CS 3773
Software Engineering
Lecture 3

Dr. Mark Robinson
Office: NPB 3.350

Agile SE Processes

- **Agility**: a SE characteristic that acknowledges change as a necessary part of SE and promotes efficient, appropriate response to change
- Based on iterative SE process model (iteration = sprint)
- Emerged in 1990's
- Opposition to slower, heavier, more bureaucratic process models
- Deliver working software faster
- Manifesto and Agile Alliance in 2001

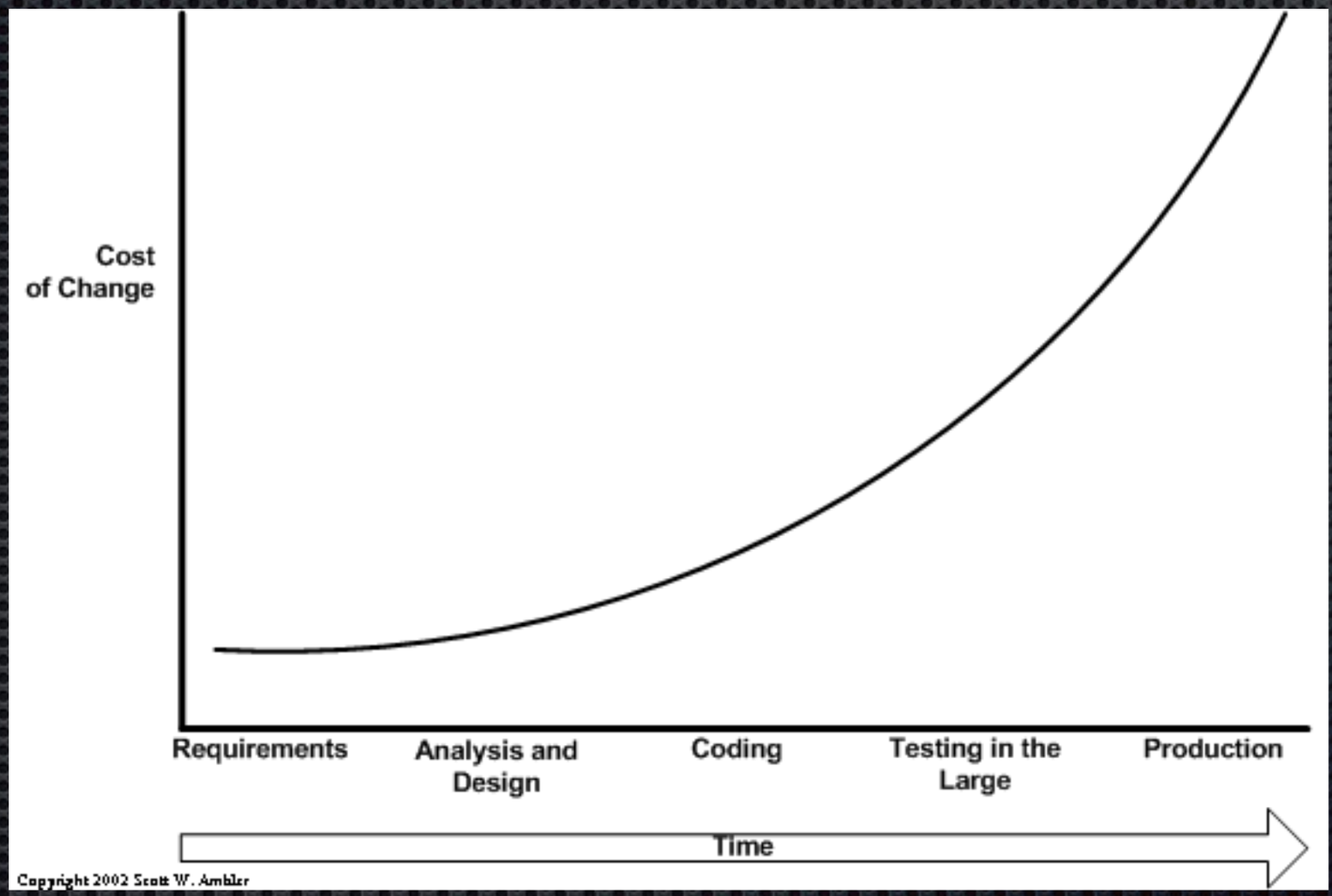
Agile Manifesto

- “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”

