CS 361
Software Engineering I

Agile

A Catalog of Some Processes

- Waterfall
  - Traditional
  - With prototyping
- Spiral
- Agile
  - Dynamic Systems Development Method (DSDM)
  - Scrum
  - Crystal
  - eXtreme Programming (XP)

Contrasting Processes

	Waterfall	Spiral	Agile
Emphasizes:	-Simplicity -Traceability	-Risk management -Exploring alternatives	-Flexibility -Immediacy
Weakness:	Requirement/design mistakes can be costly	Exploring alternatives can be costly	Continual rework can be costly
Style:	-Highly controlled -High ceremony	-Moderately controlled -Moderate ceremony	-Rapid & organic -Low ceremony

Some definitions

-"traceability": relationships between requirements and system elements are documented

-"immediacy": getting some sort of working system to the customer as fast as possible

-"rework": redesigning the architecture and/or refactoring the program code

-"controlled": conformance to process is highly valued, even if it slows a project down

-"ceremony": how much analysis, documentation, and planning is involved

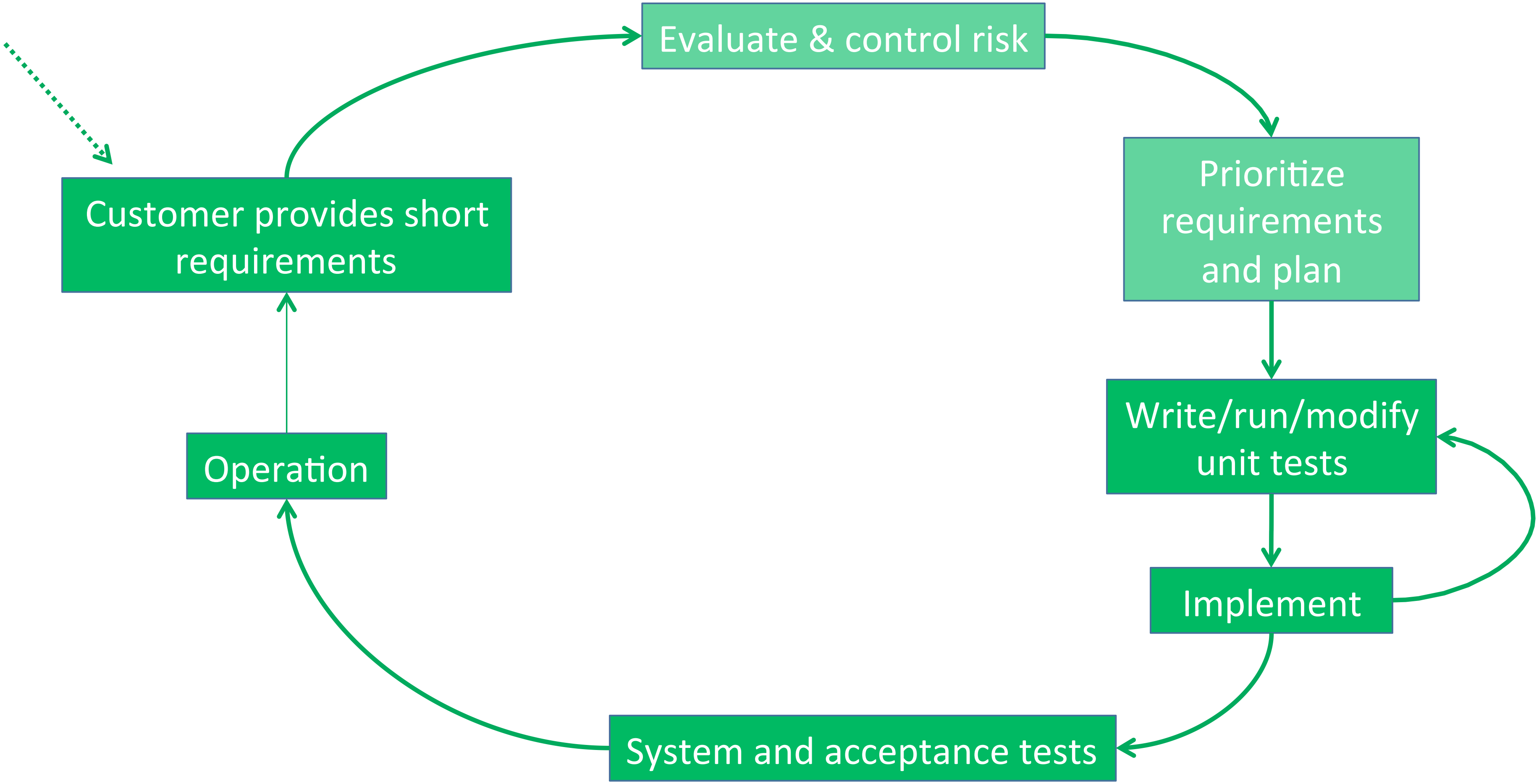
Choosing a Process

- *Waterfall* is often a good choice for small systems whose requirements can be fully understood before any design or coding.
- *Spiral* is often a good choice for larger systems with vague requirements and many alternatives for designing and coding.
- *Agile* is often a good choice for systems where you can rapidly create something small but useful, and then expand from there.

Agile Manifesto

- **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 Processes


Evaluate & control risk
Prioritize requirements and plan
Write/run/modify unit tests
Implement
System and acceptance tests
Operation
Customer provides short requirements

Iterations

- Purpose
  - Iterative development gives you a few "oh drat"s instead of one big OMG at the end.
- Timing
  - Sprint: 1 month
  - XP: 1-2 weeks
- Grouping
  - Iterations can be grouped into releases… not every iteration necessarily results in a new product release
- Sub-dividing
