

Module 1

Introduction to SaaS, Agile, Cloud Computing

CS 169A: Software Engineering

1 Overview

Welcome to the world of software engineering! As you will soon find out, becoming a capable software engineer involves much more than knowing how to write algorithms. As you'll see across the course of the semester, an apt software engineer has not only technical proficiency, but also personable, collaborative, and decisive, among many valuable characteristics.

In module 1, we introduce the concepts of Software as a Service, along with different workflows suited for different kinds of projects that allow teams to coordinate appropriately. In this worksheet, we'll ask you to recall and apply some of these concepts from the first chapter.

Do note that some of these questions don't have any one specific correct answer! Get creative! There's a lot of examples that capture and embody the same concepts. Feel free to write your answers or discuss with classmates!

2 Workflow Concepts

In this chapter, we learned about a variety of different software development processes, the main ones being Plan and Document (Waterfall), Spiral, Rational Unified Process (RUP), and Agile. Answer the following questions to refresh your understanding and think about how to apply these concepts.

What are the differences between the Waterfall Process and the Agile Process?

Can you design and build hardware with the Agile Process?

Describe a recent software project you've been involved with, and whether you think it would be a good fit for Agile or not. Do your classmates agree?

3 Verification & Validation

Testing is an extremely important skill that is integral to writing correct, efficient software. In this class, we'll explore the concepts and tools behind writing tests that verify both technical and behavioral benchmarks. for now, recall that verification is for checking "Did you build the thing right?" (Did you meet the specification?). On the other hand, validation is concerned with "Did you build the right thing?" (Is this what the customer actually wants).

You're building the shopping-cart checkout portion of an e-commerce website. What would be an example of each of the following types of tests for the shopping cart?

Unit Tests

Module or Functional Tests

Integration Tests

System or Acceptance Tests

What are some formal methods of verification and what are their pros and cons?

4 Programmer Productivity

With the increasing plethora of available, open source software, there's absolutely no need to reinvent the wheel when it comes to finding tools that bolster the amount of work you're able to get done. Throughout this class, you'll be introduced to a variety of platforms, websites, and libraries that will expedite the development process. All of these tools will typically fit under one or more categories of productivity.

Define and describe an example of each of the following productivity concepts: Clarity via conciseness, synthesis, reuse, automation and tools.

5 Software as a Service

SaaS applications dominate the majority of software that we use in our daily lives. The main premise of SaaS is that software is run in a web browser instead of having to be downloaded and install on a local machine. An example of SaaS would be any Google App (i.e. Maps, Drive), all of which are meant to be accessed from your browser. On the other hand, you would download a tool like Adobe Illustrator to your local computer, which is a more traditional approach.

How does SaaS differ from traditional software? Think about what features might make it more advantageous to deliver the app in a browser rather than as a local installation.