Agile Principles

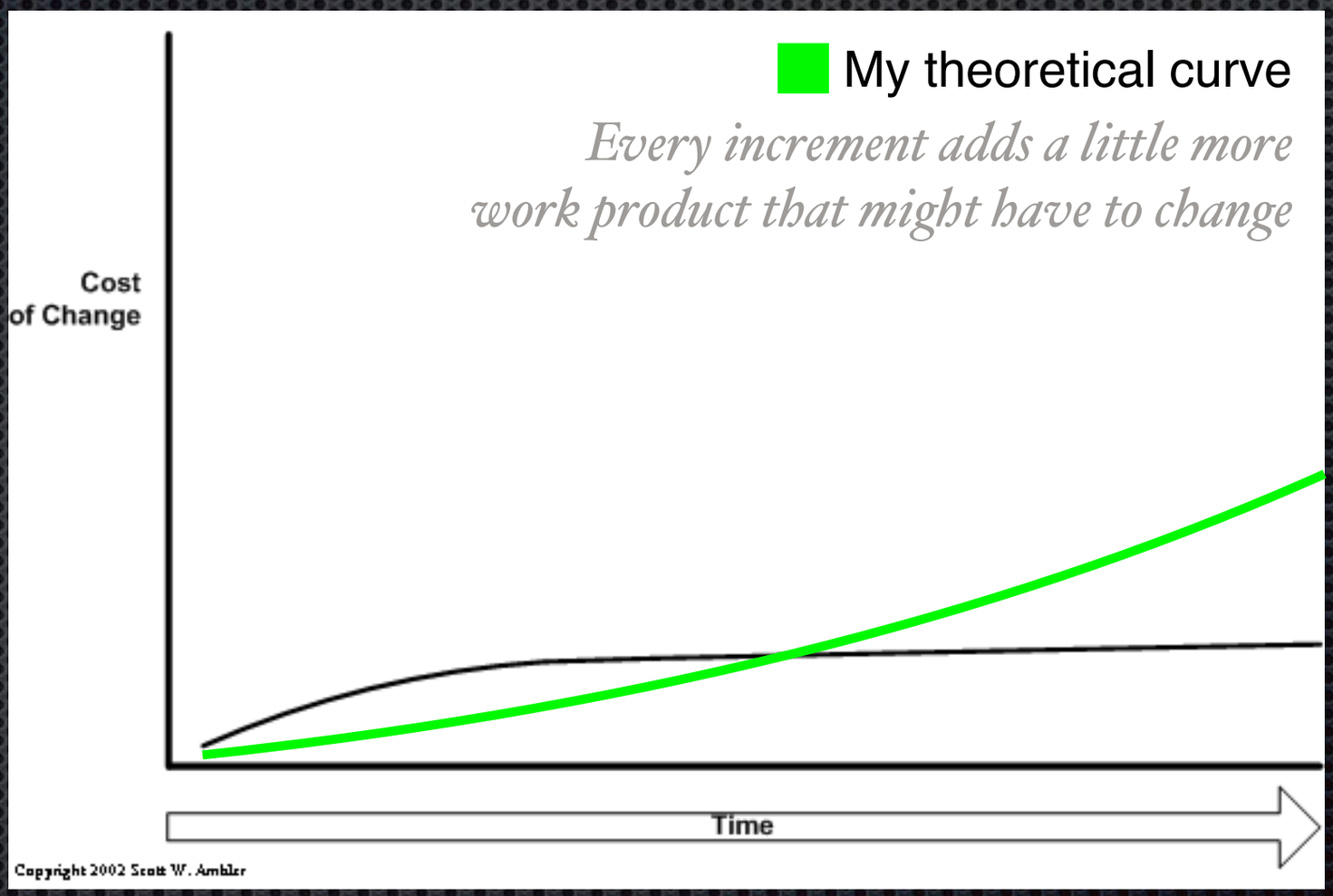
- Customer satisfaction by rapid, continuous delivery of useful software
- **Working software** is delivered frequently (weeks rather than months)
- Working software is the principal measure of progress
- Even late **changes in requirements** are welcomed
- Close, daily cooperation between business people and developers
- Face-to-face conversation is the best form of **communication** (co-location)
- Projects are built around motivated individuals, who should be **trusted**
- Continuous attention to technical excellence and good design
- **Simplicity**
- **Self-organizing** teams (team itself, the process, and sprint schedule)
- Regular **adaptation** to changing circumstances

