Module: Capstone Project using Java
Qualification: Advanced Certificate In Web Development

LITHAN

IU 3. Requirements Specification & Phase Test Plan

By the end of this session, you will be able to understand Requirements Specification & Phase Test Plan

Contents

LITHAN

S. No.	Topic Description	Required / Optional
01	Requirements and specifications of a application/system	Required
02	Propose relevant tests for application/system	Required
03	Develop a phase test plan	Required

Software Requirements Specification

LITHAN

- Description of a software system or application to be developed.
- Establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function.
- Rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign.
- Provide a realistic basis for estimating
  - Product costs
  - Risks
  - Schedules.
- Used appropriately, software requirements specifications can help prevent software project failure.

Software Requirements Specification

LITHAN

- Software requirement is a functional or non-functional need to be implemented in the system.
- Basically software requirement is:
  - Functional
  - Non-functional

Functional	Non-functional
Functional means providing particular service to the user. For example, in context to banking application the functional requirement will be when customer selects "View Balance" they must be able to look at their latest account balance.	It can be a performance requirement. For example, a non-functional requirement is where every page of the system should be visible to the users within 5 seconds.

Software Requirements Specification

LITHAN

- There are 3 general Types of Requirements:

Business requirements	Architectural and Design requirements	System and Integration requirements
They are high-level requirements that are taken from the business case from the projects. For example, a mobile banking service system provides banking services to Southeast Asia. The business requirement that is decided for India is account summary and fund transfer while for China account summary and bill payment is decided as a business requirement	These requirements are more detailed than business requirements. It determines the overall design required to implement the business requirement.	At the lowest level, we have system and integration requirements. It is a detailed description of each and every requirement. It can be in the form of user stories which is really describing everyday business language. The requirements are in abundant details so that developers can begin coding.

Software Requirements Specification

LITHAN

- Sometimes for some project, you might not receive any requirements or documents to work with.
- But still, there are other sources of requirements that you can consider for the requirement or information so that you can base your software or test design on these requirements.
- Here are five steps you can follow to write an effective SRS document:
  - Create an Outline Or Use an SRS Template
  - Start with a Purpose
  - Give an overview of what you'll build
  - Detail your specific requirements
  - Get approval for the SRS

Propose Relevant Tests

LITHAN

- The software testing group is a service provider.
- Software testers provide valuable information and insights into the state of the system.
- Software testing enables making objective assessments regarding the degree of conformance of the system to stated requirements and specifications.
- This information contributes towards reducing the ambiguity about the system.
- For example, when deciding whether to release a product, the decision makers would need to know the state of the product including aspects such as the conformance of the product to requirements, the usability of the product, any known risks, the product's compliance to any applicable regulations.

Propose Relevant Tests

LITHAN

- Software Testing has different goals and objectives. The major objectives of Software testing are as follows:
  - Finding defects which may get created by the programmer while developing the software.
  - Gaining confidence in and providing information about the level of quality.
  - To prevent defects.
  - To make sure that the end result meets the business and user requirements.
  - To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
  - To gain the confidence of the customers by providing them a quality product. Without the human element, it's difficult to get insight into visual aspects of your UI like colors, font, sizes, contrast or button sizes.
  - The tools to run automation testing can be expensive, which may increase the cost of the testing project.
  - Automation testing tool is not yet full proof. Every automation tool has its limitations which reduces the scope of automation.
  - Debugging the test script is another major issue in automated testing. Test maintenance is costly.

Propose relevant tests for applications or systems to achieve the testing objectives

LITHAN

- Software testing helps in finalizing the software application or product against business and user requirements. It is very important to have good test coverage in order to test the software application completely and make it sure that it's performing well and as per the specifications.

- While determining the test coverage the test cases should be designed well with maximum possibilities of finding the errors or bugs. The test cases should be very effective. This objective can be measured by the number of defects reported per test cases. Higher the number of the defects reported the more effective are the test cases.

- Software testing makes sure that the testing is being done properly and hence the system is ready for use.

Propose relevant tests for applications or systems to achieve the testing objectives

LITHAN

- Good coverage means:
  - That the testing has been done to cover the various areas like functionality of the application, compatibility of the application with the OS, hardware and different types of browsers, performance testing to test the performance of the application and load testing to make sure that the system is reliable and should not crash or there should not be any blocking issues.
  - It also determines that the application can be deployed easily to the machine and without any resistance. Hence the application is easy to install, learn and use.

