Choosing the Correct Generic Software Process

Simon Codrington III

Grand Canyon University

SWE-520: Advanced Software Engineering Fundamentals

Dr. Wibbenmeyer

March 27, 2024

Pre-Requisite knowledge

Objective: Define different generic software processes/SDLC (software development life cycle)

* Waterfall Model
  + Overview
    - Emphasizes well defined and carefully planned requirements and use cases.
    - Each phase of the SDLC is executed sequentially.
    - Favors complex systems where planning and timing are important.
  + Phases
    - Requirement Analysis
      * Gather documentation, specifications, and use cases for software.
    - System Design
      * Develop high level technical designs and system requirements. Choose different technology stacks.
    - Implementation
      * Code and unit testing based on specifications.
    - Testing
      * Test the entire system to make sure requirements and use cases are met.
    - Deployment
      * Release the software to end users and provide monitoring and support.
* Incremental Development Model
  + Overview
    - Emphasizes making small deliverables iteratively with each deliverable adding new functionality or features.
    - Provides value to users quickly and adapts to unstable requirements.
    - “Relies on the availability of reusable components or systems.” (Sommerville, 2015, p. 32)
  + Phases
    - Requirements
      * Focuses on user need and features for current increment.
    - Development
      * Focuses on coding and unit testing.
    - Testing
      * Validation of met requirements for current increment.
    - Deployment
      * Release of increment, gathering of feedback from release to help guide next increment.
* Integration and Configuration
  + Overview
    - Emphasizes reusable components and the use of frameworks for integration of components
    - Often used where systems are needed for a particular purpose in a particular environment
  + Phases
    - Requirements
      * Does not need to be thorough, but general overview
    - Software Discovery and Evaluation
      * Search for technology stack that is most appropriate
    - Requirement refinement
      * Requirements are refined using information about reusable components in tech stack
    - Application System Configuration
      * If off shelf system/framework exists, it is configured for the system.
    - Component adaptation and Integration
      * If no of shelf system exists, we split into individual components throughout the stack

SDLC Suggestions

Objective: The following will offer a suggestion for the next four projects and will choose from the previously mentioned processes.

1. System to control anti-lock brake system
   1. Suggestion: Waterfall Method
      1. Argument
         1. The system that controls the anti-lock brake system is a hyper-specific function that should be easy to figure out in the planning stages. A system such as this should not need to be too dynamic in terms of specifications and requirements. This embedded system will absolutely be complex and timing in system is crucial. Planning and testing step by step is best.
      2. Ethical Consideration
         1. This process will be best for securing IP rights, which is crucial for building software for a patented device like a brake system. This also is good for long term plan for security against computer misuse and tampering of a crucial system.
2. Virtual reality system to support software maintenance
   1. Suggestion: Waterfall Method
      1. Argument
         1. There are not too many systems that are more complex than virtual reality. Even though something like a maintenance system needs to be able to solve problems that have not occurred yet, the system and technologies that are meant to be maintained should be. A linear path of implanting with another system is best.
      2. Ethical Consideration
         1. Confidentiality in a virtual environment is crucial and an entire phase would be needed to secure the system and the systems that it would interface with. Best suited with a SDLC that is more long-term oriented.
3. A university accounting system that replaces an existing system
   1. Suggestion: Integration and Configuration
      1. Argument
         1. A system such as this can be completed almost exclusively with certain frameworks out of the box. An accounting system would need multiple components that can be scoped out to function.
      2. Ethical Consideration
         1. Confidentiality and competence would be the major focus here. Frameworks require a great deal of expertise to work with.
4. An interactive travel planning system that helps users plan journeys with the lowest environmental impact
   1. Suggestion: Incremental Development Model
      1. Argument
         1. This product is intended to be a solution for clients to solve a specific problem. The client should work hand-in-hand with developers to make sure that needs are being met. Things like environmental impact can be difficult to define upfront.
      2. Ethical Consideration
         1. This process would help securing property rights and making sure legal standards are met. As requirements change, so will the compliance of laws. This is best checked on an iterative basis and not linearly.

*Citations*

OpenAI. (2024, March 26). GPT-3.5 (or ChatGPT). https://www.openai.com/

Sommerville, I (2015). *Software engineering* (10th ed.). Pearson.