  - Each iteration has "micro-iterations" inside of it, where your team tries to complete some stories and communicates progress back to the customer, potentially refining the iteration's goals.

Specific Agile Processes

- DSDM
  - (briefly, as a point of comparison)

- XP
  - (in more detail, since this is what you' ll be using for the next few weeks)

Principles of DSDM

- Active **user involvement** is imperative.
- The team must be **empowered** to make decisions.
- Fitness for **business purpose** is the essential criterion for acceptance of deliverables.
- Requirements are specified at a **high level**.
- The focus is on **frequent delivery** of products.
- **Iterative and incremental** development is necessary to converge on a solution.
- All changes during development are **reversible**.
- **Testing is integrated** throughout the life cycle.
- **Collaboration** and cooperation is essential.

Key Practices of DSDM

- Ambassador Users & Facilitated Workshops
  - Users on-call & users en-masse (respectively)
- Stages of iterations:
  - Pre-project exploratory phase (kick around ideas)
  - Feasibility study (explore if ideas are do-able)
  - Business study (explore if ideas are worth doing)
  - Model, design, implement (a “timebox” of work)
  - Post-project phase (what went well, what didn’t?)

Principles of XP

- **Communication** – it is good to talk with customer and between developers
- **Simplicity** – keep it simple and grow the system and models when required
- **Feedback** – let users provide feedback early and often
- **Courage** – speak the truth, with respect

Practices of XP

- Whole team
- Metaphor
- The planning game
- Simple design
- Small releases
- Customer tests
- Pair programming

- Test-driven development
- Design improvement
- Collective code ownership
- Continuous integration
- Sustainable pace
- Coding standards

XP Practices: Role of the Customer

- Whole team
  - The customer is part of the team
- Customer tests
  - The customer participates in testing

XP Practices: Role of Reality

- The planning game
  - Be realistic about meeting customer needs
- Small releases
  - Meet customer needs in small increments
- Sustainable pace
  - No all-nighters, no superheroes