- In addition, testing validates that the system being developed is what the user needs. In essence, validation is performed to ensure that we are building the right system. Apart from helping make decisions, the information from software testing helps with risk management.

Propose relevant tests for applications or systems to achieve the testing objectives

LITHAN

- Quality improvements help the organization to reduce post release costs of support and service, while generating customer good will that could translate into greater revenue opportunities.

- Also, in situations where products need to ensure compliance with regulatory requirements, software testing can safeguard the organization from legal liabilities by verifying compliance.

Develop a phase Test Plan

LITHAN

- Test plans outline the process of testing the functionality of software. A test plan details each step taken to achieve a certain result and states the objective of each action.
- The plan also highlights the projected resources, risks, and personnel involved in the test.
- You should use a test plan if you are seeking to eliminate bugs and other errors in your software before it becomes available to customers. Follow the steps below to create a test plan.
- Steps
  - Preparing the Test Plan
    - ✓ **Know the basics -** What you put in your test plan depends largely on the complexity of the software you're planning to test. However, there are three basic sections that should always be included in a test plan: Test Coverage, Test Methods, and Test Responsibilities.
      - Test coverage defines what you will be testing and what you will not.
      - Test methods define how you will be testing each part defined in the "coverage" section.
      - Test responsibilities assign tasks and responsibilities to different parties. This section should also include what data each party will record and how it will be stored and reported.

Develop a phase Test Plan

LITHAN

- Steps
  - Preparing the Test Plan (*continuation)*
    - ✓ **Familiarize yourself with necessary IEEE standards documents -** The Institute of Electrical and Electronics Engineers (IEEE) publishes international standards for testing and documenting software and system development. To hold your test plan to the highest standard, consult with the IEEE publications below:
      - 29119-1-2013, Software and Systems Engineering - Software Testing - Part 1: Concepts and Definitions
      - 29119-2-2013, Software and Systems Engineering - Software Testing - Part 2: Test Processes
      - 29119-3-2013, Software and Systems Engineering - Software Testing - Part 3: Test Documentation
      - 829-2008, IEEE Standard for Software and System Test Documentation
      - 1008-1987 - IEEE Standard for Software Unit Testing

    - ✓ **Consult a template** -You can find templates for test plans online. The best source for templates is the IEEE library, but access does cost a fee.
      - The North Carolina Office of Information Technology Services offers an annotated test plan template, based on IEEE 829 standards, here.
      - Dublin City University also offers a free test plan template, based on IEEE 829 standards.

Develop a phase Test Plan

LITHAN

- Steps
  - Writing the Test Plan (*continuation)*
    - ✓ **Write the introduction -** Your introduction functions as the "executive summary" of the test plan: its goals, its scope, and its schedule. This should be kept brief, as you will go into further detail in subsequent sections of the test plan.
      - Your goals and scope statements should define, in general terms, the methods that will be used in the testing process and the projected results. The scope statement should also include the most critical performance measures, as well as a list of what the test plan will not address, and why.
      - A schedule details the increments of time in which each phase of the test will be completed.
      - Related documents include any peripheral material that is relevant to the current project, such as lists of specifications.
    - ✓ **Define your objectives.** Your test plan should clearly define *what* you will test and *why* you will test it. These should always be based on industry standards

Develop a phase Test Plan

LITHAN

- Steps
  - Writing the Test Plan (*continuation)*
    - ✓ **Define your objectives** (*continuation)*
      - Determine what the scope of the test is. What scenarios will be tested?
      - Determine what is out of scope for the test. What scenarios will not be tested?
      - Common scenarios include Module Testing, Integration Testing, Systems/Acceptance Testing, and Beta Testing.

    - ✓ **Write a section on required resources -** This section describes all of the resources needed to complete the testing, including hardware, software, testing tools, and staff.
      - When accounting for your staff, make sure to detail the responsibilities required of each member and the training needed to execute those responsibilities.
      - Make sure to document the exact specifications of hardware and software.