The Cost of Change



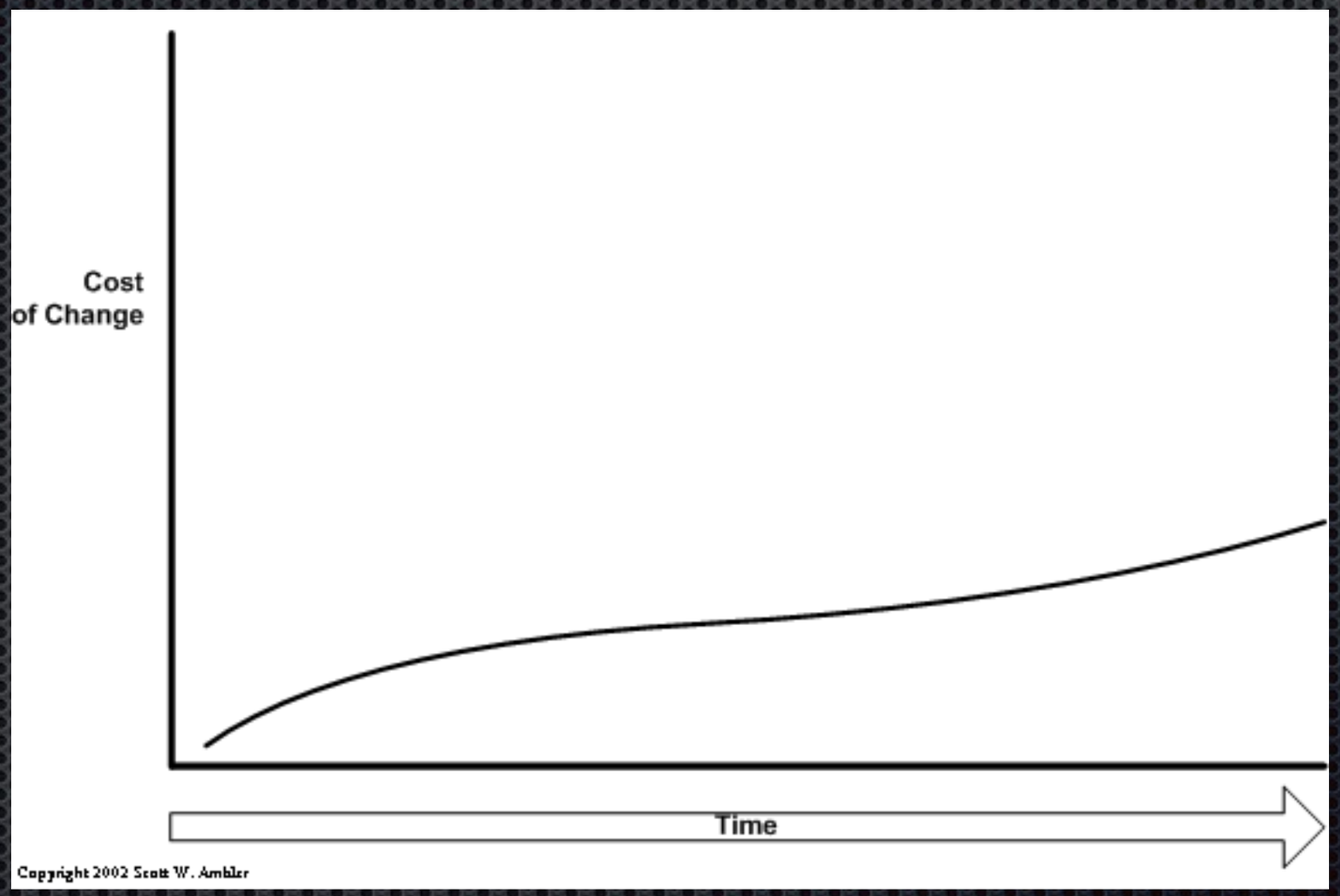
Linear (traditional) process models

The Cost of Change



Agile models (theoretical)

The Cost of Change



Agile models (realistic)

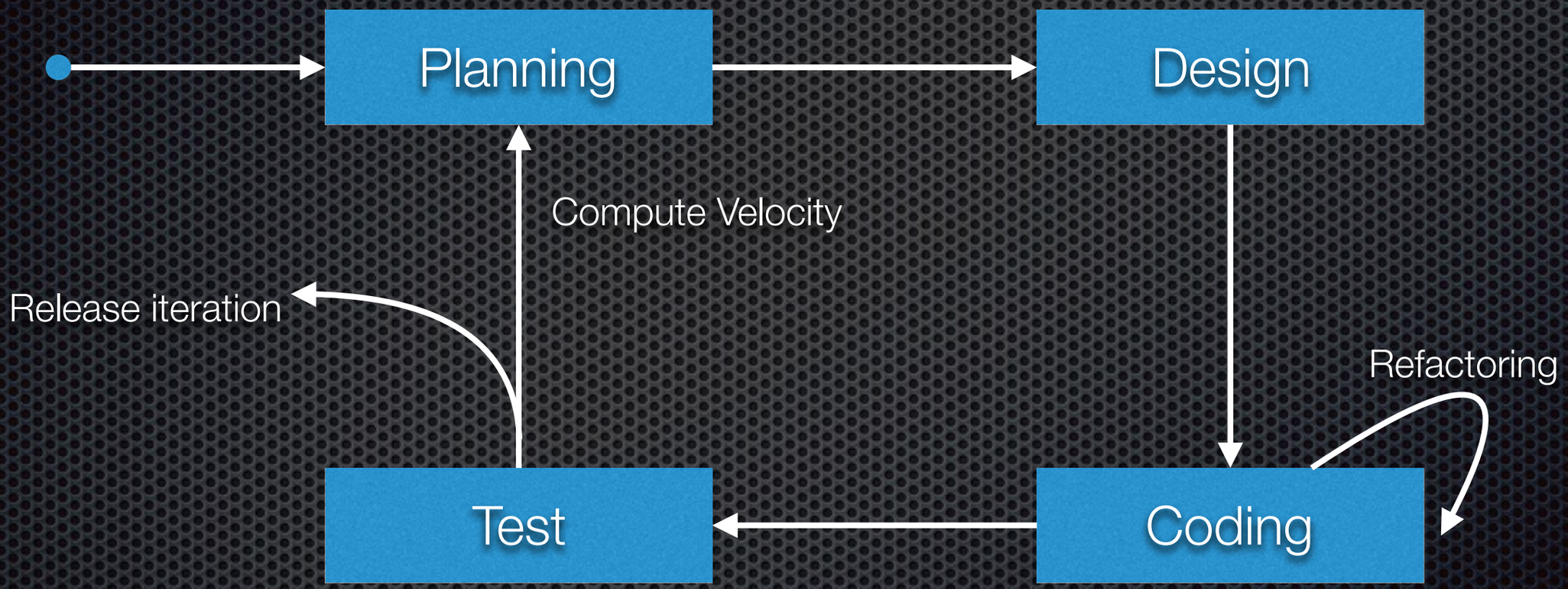
Extreme Programming (XP)

- Defines 5 values:
    1. Communication
    2. Simplicity
    3. Feedback
    4. Courage
    5. Respect

XP Facets

- Customer as team member
- User story: user perspective of functional requirement
- Test-Driven Development
- Pair programming
- Minimal design
- Refactoring
- Spikes

The XP Process



velocity: average # of tasks completed per iteration

Industrial XP (IXP)

- XP for large companies/projects
- Involves lots more stakeholders in new tasks
  - Management
  - Legal
  - QA, etc.

