Topic 2 - Process Models

COMPSCI 320 - Perretta

Slides loosely adapted from Gordon Anderson

One-Slide Summary

- Software engineering processes help us **manage risk, uncertainty, and constraints** to maximize our chances of delivering high-quality solutions
- **Process models** are blueprints for how to **divide work into smaller tasks** and **allocate time and resources** to completing those tasks
- Process models have **tradeoffs** that may make one model a better fit for an organization/team/project than another
  - We'll give a brief description of a few models, then focus on aspects of **agile** methodology

Software Engineering Activities

Process models guide **when** and **how thoroughly** we do each of these:

- Analysis
  - Gather info about problem
- Specification
  - What does the system have to do? Within what constraints?
- Design
  - What pieces does the system need to conform to the spec?

Software Engineering Activities

Process models guide **when** and **how thoroughly** we do each of these:

- Implementation
  - Build the system
- Testing
  - Does the system work as it should?
- Maintenance
  - Keep the system functioning properly

Model 1: Waterfall

Each step completed and vaildated before moving on.

- When might this be necessary?

Trade-offs:

- Difficult to go back to prior steps
- Each stage (theoretically) has complete information to work from

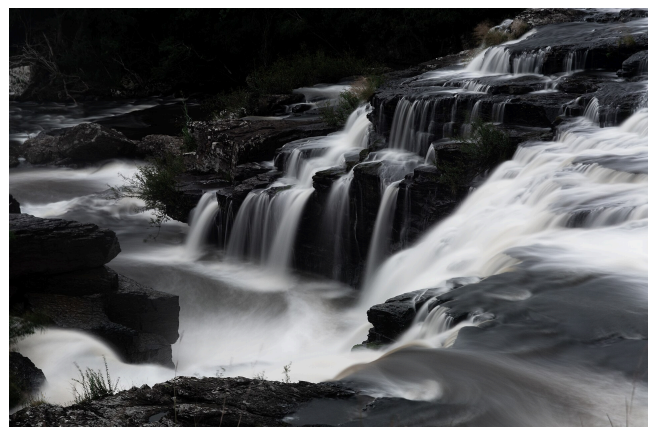


Photo by MartinKlippel, Wikimedia Commons, CC-BY-SA 4.0

Model 2: Spiral

Focuses on incremental progress towards delivery.

Trade-offs:

- Addresses bigger risks **early**
- Analysis & assessment each iteration
- Assessing risk is challenging
- High planning overhead

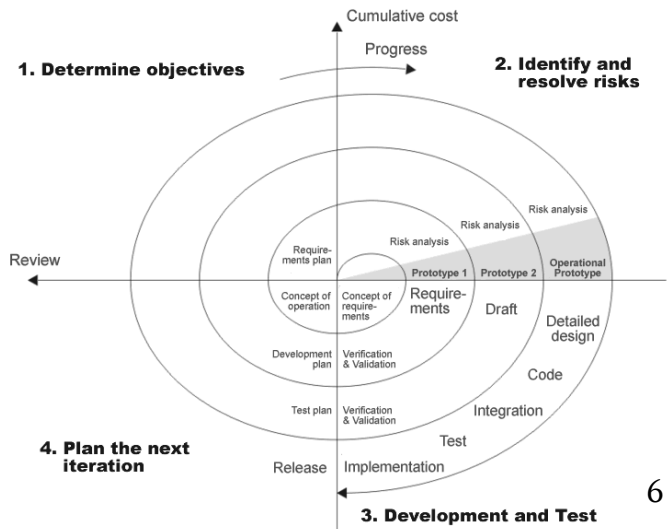


Diagram by Conny, Wikimedia Commons, CC-BY-SA 3.0 Unported

"Big Bang"

Release minimum possible prototype as soon as possible.

- What might the trade-offs of this be?

Trade-offs:

- Get your idea to market quickly
- Relies on users tolerating bugs & providing feedback
- Doesn't work for high-stakes software
  - 2020 app that delayed Iowa Caucus results

Agile

Emphasizes flexibility, customer involvement, utilizing the skills of developers.

Some key aspects:

- Developers involved in planning, design, development, & testing
  - These activities happen continuously
- Project organized into time-based intervals called **sprints**
- Requirements continuously **re-prioritized** based on customer/user input
- Pair programming (two programmers work on task together)

Agile: Scrum

Agile process by which work is organized.

We'll go through a high-level "how-to" of some key aspects.
See the reading for more details.

Scrum How-To: Product Backlog

Make a list of **all work** that needs to be done.

Manager & stakeholders assign **priority** to each item.

Developers **estimate the effort** required for each item.

Scrum How-To: Sprint Planning

Manager & developer discuss **who** should work on **what items** from the product backlog.

Chosen tasks recorded in the **sprint backlog**.

Notes:

- Pair programming encouraged; takes longer, but higher quality work.
- If a product backlog item is too big for one sprint, break it up into smaller tasks!
- Some interpretations have tasks assigned to specific people.

  Scrum prescribes having developers self-organize who works on what.

Daily Scrum/"Stand-Up" Meeting

Meeting <= **15 minutes**.

Team members report progress and any issues encountered.

- Keeps team members up to date
- Team knows there's a problem -> easier to address the problem
- Important to keep the meetings **short**!

Sprint Conclusion

Sprint retrospective:

- Team presents progress to stakeholders, gets feedback
- Team reflects on how the sprint went

Any tasks not completed **return to the product backlog**.

Which tasks to focus on is reassessed at the **next sprint planning** meeting.

Scrum Simulation (Scrumulation?)

Scenario: We're tasked with clearing snow from all walkways on campus.

Activity Steps

1. Build product backlog
   - Assign priority (customer & manager)
   - Estimate effort (developers)
2. Sprint planning
3. Stand-up meeting
4. Sprint retrospective

Conclusion

When choosing a process model, consider:

- The project scope and reliability requirements.
- Amount of unknowns & level of risk.
- Quality / cost control requirements.
- Visibility of progress (deliverables often or at the end?)
- Level of customer involvement and feedback desired.

The best model is the one that fits the project, company culture, skill level of managers and developers, customer, etc.