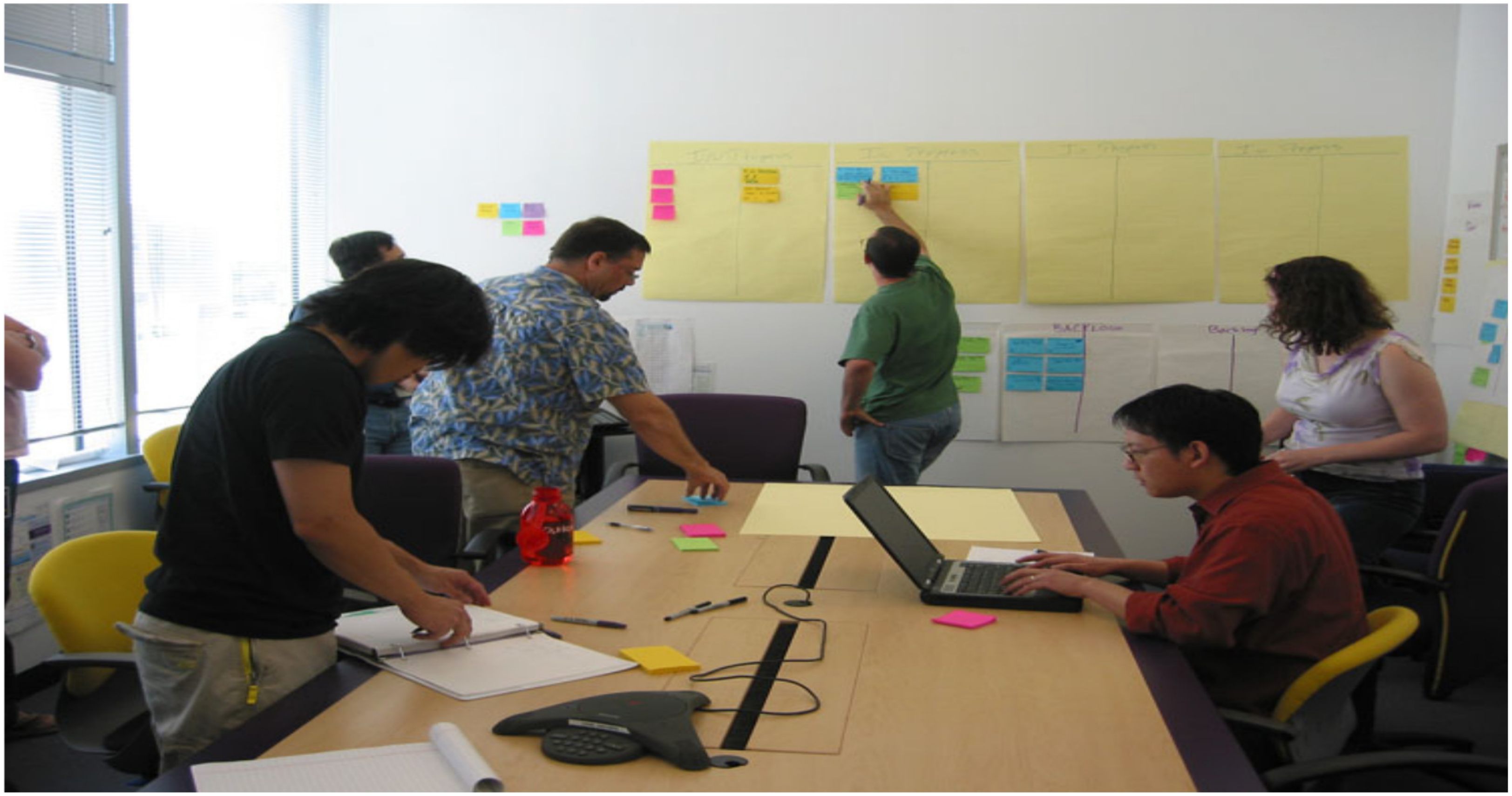
Ceremony: Sprint Planning

- Sprint
  - A 2-4 week iteration, during which features are added
- Changes should not be made to requirements during the sprint
- Each Sprint begins with the Daily Scrum Meeting

Ceremony: Daily Scrum

- Parameters
  - Daily
  - 15-minutes
  - Stand-up
  - Not for problem solving
- Three questions:
  1. What did you do yesterday
  2. What will you do today?
  3. What obstacles are in your way?
- Task estimates may need to be adjusted
- Everyone is invited …
  - Help avoid other unnecessary meetings
- But only team members should talk

Daily Scrum



Q: What might these people be doing with the colored pieces of paper?

Daily Scrum

- Is NOT a problem solving session
- Is NOT a way to collect information about WHO is behind the schedule
- Is a meeting in which team members make commitments to each other and to the Scrum Master
- Is a good way for a Scrum Master to track the progress of the Team

Ceremony: Sprint Review Meeting

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
  - 2-hour prep time rule
- As in the Daily Scrum, everyone is invited

Scrum Summary

- Agile project management framework
  - Not an engineering methodology
- Evolved refinement of Spiral Model
  - Not a completely different model
- Very popular today
  - Good chance you will encounter it in “real world”