Lance Jethro L. San Pablo
NT-4101
11/16/24

Summary of Chapters 4, 5, & 6

Chapter 4 tackles about Project Management Methodology. Under this is project management framework, which is a simple way to unite quality to project phases and synchronize project quality management to software development. Product quality refers to conformance and fitness for use, and project quality refers to delivering the product within the agreed scope and not over the budget. The project framework phases are similar to SDLC phases with its own corresponding activities. Product scope describes the characteristics of a product, while project scope specifies the work to deliver the product. Project charter is a document that recognizes funding of a project, where sponsors and customers are present with a brief summary of product scope and project scope. It should also be simple and nontechnical. The scope statement is a document that the PM updates as the team develops detailed requirements. Project Managers are responsible for coordinating with the project's stakeholders, and the earlier they are to define the product scope, the more effective they will be at managing. Test Managers on the other hand, is responsible for ensuring that a product meets the acceptable level of compliance with both functional and nonfunctional requirements. They should also always perform gap analysis to identify disparities between the requirements and specification documents. Avoiding duplication and defining test data are also done by test managers to ensure comprehensive test cases and data availability during execution of test cases. They also validate and analyze test results. Advices for the test manager includes requesting help, communicate issues, always update, learn new testing techniques, improve the process, and create a knowledge base. Project quality management has three major processes: quality planning, quality assurance, and quality control. Defect is a deviation from technical requirements, and they are classified into categories: showstopper (X)- impact is severe and system cannot be tested; critical (C)- impact is severe but an interim solution is available; noncritical (N)- all defects are not in X or C category. It also has priorities: high- further development cannot occur until defect is repaired; medium- defect must be resolved asap; low- defect is only an irritant. Defect tracking and reporting are techniques used by testers to capture every defect, and also show it to defect summary. Defect meetings to discuss every defect, and defect metrics: defect age- time duration between identification and closure; defect density- calculated per thousand lines of code.

Chapter 5 discusses the emerging specialized areas of testing. It talks about companies, over a period of time, realized that software testing is a must to prevent catastrophic bugs that have adverse effects on the business. This chapter also tackles different kinds of testing and various diagrams about testing inputs and outputs. The test design methodology defines how test cases are defined, trace-ability matrix is established, and how the test data is linked to the test cases. Regression testing is the selective testing of a system or component to verify that modifications have not caused unintended effects. Test automation approach determines how to ensure that requirements and end goals of the application are achieved. Identify applications to be tested and identify best automation tools are the things companies should always consider. Approaches such as test scripting and test execution are automated and offers wide range of advantages, such as limited human intervention. Additionally, testing scripts are also need to be regularly checked to determine where improvements can be implemented to improve the testing on the next iteration. Automation frameworks helped companies to realize that they need a structured automated testing framework to prevent failures. As an automation expert, one should understand that 100% of any standard applications cannot be automated. These are some of the best practices for a test automation framework: defining folder structure, modularize scripts, reuse generic functions and libraries, develop scripting guidelines and checklists, define error handling, and define maintenance process. Basic failures refer to failures at the system level (e.g. file not found, gui map not loaded, etc.), while application failures are failures of the application itself (e.g. unexpected pop-up window, page not found, button not found, etc.). Test Automation frameworks have evolved over a period of time depending on the maturity levels in the automation testing organization. The Data-Driven Framework, Modular Framework, Keyword-Driven Framework, and Hybrid Framework are some of the popular framework models that are being used across the test automation areas. Data-driven testing is a framework in which test input and output values are read from files to drive the tests. In Modular framework, it requires creation of small independent automation scripts, and are used in a hierarchical method to construct bigger tests. Keyword-driven is where different screens, functions, and business components are specified as keywords in a data table. Lastly, the hybrid framework is the combination of all the frameworks. Nonfunctional testing verifies how a system must behave, load testing is the practice of modeling the expected usage of the software by simulating multiple users concurrently, stress testing is where the load placed on system is beyond the normal expected usage, and volume testing is where data volume is increased to abnormal level. The different types of security testing are as follows: network scanning- uses a port to identify whether all hosts are connected to the network; vulnerability scanning- scans for outdated software versions, unpaid patches and system upgrades; password cracking- verifies whether a user is using a strong or weak password; log reviews- can be used to identify deviations from the security policy; file integrity checker- a tool to recognize changes to files, particularly unauthorized changes; virus detectors- anti-virus software to detect viruses; pen testing- evaluators attempt to circumvent the security of the system based on their understanding. Compliance testing determines that a product implementation of a particular implementation specification fulfills all mandatory elements as specified and that these elements are operable. SOA testing or service oriented architecture testing is viewing the overall business process and ensure that components interact properly. A user story is an informal statement of the requirement. With agile development, a user story is a software system requirement formulated as one or two sentences in the everyday language of the user. Test-driven development- short iterations to cover functionality first, then production code is implemented;

Chapter 6 is about modern software testing tools. According to ASQ, project management is the application of skills, knowledge, tools and techniques to meet the requirements of a project. Automated testing tools was originally to automate regression testing, which requires that a tester has developed detailed test cases that are repeatable. While automated testing tools fully automate the process of testing, companies will still need to hire a programmer or tester solely for maintaining and creating the test scripts. Automated testing tools are typically used for function/gui testing. Test scenarios are a manual process that testers have to translate the requirements to test data. GIGO stands for "garbage in, garbage out", and it is commonly used to describe failures in human decision making due to faulty and incomplete data. Finding the appropriate tool can be difficult, that is why "Testing Tool Selection Checklist" lists questions that help the QA team evaluate and select automated testing tool. Some of the available tool types are: Function/regression tools—These tools help you test software through a native graphical user interface (GUI) to ensure the functionality of the system; Bug management tools—These tools help you track software product defects and manage product enhancement requests; Unit testing tools—These tools help you unit test software, which is usually performed by the developer; Database testing tools—These tools help you verify database integrity, business rules, access, and refresh capabilities; Web testing tools—These tools help you locate broken Web links and evaluate the performance of Web-based systems under heavy loads. Using automated testing tools varies on the appropriateness of the testing that is needed to be done. It gives a different and more distinct benefits compared to testing with human intervention. However, there are also times when automation fails. The three primary reasons are steep learning curve, development effort required, and the maintenance overhead. It is always not wise to purchase a testing tool, and here are some factors that limit it: unrealistic expectations, false sense of security, technical difficulties, cost, one-time testing, adhoc testing, time crunch, predictable results.