Develop a phase Test Plan

LITHAN

- Steps
  - Writing the Test Plan (*continuation)*
    - ✓ **Write a section on risks and dependencies** - Detail all the factors that your project depends on and the risks involved in each step. The level of acceptable risk in your project will help determine what you will and will not test.
      - Consider the likelihood of various risks. You will need to prioritize the critical areas.
      - Be aware of any vague or unclear requirements. Users often lack the expertise to understand technical language or procedures, so user misunderstanding could pose a risk.
      - Use your past "bug" history to help you identify areas for concern and extra testing.
    - ✓ **Write a section on what you are going to test -** List what new aspects you will be testing and what old aspects you will be re-testing. Make sure to detail the purpose for each test.
      - You can use software application inventories, IEEE guidelines, and other sources to help you determine this list.
      - This section also represents your "deliverables," or what data you will deliver to the client once the testing is complete.

Develop a phase Test Plan

LITHAN

- Steps
  - Writing the Test Plan (*continuation)*
    - ✓ **Write a section on what you will not be testing -** List any features that will not be tested during the current project. Reasons not to test features include:
      - The feature will not be included in this version of the software
      - The feature is low-risk or has been used before without issue
    - ✓ **List your strategy -**This section outlines the overall test strategy for your test plan. It will specify the rules and processes that will apply to the tests outlined above.
      - Include information on tools to be used, what metrics will be collected and at what level, how many configurations will be tested, and whether there are any special requirements or procedures for testing.

Develop a phase Test Plan

LITHAN

- Steps
  - Writing the Test Plan (*continuation)*
    - ✓ **Develop pass/fail criteria -** These criteria will guide your testing staff so that they know whether testing objectives have been achieved. This section can also include "exit criteria," so that your staff know when it is acceptable to stop testing a certain feature.
      - You should also include a list of suspension criteria and resumption requirements. This information tells testers when to pause tests and what the acceptable level of defect is to resume them.
    - ✓ **Write a list of documents that will be produced during testing** - Also known as "deliverables," these documents are the data, reports, scripts, and results that will be produced by testing.
      - It's a good idea to assign these deliverables to "owners" who are responsible for their delivery. Assign deadlines by which they are due.
    - ✓ **Write a section on the results of your project -** Outline all the goals that you hope to achieve during the testing process. Detail who is in charge of final approvals.

Develop a phase Test Plan (TIPS)

LITHAN

- Some software developers use an independent testing company to execute their test plans. With an independent company conducting the testing, the methodology and results can be scrutinized differently.

- If your software project is broken down into several sections with different teams, each team should create its own test plan. Each team's test plan can be combined into the overall project test plan after being reviewed and approved.

- A thorough test plan can remove the need for a test procedure, which can be costly to develop. Typically, test plans describe what product is being tested and test procedures describe how to test that product. However, a detailed test plan can cover the information normally outlined by a test procedure.

Develop a phase Test Plan (TIPS)

LITHAN

- Cater your test plan to the results you expect from the testing. Some testing is done to see what features succeed and some testing is done to see what will fail. Each requires different planning.

- To quickly come up with test cases and/or to minimize risk of forgetting something important to test, consider using test plan checklists and/or test plan templates. Especially useful when working on one product and adding new and new features to that product.

- To quickly come up with test cases and/or to minimize risk of forgetting something important to test, consider giving structure to your test plan. A very good structuring method for test plans is the ACC (attributes, components, capabilities) method. Identify attributes (adjectives describing the system), components (nouns for functional parts, features of the system) and for each combination of attribute and component, identify capabilities (verbs for user actions, activities).

Develop a phase Test Plan (Template & Guidelines)

LITHAN

- Test Plan Template
  - The format and content of a software test plan vary depending on the processes, standards, and test management tools being implemented. Nevertheless, the following format, which is based on IEEE standard for software test documentation, provides a summary of what a test plan can/should contain.
- Test Plan Guidelines
  - Be specific. For example, when you specify an operating system as a property of a test environment, mention the OS Edition/Version as well, not just the OS Name.
  - Have the test plan reviewed a number of times prior to baselining it or sending it for approval. The quality of your test plan speaks volumes about the quality of the testing you or your team is going to perform.
  - Update the plan as and when necessary. An outdated and unused document stinks and is worse than not having the document in the first place.

THANK YOU