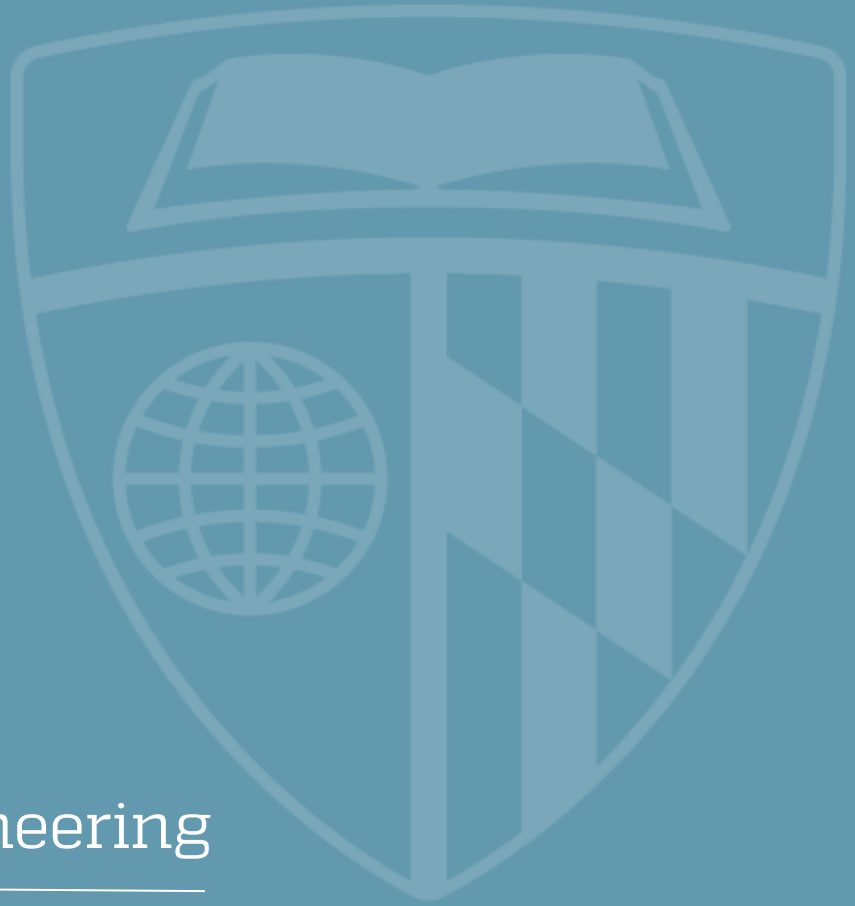


JOHNS HOPKINS
UNIVERSITY

EN.601.421 / EN.601.621

Object Oriented Software Engineering

Course Staff and Meetings

- **Instructor: Dr. Ali Darvish** (darvish@jhu.edu, www.cs.jhu.edu/~darvish/)
  Head TA: Steven Tan (wtan12@jhu.edu)
- **Meetings:** Tuesday and Thursday 1:30pm-2:45pm
- **Course Website:** https://jhu-oose.github.io/cs421_f21/
- **Piazza:** http://piazza.com/jhu/fall2021/en601421621
- **Gradescope:** https://www.gradescope.com/courses/297155
- **Github Organization:** https://github.com/jhu-oose/

