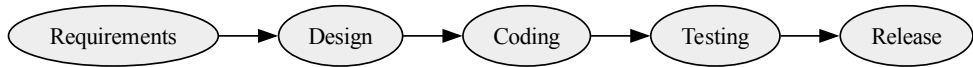# CSE 3902: Agile Development

Justin Holewinski

The Ohio State University

Traditional "Waterfall" Development

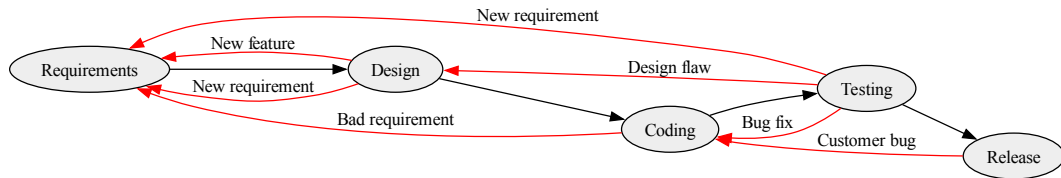Requirements → Design → Coding → Testing → Release

Sequence of steps representing a software project

- Document all requirements
- Create a design that meets all requirements, including test plans
- Code software components according to design
- Test product according to established test plan

Traditional "Waterfall" Development

What about the real world?

What is Agile?

agile: *adjective*

1. able to move quickly and easily.
2. relating to or denoting a method of project management, used especially for software development, that is characterized by the division of tasks into short phases of work and frequent reassessment and adaptation of plans.
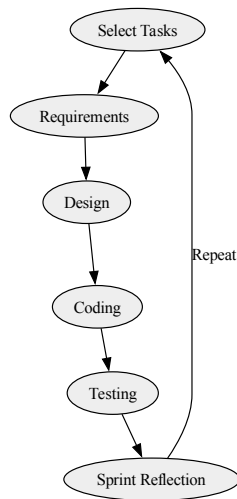
[source: https://languages.oup.com/google-dictionary-en]

What is Agile?

Maintain a *backlog* of *tasks*
- All features are broken down into *tasks*
- Each task may have sub-tasks, building a hierarchy
- The backlog is the master list of everything that needs to be done
  - Features
  - Bug fixes
  - Documentation
  - Investigations

Break down time into small units, called *sprints*
- Decide which tasks to accomplish during a sprint
- Gather requirements for sprint tasks
- Design software components for sprint tasks
- Code and test
- Reflect on the sprint before moving to next sprint

Select Tasks → Requirements → Design → Coding → Testing → Sprint Reflection → Repeat → Select Tasks

Why Agile?

Easier to adapt to changing requirements

- Requirements *will* change; embrace it
- Breaking a project into sprints give you more ability to adjust

Align internal releases to sprints

- End of each sprint should leave you with a working product
- It may be missing features, but it is *usable* and *testable*

Agile for CSE 3902

Each sprint is approximately three weeks

- Give or take a couple of days due to holidays

In each sprint

- Read over sprint requirements (given to you)
- Decide as a team how to break up the work among members
- Design the necessary software components, or changes to existing components
  - Remember to re-use as much as possible!
- Implement the designs
  - May be done individually or as small sub-groups
  - Keep the big picture in mind!
- Test!