Levi George

Sept 26, 2020

Dr. Venkata Inukollu

CS 36000: Homework-1

Homework 1

1. Agile SDLC Model
   1. Phases: Requirements, Design, Development, Testing, Deployment, and Review
   2. Examples
      1. Requirements: collecting information regarding what the software should do
      2. Design: Writing how the software should do what it will do
      3. Development: Coding the Design
      4. Testing: Ensuring that the code accomplishes what was listed in the requirements
      5. Deployment: Selling the software that has been tested
      6. Review: Checking for bugs while being used in real-time
   3. Pros: Little planning required, Flexible
   4. Cons: Hard to handle complex systems, high dependence on customer provided requirements
2. Verification and Validation
   1. Phases: Requirement Analysis, System Design, Architecture Design, Module Design, Coding, Unit testing, Integration testing, System Test, User-Acceptance Test
   2. Examples
      1. Requirement Analysis: collecting information regarding what the software should do
      2. System Design: writing how the system will work and be organized
      3. Architecture Design: describes the interface relationships, dependencies, and database tables.
      4. Module Design: The organization of smaller units of the system, can represent error messages, input/output for modules, and interface details
      5. Coding: The actual implementation is written in the programming language of choice
      6. Unit Test: Each module of the system is tested for accuracy
      7. Integration test: Ensures that each module designed can work together without crashing the system
      8. System Test: System is tested for proper functional and nonfunctional requirements are being met. Testing is also performed to ensure it will work in a real-world environment.
      9. User Acceptance Test: User runs these tests to ensure that the system runs properly in a production environment.
   3. Pros: More chances to prevent overhead, testing is done after each major stage
   4. Cons: Less flexibility, not suitable in situations where changes are likely to occur in requirements.
3. DevOps
   1. Phases: Plan, Develop, Deliver, Operate
   2. Examples
      1. Plan: establish requirements, functional and nonfunctional. Track bugs create backlogs.
      2. Develop: Coding, writing, testing, reviewing, and integrating libraries.
      3. Deliver: Deployment of software system
      4. Operate: Maintenance, monitoring, and fixing issues that arise from production lifetime.
   3. Pros: Quick to release software system into production, efficient
   4. Cons: Hard to transition quickly, Major mindset changes need to occur on teams.
4. Actors: Customers, Tech Support
5. Functional and Nonfunctional requirements
   1. Functional: Book tickets for trains, Customers can search for availability
   2. Non-Functional: Reservation system needs compatibility over devices and systems, Consistency (Customers can only buy tickets that are available).
6. Use Case Diagram – On Next Page

Diagram

Description automatically generated