Plan for Today

- Building Software
- OOSE Goal & Audience
- Logistics

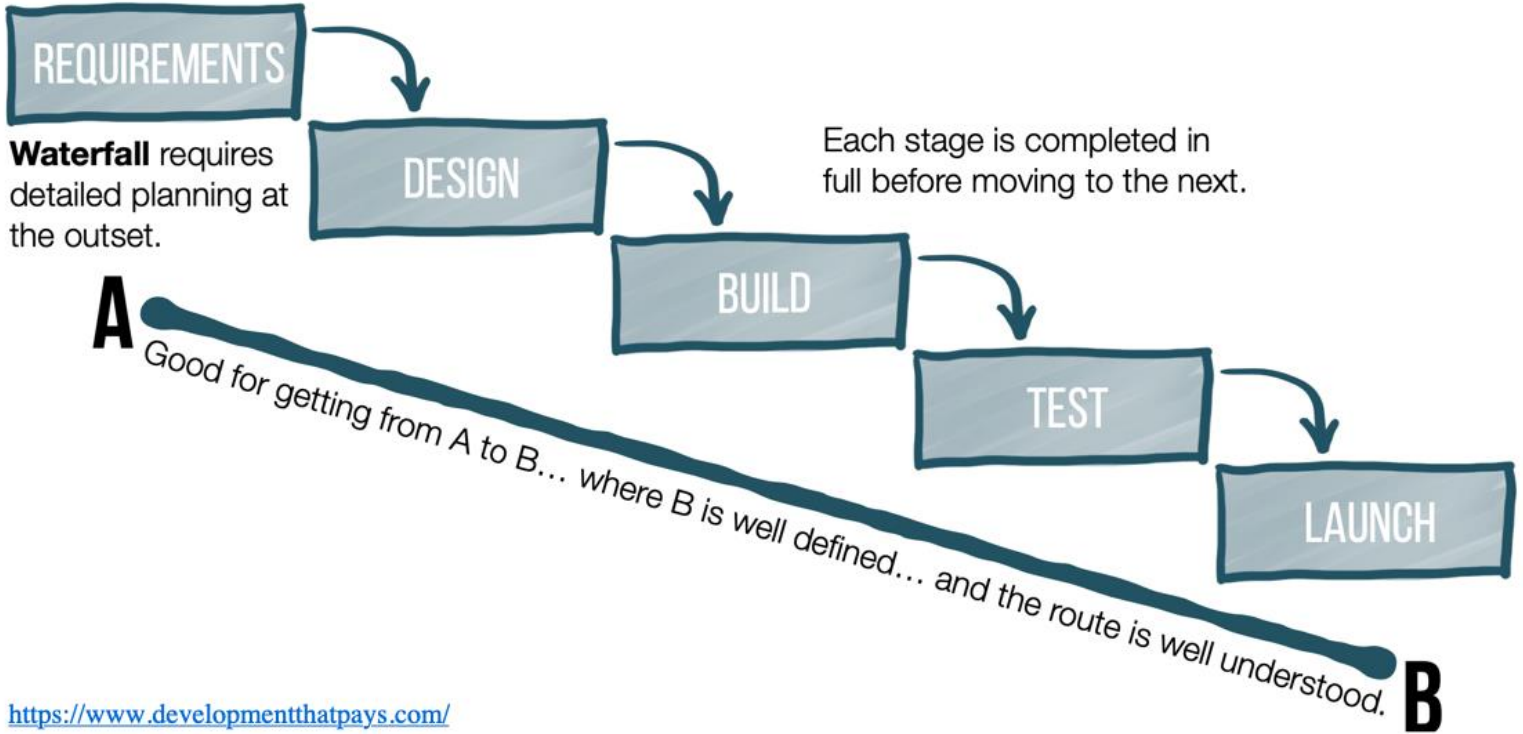
Software requirement specification

- Also known as stakeholder requirements specification
- A description of a software system to be developed:
  - All “stakeholders” involved need to be present: management, developers, customers, etc.
- SRS template