IXP Enhancements

- Readiness assessment
- Project community
- Project chartering
- Test-driven management
- Retrospectives
- Continuous learning

XP Issues

- All-or-nothing commitment to XP (partial is bad)
- Customer as team member makes reqs volatile
- Work products are limited
- Minimal design

Adaptive Software Development (ASD)

- Software releases in increments
- Process activities
  - Speculation
  - Collaboration
  - Learning

Scrum

- Agile method delivering highest business value first
- Stakeholders (e.g., clients) assign value to reqs.
- Scrum team selects subset of work tasks for next iteration (**sprint**)
  - Select based on biz value, dependency, finish time
- Sprint length: 2 to 4 weeks
- Lots of meetings for task assignment, progress updates, problem resolution, brainstorming (planning, standups, reviews)

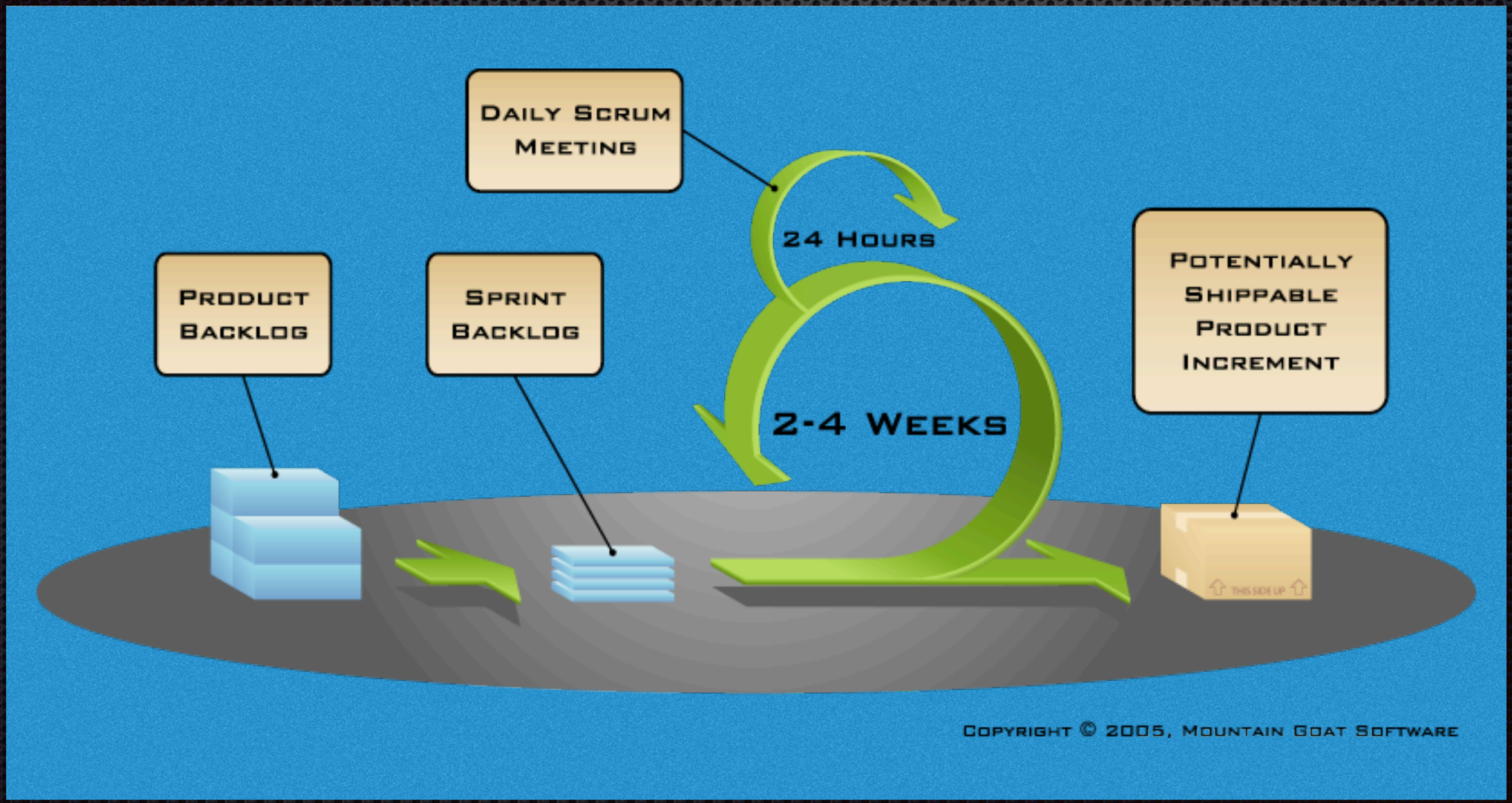
Scrum Backlogs

- **Product backlog** is a master list of things to do
  - Items have 2 extra descriptors: business value and time to finish
- **Sprint backlog** is a subset of product backlog items for that sprint
  - Items selected based on biz value, how long items will take, and how much team feels it can do that sprint (a **velocity** calculated from previous finished sprint items)
  - Items not finished return to Product backlog
- Product backlog can change during a sprint but the Sprint backlog **SHOULD NOT** be changed

Scrum-specific Roles

- **Product Owner (PO)**: voice of the customer, manages product backlog and user stories
- **User story**: a free text, story-like, step-by-step description of a functional requirement from an end-user perspective (less formal than a use case detail)
- **Scrum Master (SM)**: enforcer of scrum rules (meeting goals and time limits, no distractions during sprint)

The Scrum Process



Building a Product Backlog

- Make a set of all **items** that need to be done to deliver the software (maintained by Product Owner)
  - e.g., features, bugs, tool research, reqs. meetings
    - Items can be user stories, use cases, free text,... anything that is understandable and useful
- All items have a score for business value, dependencies, time to finish (can also have risk)
  - PO assigns biz value, team assigns time/risk

Building a Sprint Backlog

- Set of items for the next sprint
- Select highest biz. value items first (and their dependencies)
  - Keep picking until the team feels can't do any more
- Items can be broken into smaller tasks (e.g., UI task, DB task, test case creation task, testing task, etc.)
- Team members pick tasks to work on (self-organizing)

Dynamic Systems Development Method (DSDM)

- Starts with feasibility study
- Prototype each iteration's functionality
- Determine if prototype adds value
  - If yes, "operationalize" it
- Focus on delivering 80% of product quickly (20% time)
  - Plan to finish it once it is being used