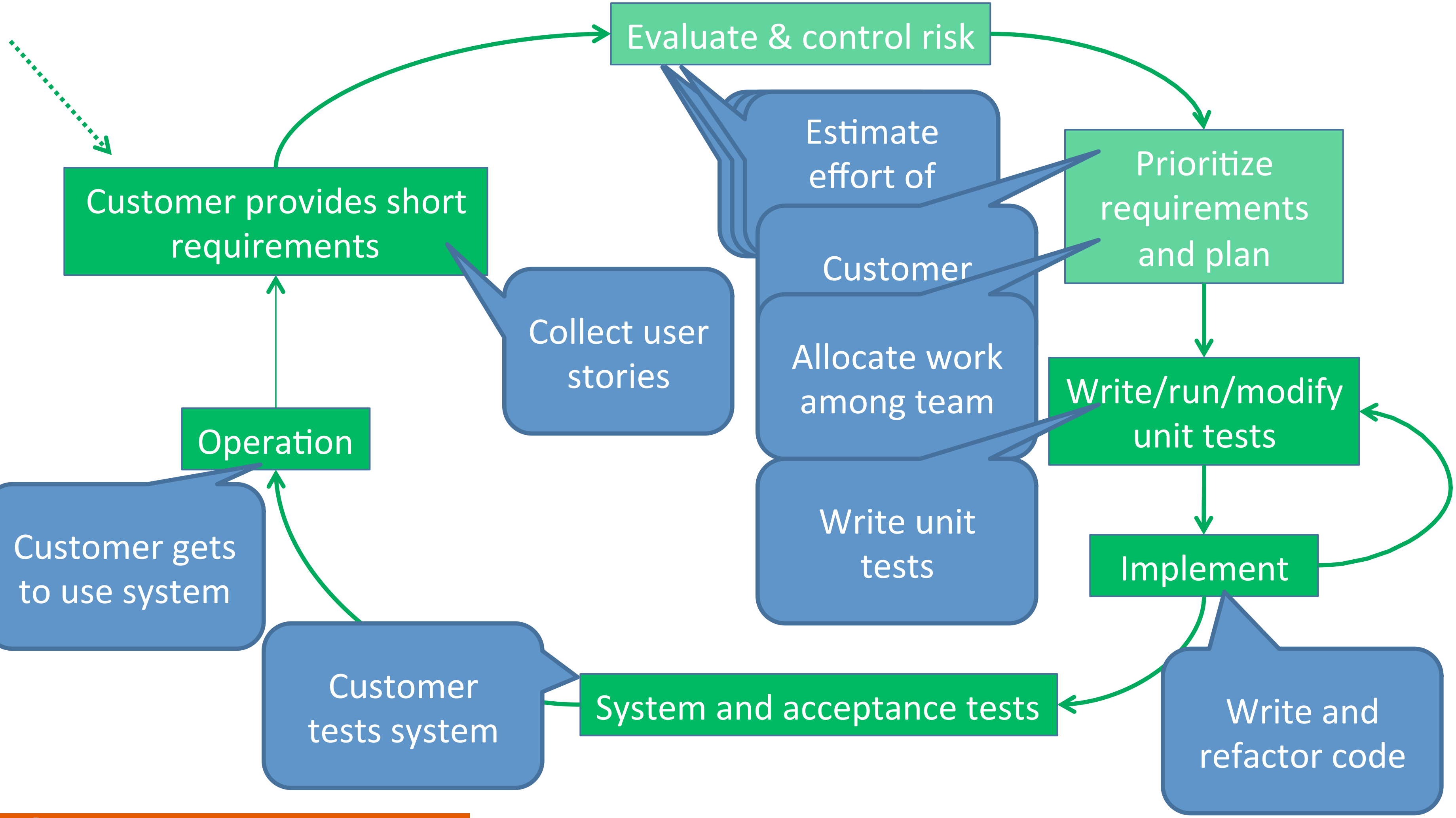
XP Practices: Role of Design

- Simple design
  - Simple models, simple architecture, simple code
- Design improvement
  - Refactor as needed
- Metaphor
  - Design around a coherent idea
- Continuous integration
  - Regularly check to see if the system is on track

XP Practices: Role of Teamwork

- Pair programming
  - All code is written with a “co-pilot”
- Test-driven development
  - Write tests first, then write code
- Collective code ownership
  - A big ego… one that includes the team!
- Coding standards
  - Pick a format, use it, and move on

XP Process in Detail



Concerns About XP

- Constant refactoring can be expensive
- XP can degrade into a hacker’s paradise
- Pair programming can take extra effort
- Programmers don’t always specialize
- Knowledge lives in heads, not on paper
- XP is not very standardized

Lessons from DSDM & XP

- Learn from…
  - Users (DSDM)
  - Customers (XP)
- Design based on…
  - Business value (DSDM)
  - Customer direction (XP)
- Requirements should be…
  - High-level (DSDM)
  - Succinct (XP)
- Engineers must demonstrate…
  - Empowerment (DSDM)
  - Courage (XP)