Software Process Models

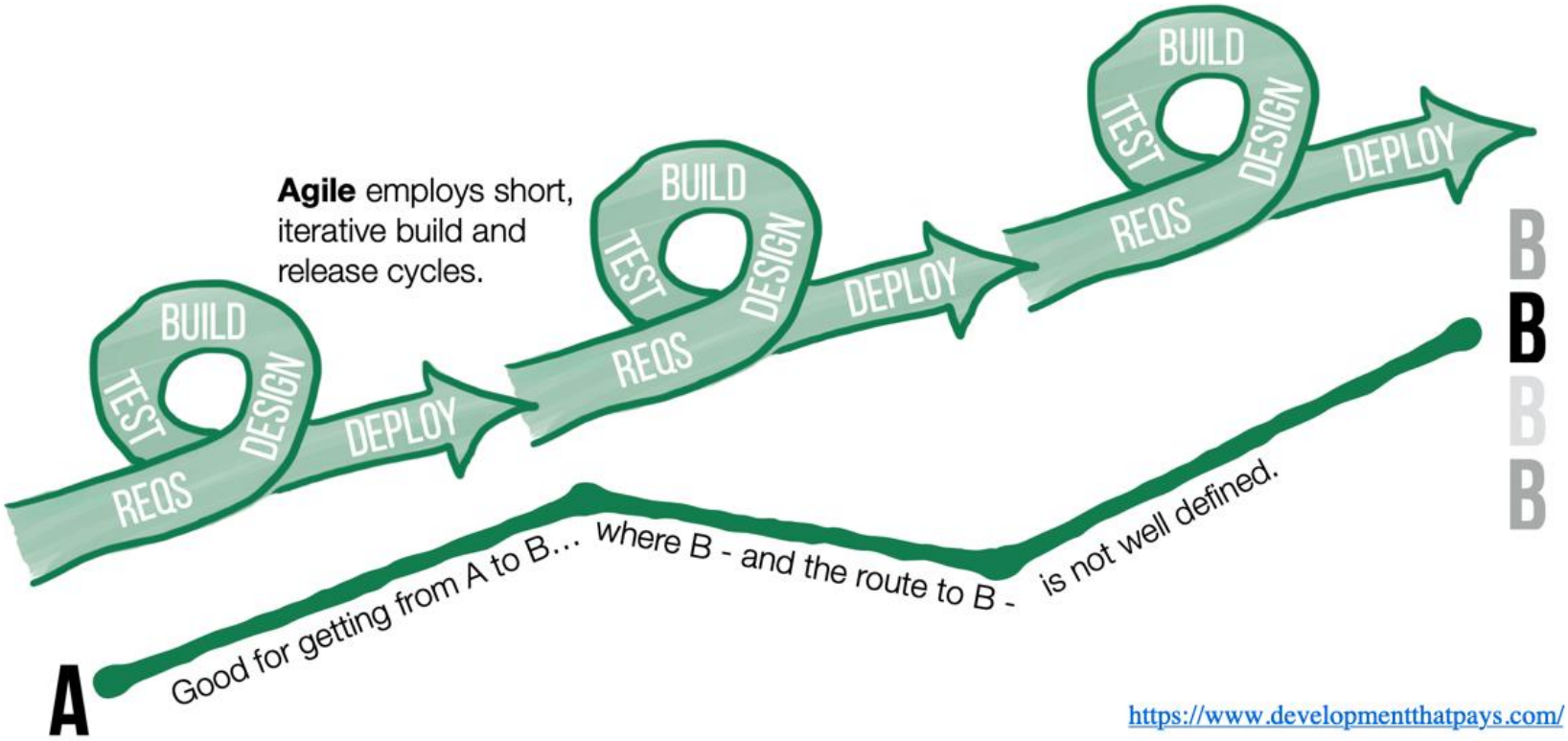
- A set of related activities that leads to the production of a software product
  - **Plan-driven**: process activities are planned up front, and progress is measured against this plan
  - **Agile**: planning is incremental, and it is easier to change the process in accordance with changing customer requirements

Waterfall



https://www.developmentthatpays.com/

Agile


Agile employs short, iterative build and release cycles.
BUILD
TEST
DESIGN
REQS
DEPLOY
A
Good for getting from A to B… where B - and the route to B - is not well defined.
B
B
B
B
https://www.developmentthatpays.com/

OOSE Model

- (mostly) Agile
- **A total of 5 Iterations (each iteration is 2 weeks)**
  - Start with a first pass at requirements and design
  - Tackle the most-key features (i.e., basic functionality) first
  - Refine the requirements and design
  - Implement more features, continue to refine requirements/design more
  - Provide continual releases of the running application

User Story

- A widely used way to capture requirements
- Expressed in role-goal-benefit form:
  - *As a ‹type of user›, I want ‹some goal or objective› so that ‹benefit/value›*

JBApp

- The goal of this project is to create a web application where we can view, post, and search among job posts. The app should streamline the process of adding new job posts for employers. Also, it should allow potential employees to view all job listings and be able to search among them using different filters (e.g., search keywords, job field, job type, pay range, etc.). Ideally, A "sign up/in" functionality would be nice (not a "must have") where only signed-in users can apply for jobs.