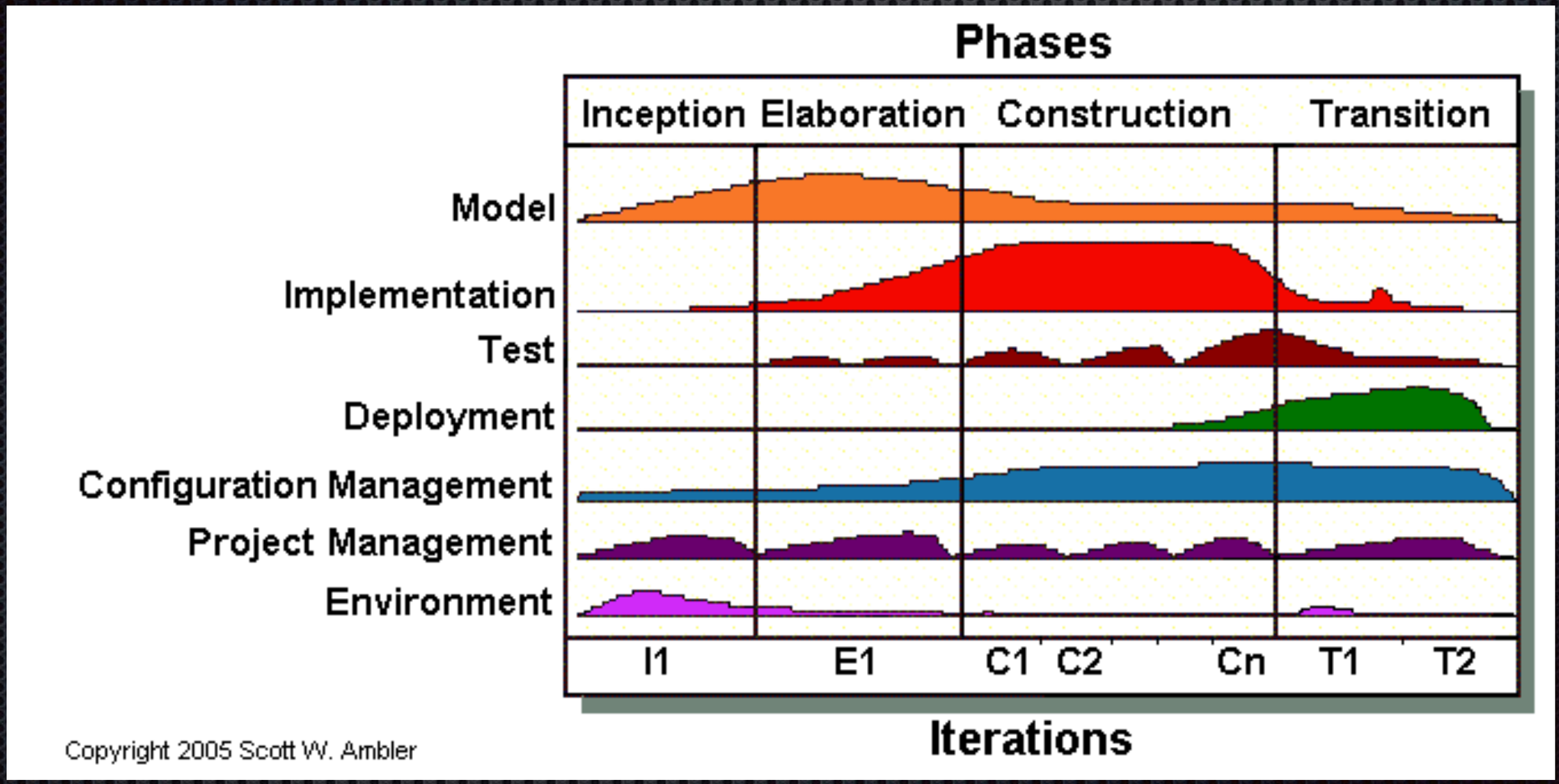
Agile Unified Process

- Agile version of the Rational Unified Process (a heavier, object-oriented, iterative process hybrid)
- Same linear activities:
  - **Inception**: identify scope, architecture, budget, and signoff
  - **Elaboration**: prove the architecture works
  - **Construction**: build working software in highest-priority first iterations
  - **Transition**: validate and deploy system to production

AUP

- Iterate in each activity:
  - **Model** : analyze and understand the goals of the activity
  - **Implement** : design and build according to models
  - **Test** : make sure what you built is correct and not defective
  - **Deploy** : deliver working product
  - **Manage** : handle req. changes, check risk, team support infrastructure

AUP



Summary

- All deal with the same basic actions (requirements, design, etc.)
  - But arrange (and call) actions differently
- **All accept change as part of the process**
- Remember: agile methods are tools, not a silver bullet
  - Be an engineer, not a zealot

Extra Credit Opportunity

- Complete a survey involving privacy and wearable devices
  - I don’t know how long it will take
  - Email me if you want to participate with a good contact email address
  - Must be completed within next 2 weeks
  - Worth 3 points on your final grade