|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| Capstone: Test Cycle  Introduction and Test Cycle |
| |  |  |  | | --- | --- | --- | | Tyler L. Kemnitz | 9/22/23 | Capstone: Application Development | |

# Introduction

In the realm of software development, ensuring the reliability and quality of a product is paramount. To achieve this, a structured testing cycle is employed—an iterative journey that encompasses various testing types and phases conducted throughout the development process. The primary objective is to detect and rectify issues early in the software's lifecycle, guaranteeing a dependable and high-quality end product.

One critical facet of this testing cycle is regression testing, a linchpin in the pursuit of software excellence. It plays a pivotal role in safeguarding the integrity of existing functionality when new features or fixes are introduced.

Each phase within this comprehensive testing cycle serves a unique purpose, boasting a defined scope, a structured process, and the potential employment of specific tools to facilitate management and documentation. This orchestrated approach to testing not only ensures the elimination of defects but also paves the way for a smoother, more reliable software development journey.

# Test Cycle

## Unit Testing:

* + **Purpose**: To test individual components (functions, methods, or classes) of the code.
  + **Scope**: Focused on isolated parts of the code.
  + **Tools**: Testing frameworks like JUnit, NUnit, Postman, Swagger or built-in testing tools.
  + **Process**: Developers write unit tests to validate the behavior of individual code units.

## Integration Testing:

* + **Purpose**: To ensure that different components of the code work together as expected.
  + **Scope**: Testing interactions between modules or services.
  + **Tools**: Integration testing frameworks, test launches, and mocking frameworks.
  + **Process**: Test cases are designed to verify how components integrate and exchange data.

## Functional Testing:

* + **Purpose**: To validate that the software meets its functional requirements.
  + **Scope**: End-to-end testing of features and user scenarios.
  + **Tools**: Automated testing tools like Selenium, Cucumber, or manual testing.
  + **Process**: Test cases are created based on user stories or requirements and executed to ensure features work as expected.

## Regression Testing:

* + **Purpose**: To check if new code changes have introduced any unintended side effects or broken existing functionality.
  + **Scope**: Repeated testing of previously tested functionality.
  + **Tools**: Automated testing frameworks, test suites.
  + **Process**: Execute a predefined set of test cases covering critical paths and known areas of concern after code changes or updates.

## User Acceptance Testing (UAT):

* + **Purpose**: To validate the software from the user's perspective and ensure it meets their needs.
  + **Scope**: Real-world scenarios performed by end-users or stakeholders.
  + **Tools**: Manual testing, user feedback, user stories.
  + **Process**: End-users execute test cases or interact with the system to verify that it aligns with their requirements.

## Deployment Testing:

* + **Purpose**: To ensure a smooth deployment process and minimize downtime.
  + **Scope**: Verifying that deployment scripts and configurations work as intended.
  + **Tools**: Deployment automation tools, monitoring systems.
  + **Process**: Test the deployment process in a test environment to catch any issues before production deployment.

## Exploratory Testing:

* + **Purpose**: To uncover unexpected issues and defects that scripted tests might miss.
  + **Scope**: Unstructured testing, often used in combination with other testing types.
  + **Tools**: None, manual testing.
  + **Process**: Testers explore the software, trying different inputs and scenarios, and reporting any anomalies.

# Test Cycle Timing

A diagram of a test cycle

Description automatically generated