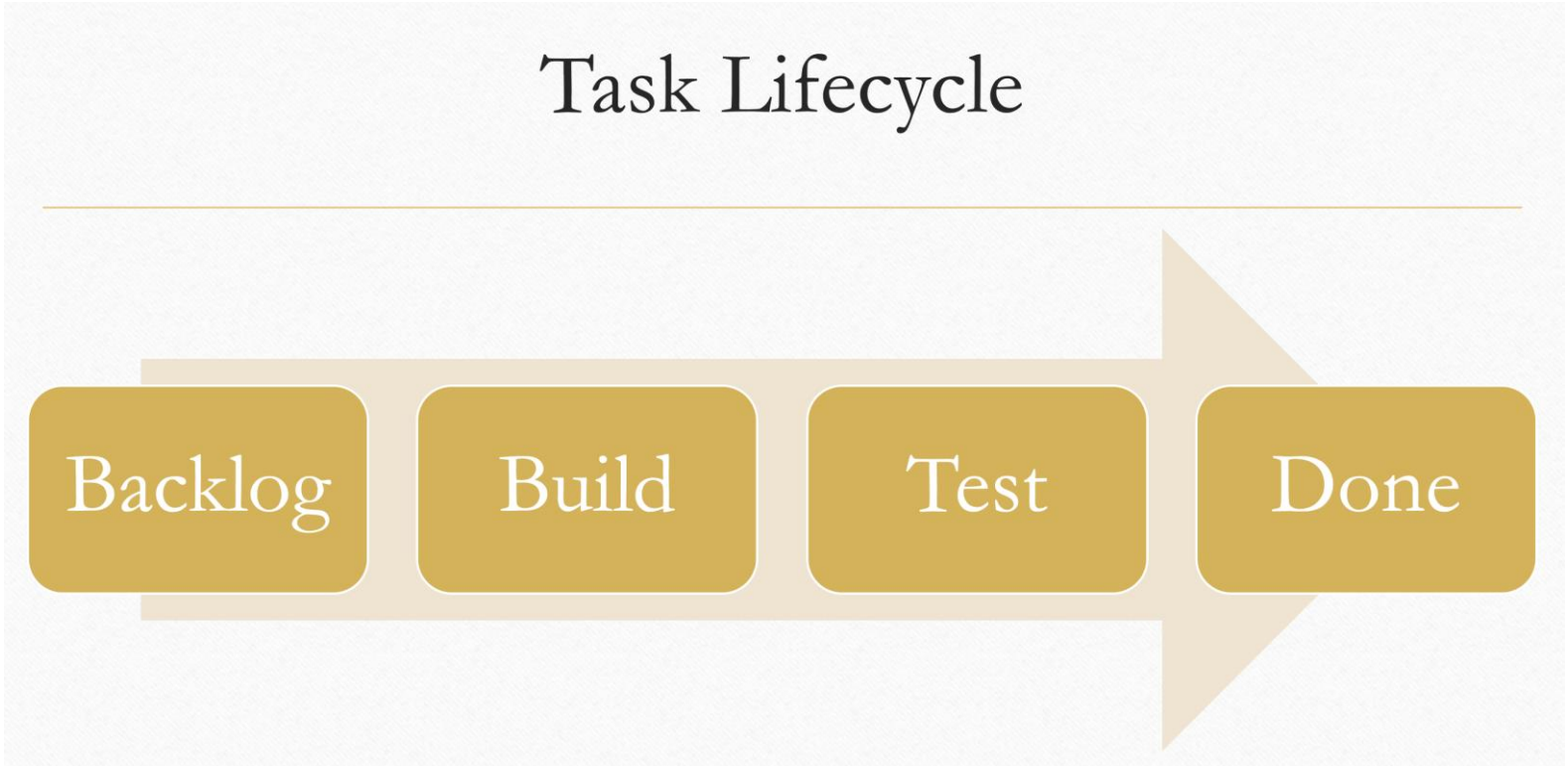
User Story Characteristics

- Keep it short, clear, and to the point
- A user story only captures one functionality requirement
- User stories typically capture “functional” requirements only
- Do not get caught up writing too many nice to haves

Project Planning

- Provide an estimate for each user story
- Be aware of *Planning Fallacy*
- Focus on core functionalities first (i.e., must haves)
  - Aim low and hit!
  - As you work through the iterations, add/remove/refine user stories
- **End goal**:
  - *Alpha* release: at the end of iteration 4
  - *Beta* Release: at the end of iteration 5

Project Backlog



Build Software that is needed

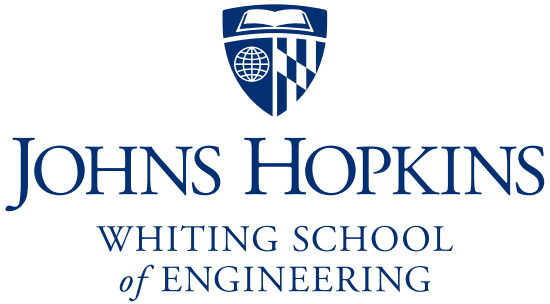
- Solves a real problem (not some hypothetical non-existent need)
- Goes beyond CRUD (more on this later)
- **Strongly Recommended**: conforms to Client-Server Architecture (more on this later)
- **Ideally**: has novelty to it to make it stand out from similar products

OOSE Goal and Audience

- A course aimed to make you a better software engineer!
- Designed on the premise of:
  - *Self-efficacy*
  - *Self-directed learning*
- Time commitment:
  - (roughly) 7 or 8 hours a week a person

Logistics

- Schedule
- Syllabus
- Toolbox
- Group formation starts next week!
- Github Survey form


JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING